Local Hough Transform for 3D Primitive Detection

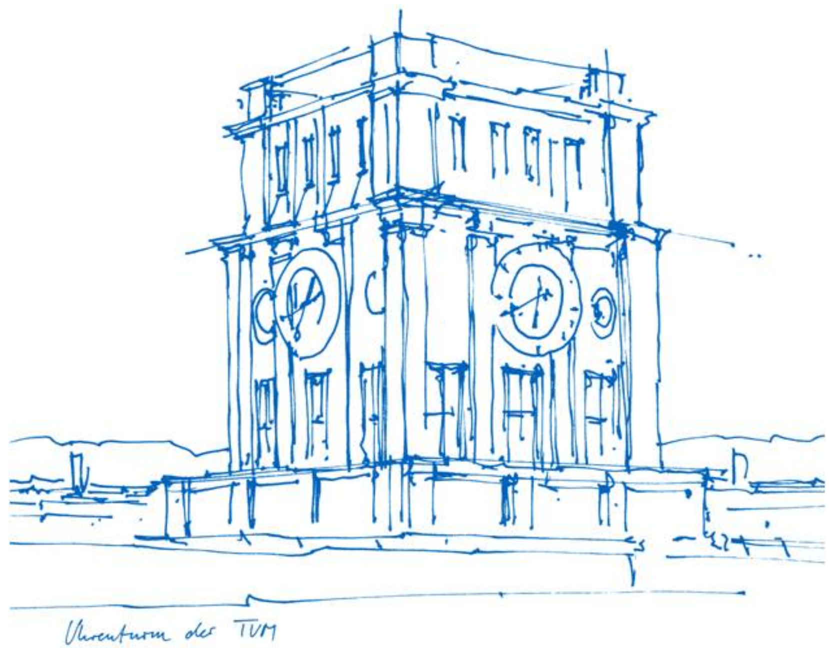
Karthikeya Ramesh Kaushik

Technical University of Munich

Department of Informatics

Chair for Computer Vision and AI

26. June 2019, Garching



Introduction

- A geometric primitive is a simple decomposable shape in a 3D system. Examples include plane, sphere, cylinder, pyramid etc.,
- 3D Primitive detection is an important application in scene understanding, robotics, reverse engineering and other applications.
- Primitives considered in this paper – Planes, Spheres and Cylinders.
- Advantage of Hough transform is the deterministic runtime (details follow).
- Primary disadvantage is that the voting space increases exponentially with number of parameters, thus slower processing, more memory needs and difficulty sampling.

Introduction

- Proposed solution utilises the inherent symmetry present in all geometric primitives.
- Makes detection faster, robust and more generic.

Outline

1. Related work
2. Methodology
   - Local Hough Transform
     - Point Pairs
     - Plane
     - Sphere
     - Cylinder
   - Local Parameter Space
   - Voting Scheme
   - Detection Pipeline
3. Evaluation

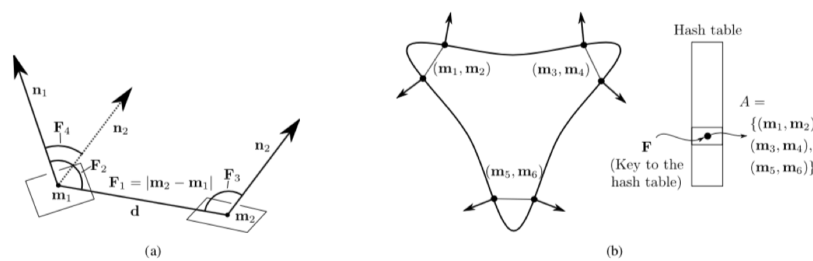
Related Work

- Region Growing – Oversegmentation, and combining spatially similar segments.
- RANSAC – Random sampling of scene points, and finding the primitive that best describes this set of points. Slow to compute, and non-deterministic.
- Voting – Detected features vote for parameters in the discretised parameters space, according to what explains them best.
- Other approaches – Semi Manual approach where user selects primitive type and one point. Unsuitable as it is not fully automatic.

Local Hough Transform

- Point pairs are pairs of oriented 3D points which are used as basic features.
- These basic features (of the model, not the scene) are used in an offline phase to build a hash table.
- Obvious disadvantage of building such a table is – absence of shape parameters, possible mismatches at boundary etc.,

$$F(p_1, p_2) = (|d|, \measuredangle(n_1, d), \measuredangle(n_2, d), \measuredangle(n_1, n_2))$$



Features of primitives

- Implicit nature of primitives is leveraged to match point pairs and extract shape parameters.

1. Plane : Ideally all points have parallel orientation with each other.

$$F(p1, p2) = (|d|, \pi/2, \pi/2, 0)$$

$$\begin{aligned} F(p1, p2) \in \mathbb{R} \times & [\pi/2 - \delta_{\measuredangle}, \pi/2 + \delta_{\measuredangle}] \times \\ & [\pi/2 - \delta_{\measuredangle}, \pi/2 + \delta_{\measuredangle}] \times \\ & [0, 2\delta_{\measuredangle}] \end{aligned}$$

Features of primitives

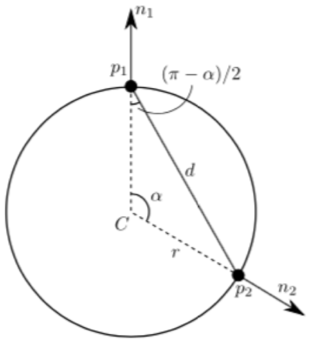
2. Sphere : Point pair can be described based on angle formed at the center.

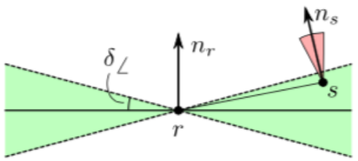
$$F(p_1, p_2) = (|d|, \measuredangle(n_1, d), \measuredangle(n_2, d), \measuredangle(n_1, n_2)) \\ = (2r\sin(\alpha/2), (\pi-\alpha)/2, (\pi+\alpha)/2, \alpha)$$

$$\overline{\alpha} \in [\max(0, \alpha - 2\delta_n), \alpha + 2\delta_n] \\ \overline{d} \in [\max(0, |d| - \delta_p), |d| + \delta_p]$$

Possible thresholds on angular and distance noise are set, and range of values of radius is calculated.

$$r \in \left[\frac{|d| - \delta_p}{2\sin((\alpha + 2\delta_n)/2)}, \frac{|d| + \delta_p}{2\sin(\max(0, \alpha - 2\delta_n)/2)}\right]$$





Features of primitives

3. Cylinder : Although an implicit cylinder model is possible, it is computationally expensive. Instead, a unit model is precreated. The features are calculated with radius 1 and some max *l*.

$$F(p_1, p_2) = (\sqrt{d^2 + l^2}, \measuredangle((1,0,0)^T, (r(1-\cos\alpha), r\sin\alpha, l)^T), \pi - F_2, \alpha)$$

Only the first term scales linearly with radius. From observed data, *F2*,*F3* and *F4* are matched, and the radius is calculated by computing

$$r = F_1^r / F_1^1$$

The matching pair is discarded if radius exceeds a certain threshold.

Local Parameter Space

- In “Model Locally, Match Globally“, Drost et al suggested a parametrisation utilising only (m,**α).**
- While (m, α) is good for parametrising the transformation of an arbitrary free-form object, it is an overparameterization for geometric primitives.
- Reduction in parameter space is obtained as a natural result of the way the primitives have been defined.

	Number of parameters		
Shape	**Rigid**	**Local**	**Shape**
Free-form	6	3	0
Plane	3	0	0
Sphere	3	0	1 (radius)
Cylinder	4	1	1 (radius)

Local Parameter Space

- The parameter space of spheres and cylinders need to be extended keeping in mind the scale of these objects. Therefore, the radius is added to the parameter space to account for detect differently scaled objects.

Voting Scheme

- Hough transform is performed on the local parameter space.
- For a given reference point *r*, iteration is performed over all sampled points *s* in the scene *S*, and a point pair feature is calculated for (*r*,*s*).
- The parameter space is extracted from this feature for the primitives, and a vote is cast for matches.
- Voting is local, and is performed for all reference points that are sampled with sampling distance being the smallest expected size of primitive, to ensure coverage over all possible locations of primitive.

Detection pipeline

- In order to remove similar candidates due to noise, refinement and non maximum suppresion is performed.
- Segmentation : During the segmentation step, by means of thresholding and successive segmentation into connected components, detected points that are not on the primitive are discarded.
- Refinement : During the refinement step, iteration is performed over all scene points to identify more precisely which points are on or inside the primitive.

Detection pipeline

- The target energy equation

$$E(p) = \sum_{s \in S} w_s d_{\text{primitive}}(p, s)^2$$

- Reduction of Energy equation

$$E(p) = e^T e$$
$$\nabla E(p) = 2(\nabla e)^T e$$

- Update step

$$(J^T J) d_k = -J^T e$$
$$p_{k+1} = p_k + d_k$$

- Weight recomputation

$$w_c = w(d) = \begin{cases} (1 - d^2/d_{max}^2)^2 & \text{if } d < d_{max} \\ 0 & \text{else} \end{cases}$$

- Primitives are described by : point *c* and normal *n* for plane, sphere by center *c* and radius *r,* cylinder by (T*,r)* where T is SE(3).

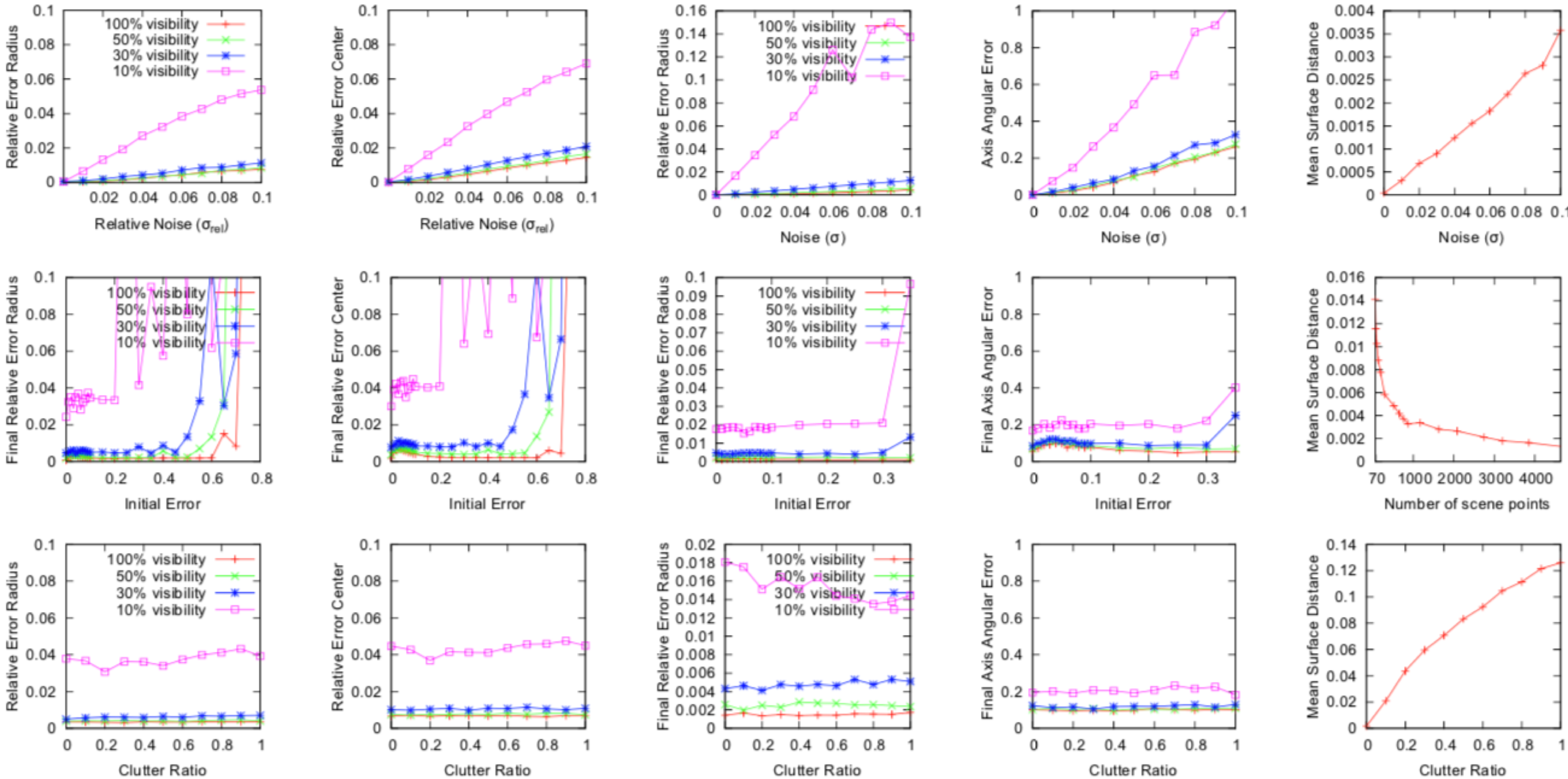
Detection pipeline

- Non Maximum Suppresion : NMS removes duplicates that are too similar by only keeping the candidate that has the highest score.

- In summary, NMS is first applied on candidates from voting step, after which refinement is performed in coarse to fine manner ie., Refinement on subsampled data after which refinement on full data. NMS is applied again to remove duplicate candidates.

Evaluation

- Refinement evaluation :



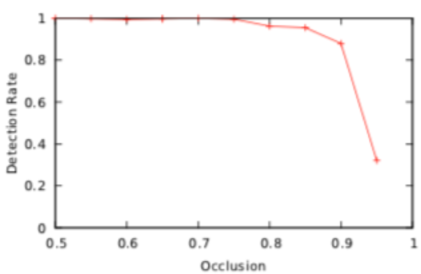
Sphere r Sphere c Cylinder r Cylinder n Plane

Evaluation

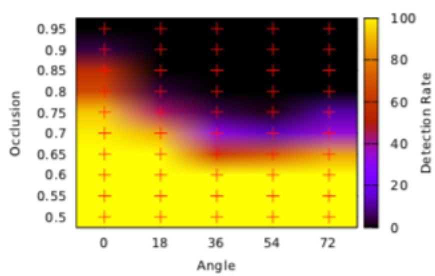
- Detection : Quantitative (ABW SecComp)

approach	correct	over	under	missed	noise
USF	12.7 (83.5%)	0.2	0.1	2.1	1.2
WSU	9.7 (63.8%)	0.5	0.2	4.5	2.2
UB	12.8 (84.2%)	0.5	0.1	1.7	2.1
UE	13.4 (88.1%)	0.4	0.2	1.1	0.8
OU	9.8 (64.4%)	0.2	0.4	4.4	3.2
PPU	6.8 (44.7%)	0.1	2.1	3.4	2.0
UA	4.9 (32.2%)	0.3	2.2	3.6	3.2
UFPR	13.0 (85.5%)	0.5	0.1	1.6	1.4
MRPS	11.1 (73.0%)	0.2	0.7	2.2	0.8
ours	12.3 (80.7%)	0.2	0.8	2.6	0.0

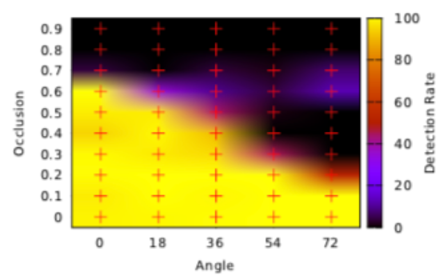
- Synthetic dataset



Spheres



Cylinders



Planes

Thank you