Local Hough Transform for 3D Primitive Detection

Karthikeya Ramesh Kaushik

Department of Informatics - Technische Universität München

Abstract

Primitive detection is an important application in 3D Computer Vision, primarily from the point of view of automation, robotics and reverse engineering. The Hough Transform is a technique where votes are accumulated for parameters that best explain detected features. But in order to keep the voting space manageable, a local voting space is suggested. In this local Hough Transform, voting is performed on sub-manifolds of the original parameter space. The primitive which is aligned with the scene point is then locally voted for and recovered. This scheme is used in conjunction with a coarse-to-fine filtering pipeline to refine candidates and remove duplicates. Evaluation indicates that the detection is robust against noise and clutter.

1 Introduction

Geometric primitives are easily visualisable and explainable, making them very important from the point of reverse engineering, robotics etc., Most 3D objects can be shown to consist of multiple primitives, which helps in processing on a higher, abstract level. The primitives dealt with here include planes, spheres, and cylinders.

Earlier primitive detection schemes like RANSAC and oversegmentation followed by region growing have nondeterministic runtimes. The Hough transform relies on voting to identify which model is best explained in the scene. Although it has a deterministic runtime, the global Hough transform has a drawback because its voting space increases exponentially with parameter size.

In order to overcome this problem, a local voting scheme is suggested in [1] where local scene points vote for what primitive best explain them. This scheme is made more efficient by using the inherent symmetry present in the primitives selected to reduce the parameter space further.

2 Related Work

2.1 Region Growing

This is a method wherein the scene is first decomposed or oversegmented, and regions that are consistent with a primitive are combined successively. There are different ways of combination, as well as segmentation. [7] fit B-splines of first or segment order into adjacent segments, combining them if the fit of the combined segment is better than the individuals themselves. [4] combine planes that lie on the same curvature/edges that have been extracted beforehand. In summary, the common technique being a combination of common structural entities from an oversegmentation resulting in the primitive detection.

2.2 RANSAC

This approach first selects randomly a set of scene points and computes a primitive that contains all of these points, and the primitive is evaluated by checking against all scene points[5]. [3] uses domain specific knowledge of shape sizes in the processing.

2.3 Voting

Voting schemes based on the Hough transform recover primitive parameters by voting for discretized cell candidates (which are the parameters). As described by [2], it is hard to find a Hough space that is accurate, fast, and has good runtime.

2.4 Others

Some of the methods suggested are semi manual, where the user chooses one primitive point and the type of primitive [6]. The primitive is then fit to the neighborhood of the selected point. Although a robust method, it is not automatic.

3 Local Hough Transform

3.1 Point Pair Features

The proposed voting scheme utilizes point pair features to quantify the relationship between two points as follows :

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

where $d = p_1 - p_2$. In [1] the preliminary step involved processing the sampled model points and building a hash table of the point pair features computed. During the detection stage, the point pair features of the scene were matched against the previously computed hash table, and the primitive with maximum votes was retrieved. The primary disadvantages of this method are the necessity of the offline phase, inability to include any size information about the model, and no explicit inclusion of noise information.

In order to overcome these issues, the intrinsic properties of the primitives, along with noise estimates are used to describe the primitives. δ_p and $\delta_\angle$ indicate the noise in position and angle respectively.

3.2 Plane

All points on the plane are have their normals in parallel, with arbitrary distance between the points. The angle between the point normal and the difference vector is 90°. Ideally, the point pair feature would look like :

$$F(p_1, p_2) = (|d|, \pi/2, \pi/2, 0)$$

Most realistic observations are noisy, and therefore the normals are not always perfectly parallel, and the difference vector is not perfectly perpendicular to the normal. This is included in the noisy estimate of the point pair feature (positional noise is uninformative, since the distances are arbitrary).

$$F(p_1, p_2) \in R \times [\pi/2 + \delta_\angle, \pi/2 - \delta_\angle] \times [\pi/2 + \delta_\angle, \pi/2 - \delta_\angle] \times [0, 2\delta_\angle]$$

3.3 Sphere

Given two points on a sphere, the feature vector can be described using the angle formed at the center, and the radius of the sphere. From the feature vector $(F_1, F_2, F_3 F_4)$, we find angle α as F_4, and d from F_1. The exact ground truths are calculated as a range of values from the error estimates as follows

$$F(p_1, p_2) = (2r\sin(\alpha/2), (\pi - \alpha)/2, (\pi + \alpha)/2, \alpha)$$

From this the range of possible radii is calculated as

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

For very small values of α (points very close to each other on the surface) less than the noise estimate, the upper boundary for r tends to infinity. In such cases, F_1 and F_4 are used to find if the points lie on the sphere using the possible range of values, and are discarded if not.

3.4 Cylinder

Since it is difficult to find an implicit model of a cylinder, a model is pre-created with a fixed radius 1, some maximum length l, and the feature vectors are calculated for this model. For parameters r being the radius, l the length along the main axis, α the angle between two points when projected along the cylinder's axis and d the distance between points projected along the axis. The feature vector then is

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

Scaling by 1/r gives us

$$F^1(p_1, p_2) = (F_1^r/r, F_2^r, F_3^r, F_4^r)$$

During the detection phase, the scale invariant features F_2, F_3 and F_4 are matched with the precomputed features, and the radius is calculated using $r = F_1^r/F_1$. If the radius is found to be out of range, the point pair is discarded. For points very far from each other, the feature vector looks like

$$F(p_1, p_2) = (|p_1 - p_2|, \pi/2, \pi/2, \alpha)$$

In this case for all distances that exceed the $|p_1 - p_2|$ threshold, special bins are added for which $F_2 = F_3 = \pi/2$ and $F_4 \in [\pi, 2]$.

4 Local Parameter Space

[1] suggested (m, α) as a parametrization, where m is the reference point on the model, and α is the angle the normal associated with m has to be rotated by to lie on the z axis. While $(m,\ \alpha)$ is a good parametrization for free form objects, it is an overparametrization for primitives. Due to symmetries around normals of reference points of planes and spheres, they look identical from each point on their surfaces. Thus, one 3D point and its normal fully define the primitive. In case of cylinder with known radius, only the angle of rotation α w.r.t to the reference point is sufficient, as in [1].

In order to include the scale of the primitive in case of the sphere and cylinder, the parameter space is extended by radius r to include differently sized objects. The table below summarises the parameter space.

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

5 Voting Scheme

Hough Transform is performed on the reference points that have been uniformly sampled from the scene space, and the primitive's local parameters are extracted and a vote is cast for each possible match. The sampling frequency is such that all the spacing between the chosen points is less than the minimum expected size of the primitive.

6 Detection Pipeline

6.1 Segmentation

In order to remove points that lie too far away in the scene, an initial segmentation is performed. This is necessary, since the voting does not provide bounding information and may include candidates that are too far in reality. All points on the primitive, along with noise thresholds are included in the set of possible points. They are then segmented using connected components, and user defined segmentation distance, and only the component containing the reference point is kept for further refinement.

6.2 Refinement

In order to minimize the number of noisy points relating to a primitive, an iterative weighting scheme is used. The target energy to be minimized is :

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

Where p is the primitive's parameters, and $d(p, s)$ is the distance of the point to the primitive's surface. The energy $E(p)$ is then written as the square of the residuals, and using the Jacobian notation as

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

where the Jacobian matrix $J = \nabla e$. The Gauss-Newton update step is performed as follows

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

The optimization is iterated and the weights are calculated as follows :

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

Planes are parametrized using a point c, and its normal. Spheres using center c, and radius r. Cylinders are parametrized by (T, r) where T is a rigid 3D transformation mapping the cylinder's main axis to the z axis. The distance of point from the cylinder then is

$$d_{cylinder}(p, s) = (|Ts - Ts.(0, 0, 1)^T| - r)$$

In order to avoid overparameterization of the update step, the update is parameterized as

$$\delta_p = (\delta_r, \delta_{tx}, \delta_{ty}, \delta_{rx}, \delta_{ry})$$

where δ_r encodes the change in the cylinder's radius, δ_{tx} and δ_{ty} encode the change in the cylinder's position, orthogonal to its main axis, and δ_{rx} and δ_{ry} encode the change in the cylinder's main axis by tilting it. The update δ_T is thus composed of the translation $(\delta_{tx}, \delta_{ty}, 0)^T$ and the described rotation. Given current cylinder parameters $p = (T, r)$ and an update δp, the new parameters $p' = (T', r')$ are computed as

$$r' = r - \delta^T$$
$$T' = \delta_T^{-1} T$$

6.3 Non Maximum Suppression

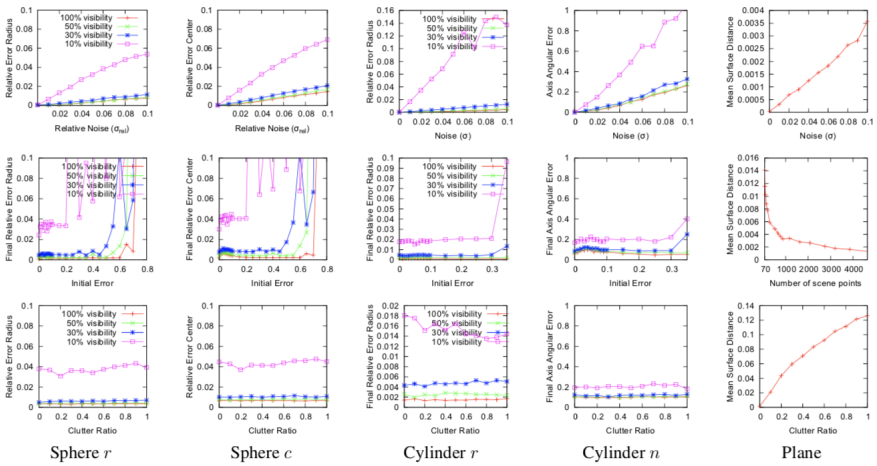
NMS is needed to remove duplicate candidates generated by multiple reference points and reduce processing time for subsequent steps. All candidates are first sorted by score. In a best to worst fashion, for each one, all similar candidates with lower scores are removed. Indexing by direction of main axis (for cylinder), normal vector (for plane), and center point (for circle) speeds up the process.

In summary, firstly NMS is applied to candidates from voting step. Refinement is then performed in a coarse to fine manner. Coarse sampling on the sub-sampled data ensures that highly erroneous components are removed very quickly, after which the slower, more accurate full cloud refinement is performed. After voting and refinement, NMS is used to remove duplicate candidates.

7 Evaluation

7.1 Refinement

In order to check the robustness of the refinement process, evaluation was performed with known ground truth, based on three parameters – cluttering by close range points, noisy initial parameters, noisy data, and visibility of primitive. Measurement is standardized by considering distance based values relative to size of primitive : Maximum diameter of primitive, and radius of sphere and cylinder, as shown in the figure. Results indicate that the refinement is robust against clutter and does well in noisy environments. For visibility <10%, the primitive starts to seem more planar, and the result of refinement deteriorates.



7.2 Detection

Evaluation on the ABW SecComp dataset of planes showed 80.7% detection accuracy, less than the best of 88.1% (UFPR), but the observation is the zero noise in detection ie., No False positives, making it the most robust detection scheme. Most misses were due to a high angle of inclination of the GT plane, which meant very few disconnected pixels of the plane were visible, on which the normal estimator failed, leading to a loss in detection. On a synthetic dataset with 1000 scenes with known GT, evaluation results were promising for well positioned primitives with low occlusion. Qualitative tests showed difficulty identifying cylinder length, but robustness despite occlusions was found.

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

8 Conclusion

The Local Hough transform avoids the ballooning of the parameter space as seen with global methods. The voting scheme is embedded in a coarse to fine refinement scheme. The Local Hough Transform also guarantees deterministic runtimes. The paper discusses only qualitatively the improvement in the parametrization by using symmetry, but does not provide any quantitative speed up results. There is also no indication regarding the runtime or hardware used for computation. Parallelization is possible during the point pair feature calculation of multiple reference points. Evaluation results indicate that refinement works well even with low visibility and noisy conditions. The only detection failure is in the case of bad alignment and very low visibility, which is an ill posed problem due to the primitives becoming more and more planar. With a deterministic runtime and robust refinement scheme, this method proves to be a competitive candidate against RANSAC and Oversegmentation based approaches.

References

[1] N. Navab B. Drost, M. Ulrich and S. Ilic. Model globally, match locally: Efficient and robust 3d object recognition. *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 998–1005. IEEE, 2010.*

[2] K. Lingemann D. Bornmann, J. Elseberg and A. Nüchter. The 3d hough transform for plane detection in point clouds:a review and a new accumulator design. *3D Research, 2(2):1–13, 2011.*

[3] T. Landes F. Tarsha-Kurdi and P. Grussenmeyer. Hough- transform and extended ransac algorithms for automatic de- tection of 3d building roof planes from lidar data. *ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007, vol- ume 36, pages 407–412, 2007.*

[4] andL.Silva P.F.Gotardo, O.R.P.Bellon. Rangeimageseg- mentation by surface extraction using an improved robust es- timator. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, volume 2, pages II–33. IEEE, 2003.*

[5] R. Wahl R. Schnabel and R. Klein. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum, volume 26, pages 214–226. Citeseer, 2007.*

[6] S. Roth-Koch S. J. Ahn, I. Effenberger and E. Westkämper. Geometric segmentation and object recognition in unordered and incomplete point cloud. *Pattern Recognition, pages 450–457. Springer, 2003.*

[7] J. Prankl M. Zillich T. Morwald, A. Richtsfeld and M. Vincze. Geometric data abstraction using b-splines for range image segmentation. *In Robotics and Automation (ICRA), 2013 IEEE International Conference on, pages 148– 153. IEEE, 2013.*