Demisting the Hough Transform
for 3D Shape Recognition and Registration

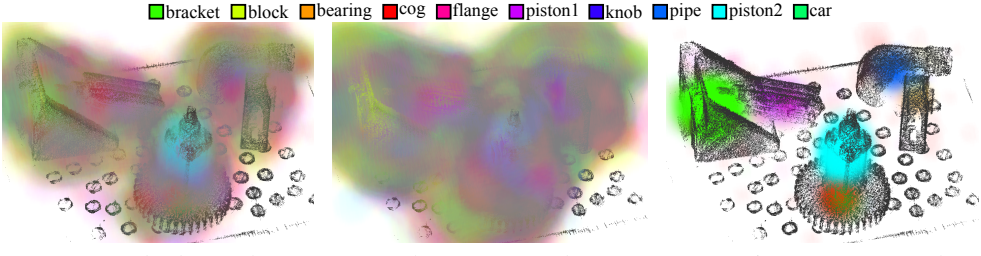
Oliver J. Woodford
Minh-Tri Pham
Atsuto Maki
Frank Perbet
Björn Stenger

Computer Vision Group,
Toshiba Research Europe Ltd.
Cambridge, UK

firstname.lastname@crl.toshiba.co.uk
www.toshiba-europe.com/research/crl/cvg/

Abstract

In applying the Hough transform to the problem of 3D shape recognition and registration, we develop two new and powerful improvements to this popular inference method. The first, *intrinsic Hough*, solves the problem of exponential memory requirements of the standard Hough transform by exploiting the sparsity of the Hough space. The second, *minimum-entropy Hough*, explains away incorrect votes, substantially reducing the number of modes in the posterior distribution of class and pose, and improving precision. Our experiments demonstrate that these contributions make the Hough transform not only tractable but also highly accurate for our example application. Both contributions can be applied to other applications that already use the standard Hough transform, as well as making it feasible and competitive for potentially many more.



(a) Standard Hough (b) Max. Hough (c) Min.-entropy Hough

Figure 1: **Demisting the Hough transform.** Posterior distributions over translation and ten object classes (six of which are present in the scene), with scale and rotation marginalized out, for the three types of Hough transform tested: (a) the standard Hough transform, (b) the max. Hough transform and (c) the minimum-entropy Hough transform.

1 Introduction

The Hough transform [7], named after Hough's 1962 patent [10] describing a method for detecting lines in images, has since been generalized to detecting, as well as recognizing, many other objects: parameterized curves [7], arbitrary 2D shapes [2], cars [9, 15], pedestrians [3, 9], hands [19] and 3D shapes [12, 20, 24], to name but a few. This popularity stems

from the simplicity and generality of the first step of the Hough transform—the conversion of *features*, found in the data space, into sets of *votes* in a Hough space, parameterized by the pose of the object(s) to be found. Various different approaches to learning this feature-to-vote conversion function have been proposed, including the *implicit shape model* [15] and *Hough forests* [9, 19].

The second stage of the Hough transform simply sums the likelihoods of the votes at each location in Hough space, then selects the modes. One problem with this step is that the summation can create modes where there are only a few outlier votes. Indeed, as stated in [9], no probabilistic interpretation that fully explains this approach (since likelihoods of independent variables should be multiplied, not added) has yet been provided. A second problem is that, given a required accuracy, the size of the Hough space is exponential in its dimensionality.

The application we are concerned with, object recognition and registration (R&R) from 3D geometry (here, point clouds), suffers significantly from both these problems. The Hough space, at 8D (one dimension for class, three for rotation, three for translation and one for scale), is to our knowledge the largest to which the Hough transform has been applied, and the feature-to-vote conversion generates a high proportion of incorrect votes, creating a "mist" of object likelihood throughout that space, as shown in figure 1(a).

In the face of this adversity, we have developed two important contributions which enable inference on this task, and potentially many others, using the Hough transform to be both feasible and accurate:

- We introduce the *intrinsic Hough transform*, which substantially reduces memory and computational requirements in applications with a high dimensional Hough space.
- We introduce the *minimum-entropy Hough transform*, which greatly improves the precision and robustness of the Hough transform.

These extensions of the Hough transform are not task specific; they can be applied, either together or independently, to any application that does or is able to use the standard Hough transform.

The rest of this paper is organized as follows: The next section describes inference using the Hough transform, and briefly reviews the literature relevant to our contributions. In §3 we describe our new inference methods. The section following that describes and discusses our experiments. Finally, we conclude in §5.

2 Background

2.1 3D shape recognition and registration

The *implicit shape model* of Leibe *et al*. [14, 15] pioneered the use of the Hough transform for object recognition in 2D images. This approach has since been applied to object recognition in 3D geometric data [12], and extended to object registration [20, 24]. For the dual problem of R&R in 3D, the Hough space is either 7D (if scale is known) [24] or 8D [20].

The feature extraction stages of these methods follow the same pipeline: features are detected at a given scale and position; a canonical orientation of the feature is computed; a descriptor for the feature is computed. The votes are then computed by matching features in the test data with features from training data with ground truth class and pose, either directly

(*i.e.* a nearest neighbour search) [20], or via a codebook created by clustering the feature descriptors [12, 15, 24].

In this work we will be using the feature-to-vote conversion process of Pham *et al.* [20] as an off-the-shelf method, since our contributions lie in the second stage of the Hough transform. It is this process that generates a high proportion of incorrect votes, amongst which the correct votes need to be found.

2.2 The Hough transform

We consider the second stage of the Hough transform to be a discriminative model of the posterior distribution of an object's location, $\mathbf{y}$, in a Hough space, $\mathcal{H}$, which is the space of all object poses (usually real) and, in the case of object recognition tasks, object classes (discrete). The model is a non-parametric kernel density estimate based on the votes, $\mathbf{X} = \{\mathbf{x}_{ij}\}_{\forall i,j}$, cast in $\mathcal{H}$ by N features, thus

$$p(\mathbf{y}|\mathbf{X},\omega,\theta) = \sum_{i=1}^{N} \omega_i \sum_{j=1}^{J_i} \theta_{ij} K(\mathbf{x}_{ij},\mathbf{y}), \tag{1}$$

where J_i is the number of votes generated by the i^{th} feature, $K(\cdot,\cdot)$ is a density kernel in Hough space, and $\omega = \{\omega_i\}_{i=1}^{N}$ and $\theta = \{\theta_{ij}\}_{\forall i,j}$ are feature and vote weights respectively, s.t. $\omega_i, \theta_{ij} \geq 0$, $\forall i,j$, $\sum_{i=1}^{N} \omega_i = 1$, and

$$\sum_{j=1}^{J_i} \theta_{ij} = 1, \quad \forall i \in \{1,..,N\}. \tag{2}$$

For example, in the original Hough transform used for line detection [7], the features are edgels, votes are generated for a discrete set of lines (parameterized by angle) passing through each edgel, the kernel, $K(\cdot,\cdot)$, returns 1 for the nearest point in the discretized Hough space to the input vote, 0 otherwise, and the weights, ω and θ, are set to uniform distributions. Recently a method has been proposed for learning *a priori* more discriminative weights [18], for object detection.

The final stage of the Hough transform involves finding, using non-maxima suppression, the modes of this distribution whose probabilities are above a certain threshold value, τ.

2.2.1 Computational feasibility

Finding the modes in $\mathcal{H}$ involves sampling that space, the volume of which increases exponentially with its dimensionality, d. Several approaches have been proposed to reduce this burden, which we categorize as one of approximate, hierarchical, irregular or mode-seeking.

Approximate methods use reduced-dimensionality approximations of the full Hough space to find modes. For example, given a 6D pose (translation & rotation), Fisher *et al.* [8] quantize translations and rotations in two separate 3D arrays (peak entries in both arrays indicate an object, but multiple objects create ambiguities), while Tombari & Di Stefano [24] find modes over translation only, then compute an average rotation for each mode. Geometric hashing techniques, *e.g.* [13], also fall into this category.

Hierarchical approaches, such as the *fast* [16] and *adaptive* [11] Hough transforms, sample the space in a coarse-to-fine manner, exploiting the sparsity of some areas, though their complexity is still exponential in d.

Irregular methods do not sample the Hough space regularly, but rather sample only where objects are likely to be detected, again exploiting potential sparsity in $\mathcal{H}$. For example, the *combinatorial* [9] and *randomized* [27] Hough transforms generate lists of sampling locations, the former for all lines (in line dection) joining pairs of edgels in confined regions, the latter for curves (in curve detection) defined uniquely by random sets of edgels. Both these approaches are task specific, whereas the intrinsic Hough transform introduced here, which also falls into this category, is not.

Mode-seeking methods find modes in $\mathcal{H}$ through iterative optimization. Mean shift [6] is the most commonly used approach, the complexity of which is $O(nd^2)$, where $n = \sum_{i=1}^{N} J_i$ (the total number of votes). It has successfully been applied to an 8D Hough space [20]. However, it needs to be initialized in many, perhaps $O(n)$, locations, making the total complexity $O(n^2d^2)$, and is not guaranteed to find every mode. Two extensions of this approach, though generally applied to clustering rather than mode seeking, are medoid shift [23] and quick shift [25].

These approaches can also be combined. For example, modes found in a coarse sampling of $\mathcal{H}$ can be refined using mean shift [15], an approach we employ here.

2.2.2 Explaining away votes

The summing of votes in the Hough transform enables incorrect votes to generate significant modes in $\mathcal{H}$. Recently Barinova *et al.* [3] introduced an alternative vote-based inference method which correctly (in a probabilistic sense) multiplies the vote likelihoods. In addition, they novelly exploit the assumption that *only one vote cast by each feature is correct*[1], with the result that correct votes are able to explain away incorrect votes from the same feature for the first time. They demonstrated state-of-the-art results on line and pedestrian detection using this method. We exploit this same assumption in our minimum-entropy Hough transform, with similar results.

3 Our framework

This section describes our improvements to the Hough transform. In §3.1 we introduce the intrinsic Hough transform, which overcomes the high memory requirements of the standard Hough transform with high-dimensional Hough spaces. In §3.2 & 3.3 we introduce two methods which exploit the assumption that only one vote per feature is correct.

3.1 The intrinsic Hough transform

As discussed in §2.2, high-dimensional Hough spaces require infeasible amounts of memory to sample regularly. However, we note that while the Hough space increases exponentially with its dimensionality, the number of votes generated in applications using the Hough transform generally do not, implying that higher dimensional Hough spaces are often sparser. We exploit this sparsity by sampling the Hough space only at locations where the probability (given by equation (1)) is likely to be non-zero—at the locations of the votes themselves. Since the votes define the distribution, therefore are intrinsic to it, we call this approach the *intrinsic Hough transform*.

[1] This assumption is empirically valid for applications in which each feature is generated by a single object, which is usually the case.

While similiar in some respects to intrinsic mode-seeking algorithms [23, 25], the intrinsic Hough transform does not seek modes through iterative updates. Rather, the modes of the distribution are detected using non-maxima suppression, as per the standard Hough transform; here, a sample location, $\mathbf{y}$, is classified a mode if no other sample location, $\mathbf{z}$, within a certain distance, s.t. $K(\mathbf{y},\mathbf{z}) > \gamma$, has a higher probability. As a final step to improve accuracy, the location of each mode found is updated with one step of mean shift. The memory and computational requirements of this approach are $\mathrm{O}(n)$ and $\mathrm{O}(n^2d^2)$ respectively.

3.2 The maximum Hough transform

If, as in [3], we assume that only one vote per feature is correct, a simple way of exploiting this is to further assume that the correct vote is the one that best explains the location, $\mathbf{y}$, in question. This suggests a different formulation of equation (1), which we call the maximum Hough transform:

$$p_{\max}(\mathbf{y}|\mathbf{X},\boldsymbol{\omega},\theta) = \sum_{i=1}^{N} \omega_i \max_{j=1}^{J_i} \theta_{ij} K(\mathbf{x}_{ij},\mathbf{y}). \tag{3}$$

The process of detecting objects from this distribution is then identical to the standard or intrinsic Hough transforms. Since the method described below supersedes this method, we include it here purely for interest and completeness.

3.3 The minimum-entropy Hough transform

To better exploit the assumption that only one vote per feature is correct, a vote that is believed to be correct should explain away the other votes from that feature. This suggests that, rather than being given θ a priori, it would be beneficial to optimize over its possible values, giving those votes which agree with votes from other features more weight than those which do not.

One way of achieving this is by minimizing the information *entropy*[2] of $p(\mathbf{y}|\mathbf{X},\boldsymbol{\omega},\theta)$ w.r.t. θ. A similar approach, but minimizing entropy w.r.t. some parameters of the vote generation process, has already been used for lens distortion calibration [21]. A lower entropy distribution contains less information, making it more peaky and hence having more votes in agreement. Since information in Hough space is the location of objects, minimizing entropy constrains features to be generated by as few objects as possible. This can be viewed as enforcing Occam's razor.

Since computing entropy involves an integration over Hough space (here, very large), we use importance sampling [17, §29.2] to make this integration tractable; as with the intrinsic Hough transform, we sample the Hough space at the locations of all the votes. The value of θ is therefore given as

$$\theta = \operatorname*{argmin}_{\theta} -\sum_{i=1}^{N}\sum_{j=1}^{J_i} \frac{p(\mathbf{x}_{ij}|\mathbf{X},\boldsymbol{\omega},\theta)}{q(\mathbf{x}_{ij})} \ln p(\mathbf{x}_{ij}|\mathbf{X},\boldsymbol{\omega},\theta) \tag{4}$$

where $q(\cdot)$ is the (unknown) sampling distribution from which the votes are drawn. Once this optimization (described below) is done, the estimated θ is applied to equation (1), and

[2] Specifically we use the Shannon entropy [22], $H = E[-\ln p(x)] = -\int p(x)\ln p(x)\,\mathrm{d}x$.

inference continues as per the standard (or intrinsic) Hough transform. We call this approach the *minimum-entropy Hough transform*.[3]

3.3.1 Optimization framework

Since $p(\mathbf{y}|\mathbf{X},\boldsymbol{\omega},\Theta)$ is a linear function of Θ, and $-x\ln x$ is concave, as is a sum of concave functions, the cost function of equation (4) is concave. Its minimum therefore lies at an extremum of the parameter space, which is constrained by equation (2), such that the optimal value of $\Theta_i = \{\theta_{ij}\}_{j=1}^{J_i}$ (*i.e.* the vector of feature i's vote weights) must be an all 0 vector, except for one 1. The search space for each Θ_i is therefore a discrete set of J_i possible vectors, making the total number of possible solutions $\prod_{i=1}^{N} J_i$. It should be noted that this search space is not uni-modal—for example, if there are only two features and they each identically generate two votes, one for location $\mathbf{y}$ and one for location $\mathbf{z}$, then both $\mathbf{y}$ and $\mathbf{z}$ will be modes. Furthermore, as the search space is exponential in the number of features, an exhaustive search is infeasible for all but the smallest problems.

We therefore use a local approach, *iterated conditional modes* (ICM) [5], to quickly find a local minimum of this optimization problem. This involves updating the vote weights of each feature in turn, by minimizing equation (4) conditioned on the current weights of all other votes, and repeating this process until convergence. The correct update equation for the vote weights of a feature f is as follows:

$$p_{fk}(\mathbf{y}|\mathbf{X},\boldsymbol{\omega},\Theta) = \omega_f K(\mathbf{x}_{fk},\mathbf{y}) + \sum_{\forall i\neq f} \omega_i \sum_{j=1}^{J_i} \theta_{ij} K(\mathbf{x}_{ij},\mathbf{y}), \tag{5}$$

$$k = \operatorname*{argmin}_{k=1}^{J_f} -\sum_{i=1}^{N}\sum_{j=1}^{J_i} \frac{p_{fk}(\mathbf{x}_{ij}|\mathbf{X},\boldsymbol{\omega},\Theta)}{q(\mathbf{x}_{ij})} \ln p_{fk}(\mathbf{x}_{ij}|\mathbf{X},\boldsymbol{\omega},\Theta), \tag{6}$$

$$\theta_{fk} = 1, \quad \theta_{fj} = 0, \quad \forall j \neq k. \tag{7}$$

However, since this update not only involves $q(\cdot)$, which is unknown, but is also relatively costly to compute, we replace it with a simpler proxy which in practice performs a similar job:

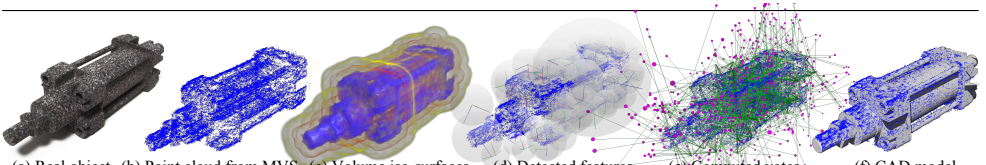
$$k = \operatorname*{argmax}_{k=1}^{J_f} p_{fk}(\mathbf{x}_{fk}|\mathbf{X},\boldsymbol{\omega},\Theta). \tag{8}$$

Since the optimization is local, a good initialization of Θ is key to reaching a good minimum. In our experiments we started at the value of Θ used in the standard Hough transform, then applied the following update to each vote weight simultaneously:

$$\theta_{ik} = \frac{p_{ik}(\mathbf{x}_{ik}|\mathbf{X},\boldsymbol{\omega},\Theta)}{\sum_{j=1}^{J_i} p_{ij}(\mathbf{x}_{ij}|\mathbf{X},\boldsymbol{\omega},\Theta)}, \tag{9}$$

iterating this five times before starting ICM. Initially updating weights softly, *i.e.* not fixing them to 0 or 1, and synchronously, avoiding ordering bias, in this way helped to avoid falling into a poor local minimum early on, thus improving the quality of solution found.

[3] Strictly speaking, the minimum-entropy Hough transform is not a transform, because the probability of each location in Hough space cannot be computed independently.



(a) Real object (b) Point cloud from MVS (c) Volume iso-surfaces (d) Detected features (e) Computed votes (f) CAD model

Figure 2: **The application:** 3D-shape-based object R&R (framework and figure from [20]). (a) Real object, fabricated from a CAD model. (b) Point cloud extracted using a multi-view stereo (MVS) system [26]. (c) Iso-surfaces of the scalar volume computed from the points. (d) Features (with position, scale and orientation) detected in the volume. (e) Votes for the object centre, based on detected features matched with a library of learnt features. (f) The registered CAD model.

4 Experiments

4.1 Setup

For our test application, 3D shape R&R, we use the same framework as [20], outlined in figure 2. In addition, we use the same test data (the ten test object classes used are shown in figure 4), evaluation framework, density kernel,[4] $K(\cdot,\cdot)$, and bandwidth parameters. Briefly, experiments are carried out on 1000 point clouds, each containing one object, with ground truth class and pose. Further details can be found in [20]; the evaluation data can be found online [1].

4.2 Methods

As well as evaluating the relative performance of the three Hough transforms introduced in §3.1, 3.2 & 3.3, we compare them with the SRT mean shift method of [20] (henceforth referred to as "mean shift"), and the inference method of Barinova *et al*. [3] (here referred to as BLK, after the authors, for short). Apart from mean shift, the methods all use the intrinsic Hough transform to make sampling $\mathcal{H}$ feasible. For the mean shift refinement step of the intrinsic Hough transform, we use the closed-form mean given in [20], despite our slightly different density kernel. However, we do not refine the detections of BLK because their probability distribution is not amenable to this, since it does not sum the votes. The likelihood function used in our implementation of BLK is the same kernel density function used in our three new methods. We note that the parameters of this kernel were learned in [20] specifically for Hough-based inference, therefore might not be optimal for BLK. Parameter values used for the various methods are summarized in table 1.

4.3 Results

4.3.1 Quantitative results

Quantitative results mirroring the analysis of [20] are given in tables 2 & 3 and figure 3. There is a small increase in performance in both registration and recognition moving from mean shift to intrinsic Hough, which is most likely due to modes being missed by mean shift. There is a larger increase in performance moving to maximum Hough, which gets the best registration score in 4/10 classes as well as a 17% jump in recognition rate (over mean

[4] In fact our density kernel is slightly different: we use a symmetric version of $d_{\mathrm{SRT}}(\mathbf{X},\mathbf{Y})$ (their notation), changing the denominator of [20, equation (7)] to $\sqrt{s(\mathbf{X})s(\mathbf{Y})}$. $K(\mathbf{y},\mathbf{z}) = 0$ if $\mathbf{y}$ and $\mathbf{z}$ specify different classes.

σ_s	σ_r	σ_t	γ	λ
a–e	a–e	a–e	b–d	e
0.0694	0.12	0.12	$\exp(-8)$	10

Table 1: **Parameter values** for the inference methods tested: (a) mean shift, (b) intrinsic Hough, (c) max. Hough, (d) minimum-entropy Hough, (e) the method of Barinova *et al.* [3].

shift), indicating that, in this application, several votes from the same feature can vote for the same location, generating false detections. Minimum-entropy Hough performs the best, showing a significantly improved registration rate, with top scores on 5/10 classes, and a hugely improved recognition rate over mean shift (a 96% reduction in misclassifications); only 1.5% of objects are left unrecognized, the majority of those in the car class. The BLK method has a similarly impressive recognition rate, but slightly lower registration rate. To see if this was due to the mean shift refinement of the other methods, we refined the modes of the BLK method by switching to summing the votes and doing one step of mean shift; the results (BLK + MS in table 2) show an improvement in registration rate, though still short of that of minimum-entropy Hough, as well a small increase in recognition rate.

However, because these results only reflect the best detection per test, they do not tell the whole story; we do not know how many other (incorrect) detections had competitive weights. To see this, we generated the precision-recall curves shown in figure 5, by varying the detection threshold, τ (or λ for BLK [3]). A correct detection in this test required the class *and* pose to be correct simultaneously, and allowed only one correct detection per test. The curves show that precision remains high as recall increases for the minimum-entropy Hough transform and BLK method (more so with our method), which are both able to explain away incorrect votes, while it drops off rapidly with recall for the other methods, indicating that the latter methods suffer from greater ambiguity as to which modes correspond to real objects. Interestingly, the minimum-entropy Hough transform has the lowest maximum recall rate (around 0.76), which we propose is due to some correct modes being "explained away" in the optimization of equation (4). Since our optimization strategy finds only a local minimum we cannot be sure whether this effect is due to the objective function or the optimization strategy.

In terms of computation time (table 2), all methods tested were of a similar speed. However, we noticed that the speed of BLK was dependent on the value of λ, its detection threshold, and therefore equally the number of objects in the scene, unlike the other methods.

4.3.2 Qualitative results

The benefit of explaining away incorrect votes is demonstrated in figure 1. While the standard Hough transform shows a great deal of ambiguity as to where and how many objects there are, which appears to get worse with the maximum Hough transform, the minimum-entropy Hough transform is able to clear away the "mist" of incorrect votes, leaving four distinct modes corresponding to the objects present; there are some other modes, but these are much less significant, corroborating the results seen in figure 5.

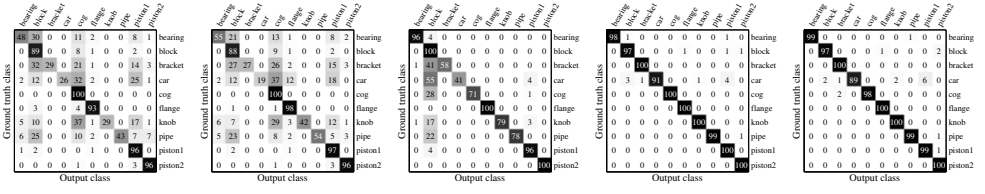
The benefit of having correct and clearly defined modes is demonstrated in figure 6, using the same point cloud as in figure 1, a challenging dataset containing three pairs of touching objects. Both minimum-entropy Hough and BLK find all six objects in the top six detections (though both mis-register the piston lying at a shallow angle), whereas the other methods find not only incorrect objects, but also multiple instances of correct objects (particularly the piston on the cog).

	Mean shift	Intrinsic H.	Max. H.	Min.-ent. H.	BLK	BLK + MS
Recog.	64.9%	67.6%	81.9%	**98.5%**	98.1%	98.3%
Regist.	68.3%	68.2%	72.5%	**74.6%**	72.5%	73.1%
Time	1.62s	**1.53s**	1.57s	1.59s	1.62s	1.65s

Table 2: **Quantitative results** for the inference methods tested.

	bearing	block	bracket	car	cog	flange	knob	pipe	piston1	piston2
Mean shift	77	13	95	75	**100**	**41**	88	86	44	64
Intrinsic Hough	77	15	96	76	**100**	35	86	86	44	67
Max. Hough	80	**21**	**100**	79	**100**	34	**91**	87	53	80
Minimum-entropy Hough	**83**	20	98	**91**	**100**	36	**91**	89	**54**	84
Barinova *et al.* (BLK) [3]	81	**21**	97	**91**	**100**	34	77	**90**	48	**86**

Table 3: **Registration rate** per class (%) for the five inference methods tested.



(a) Mean shift (b) Intrinsic H. (c) Max. H. (d) Min.-ent. H. (e) BLK

Figure 3: **Confusion matrices** for the five inference methods tested.

Figure 4: **Test objects.** CAD models for the 10 test object classes.

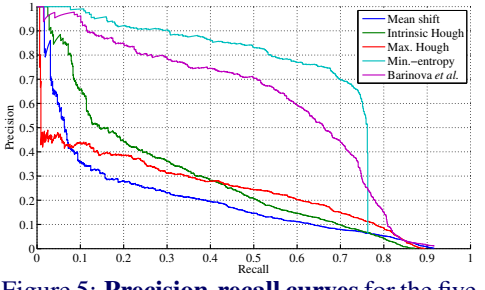
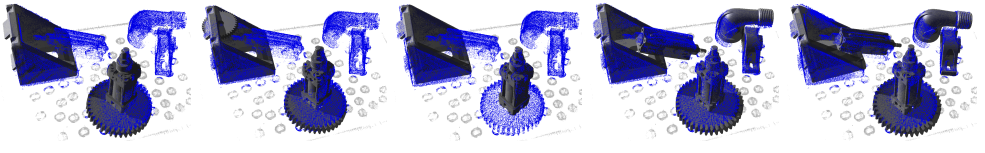


Figure 5: **Precision-recall curves** for the five inference methods tested.



(a) Mean shift (b) Intrinsic H. (c) Max. H. (d) Min.-ent. H. (e) BLK

Figure 6: **Qualitative results** for the five inference methods tested, showing the first 6 objects (in order of decreasing weight) detected by each method in a point cloud containing 6 objects. Only minimum-entropy Hough (d) and the method of Barinova *et al.* [3] (e) find all the objects.

5 Conclusion

We have introduced two key extensions of the Hough transform, which can be applied to any approach using the Hough transform. The first, the intrinsic Hough transform, changes the memory requirements of the Hough transform from $O(k^d),\ (k > 1)$ to $O(n)$, making it feasible for high-dimensional Hough spaces such as that of our 3D shape R&R application. The second, the minimum-entropy Hough transform, was shown to significantly increase detection precision over mean shift on our task. We also showed that it marginally outperformed the probabilistic method of Barinova *et al*. [3], as well as benefitting from a computation time that is independent of the number of objects in the scene, and allowing the straighforward refinement of modes using mean shift. However, since the kernel density parameters provided were optimized for Hough-based approaches and not for BLK, the real "take home" message of this paper is that the assumption that only one vote generated by each feature is correct is a powerful constraint in vote-based frameworks, which can dramatically improve inference by "clearing the mist" of incorrect votes.

References

[1] Toshiba CAD model point clouds dataset. http://www.toshiba-europe.com/research/crl/cvg/stereo_points.html.

[2] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[3] O. Barinova, V. Lempitsky, and P. Kohli. On detection of multiple object instances using Hough transforms. In *Proceedings of CVPR*, 2010.

[4] D. Ben-Tzvi and M. B. Sandler. A combinatorial Hough transform. *Pattern Recognition Letters*, 11(3):167–174, 1990.

[5] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, 48(3):259–302, 1986.

[6] Y. Cheng. Mean shift, mode seeking, and clustering. *TPAMI*, 17(8):790–799, 1995.

[7] R. O. Duda and P. E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15:11–15, 1972.

[8] A. Fisher, R. B. Fisher, C. Robertson, and N. Werghi. Finding surface correspondence for object recognition and registration using pairwise geometric histograms. In *Proceedings of ECCV*, pages 674–686, 1998.

[9] J. Gall and V. Lempitsky. Class-specific Hough forests for object detection. In *Proceedings of CVPR*, pages 1022–1029, 2009.

[10] P. V. C. Hough. Method and means for recognizing complex patterns. U.S. Patent 3,069,654, Dec. 1962.

[11] J. Illingworth and J. Kittler. The adaptive Hough transform. *TPAMI*, 9(5):690–698, 1987.

[12] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool. Hough transform and 3D SURF for robust three dimensional classification. In *Proceedings of ECCV*, pages 589–602, 2010.

[13] Y. Lamdan and H. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proceedings of ICCV*, pages 238–249, 1988.

[14] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV Workshop on Statistical Learning in Computer Vision*, 2004.

[15] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1-3):259–289, 2008.

[16] H. Li, M. A. Lavin, and R. J. Le Master. Fast Hough transform: A hierarchical approach. *Computer Vision, Graphics, and Image Processing*, 36(2-3):139–161, 1986.

[17] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[18] S. Maji and J. Malik. Object detection using a max-margin Hough transform. In *Proceedings of CVPR*, 2009.

[19] R. Okada. Discriminative generalized Hough transform for object detection. In *Proceedings of ICCV*, pages 2000–2005, 2009.

[20] M.-T. Pham, O. J. Woodford, F. Perbet, A. Maki, B. Stenger, and R. Cipolla. A new distance for scale-invariant 3D shape recognition and registration. In *Proceedings of ICCV*, 2011.

[21] E. Rosten and R. Loveland. Camera distortion self-calibration using the plumb-line constraint and minimal Hough entropy. *Machine Vision and Applications*, October 2009.

[22] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[23] Y. A. Sheikh, E. A. Khan, and T. Kanade. Mode-seeking by medoidshifts. In *Proceedings of ICCV*, 2007.

[24] F. Tombari and L. Di Stefano. Object recognition in 3D scenes with occlusions and clutter by Hough voting. In *Proceedings of PSIVT*, pages 349–355, 2010.

[25] A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking. In *Proceedings of ECCV*, pages 705–718, 2008.

[26] G. Vogiatzis and C. Hernández. Video-based, real-time multi view stereo. *Image and Vision Computing*, 29(7):434–441, 2011.

[27] L. Xu, E. Oja, and P. Kultanen. A new curve detection method: Randomized Hough transform (RHT). *Pattern Recognition Letters*, 11(5):331–338, 1990.