Monte Carlo Pose Estimation with Quaternion Kernels and the Bingham Distribution

Jared Glover
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
jglov@mit.edu

Gary Bradski and Radu Bogdan Rusu
Willow Garage
Menlo Park, CA 94025
{bradski,rusu}@willowgarage.com

***Abstract*—The success of personal service robotics hinges upon reliable manipulation of everyday household objects, such as dishes, bottles, containers, and furniture. In order to accurately manipulate such objects, robots need to know objects' full 6-DOF pose, which is made difficult by clutter and occlusions. Many household objects have regular structure that can be used to effectively guess object pose given an observation of just a small patch on the object. In this paper, we present a new method to model the spatial distribution of oriented local features on an object, which we use to infer object pose given small sets of observed local features. The orientation distribution for local features is given by a mixture of Binghams on the hypersphere of unit quaternions, while the local feature distribution for position given orientation is given by a locally-weighted (Quaternion kernel) likelihood. Experiments on 3D point cloud data of cluttered and uncluttered scenes generated from a structured light stereo image sensor validate our approach.**

I. Introduction

The goal of this paper is to determine a set of possible object poses, given a 3D object point cloud and a point cloud observation of a scene containing the object. To answer this question, we will consider the observed point cloud to be made up of many tiny overlapping surface patches, and we will construct a model of the information each observed patch gives us about the object's pose, by considering the range of locations the patch may have come from on the model. When such a surface patch contains orientation information in the form of a normal vector and principal curvature direction (or when the normal and principal curvature can be estimated from the local patch geometry), we call it an *oriented local feature* (or "oriented feature" for short).

In this paper, we present a new method to model the spatial distribution of oriented local features on an object, which we use to infer object pose given small sets of observed features. We split up the spatial distribution of oriented local features into two parts, one modeling the distribution over feature orientations, and another modeling the conditional distribution of feature position given orientation. Splitting up the feature distributions in this way allows us to exploit predictable relationships between the orientation of local surface patches on an object and the object's pose (Figure 1).

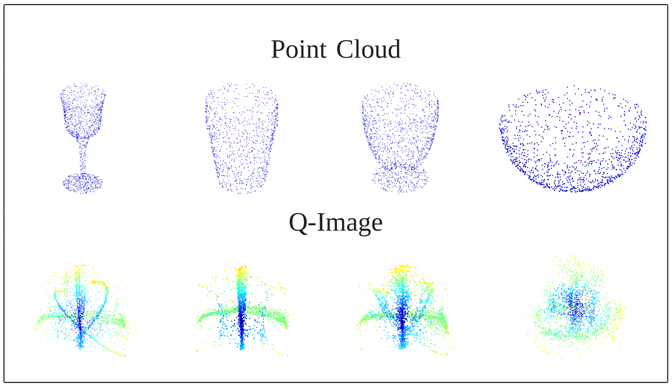


Figure 1. Four 3-D point clouds and their corresponding Q-Image transforms, respresenting the distribution of local 3-D surface orientations on each object (see section III for details).

The distribution for feature orientation is given by a mixture of Binghams on the hypersphere of unit quaternions. The Bingham distribution [3] is an antipodally symmetric probability distribution on a unit hypersphere. It can be used to represent many types of uncertainty, from highly peaked distributions to distributions which are symmetric about some axis to the uniform distribution. It is thus an ideal distribution for 3D rotations, which can be modeled as unit quaternions on the 4-D hypersphere, $\mathbb{S}^3$.

The distribution for feature position given orientation is given by a locally-weighted (Quaternion kernel) likelihood. Local likelihood is a non-parametric, kernel-based technique for modeling a conditional distribution, $p(\mathbf{x}|\mathbf{q})$, as a smoothly varying function of the observed variable, $\mathbf{q}$.

A. Outline

The technical portion of this paper is organized in a top-down fashion. We present our pose estimation algorithms in section II. Our main contribution—a new way to model the distribution of oriented local features on an object—is given in section III. Section IV reviews the Bingham distribution and introduces its mixture model, the BMM, which we use to represent uncertainty over the space of 3-D rotations. Section V contains experimental results, followed by related work and the conclusion in sections VI and VII.

II. Monte Carlo Pose Estimation

At its core, this paper presents a new way to model the spatial relationship between an object and its oriented local

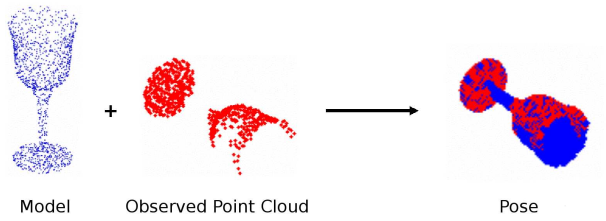


Figure 2. The Single Object Pose Estimation (SOPE) problem: given a model and a (partial) observation (left), we wish to estimate the pose of the object (right).

```
• Given: a model, $M$, and an observed point cloud, $F_{obs}$.
• For $i = 1 \ldots N$
    – 1) Sample a "proposal" oriented feature, $f_p$, at random from $F_{obs}$.
    – 2) Sample an object pose, $(\mathbf{x_i}, \mathbf{q_i})$ from $p_M(\mathbf{x}, \mathbf{q} | f_p)$.
    – 3) Sample $k$ "validation" oriented features, $\{f_{v_{1\ldots k}}\}$ at random from $F_{obs}$.
    – 4) Set sample weight $w_i \leftarrow p_M(\mathbf{x_i}, \mathbf{q_i} | \{f_{v_{1\ldots k}}\})$.
• Return: the top $n$ samples $(\mathbf{x_i}, \mathbf{q_i})$ ranked by weight, $w_i$.
```

Table I
MC-SOPE

features. This model can be used for many perceptual tasks, since there are many uses for such spatial models in a perceptual processing pipeline. As an application, we test the acuity of our model for object pose estimation, in a probabilistic Random-Sampling-Consensus (RANSAC) framework.

RANSAC [7] is a classic Monte-Carlo algorithm for using small random subsets from a large set of features to quickly generate many guesses of whatever the algorithm is trying to estimate. It then ranks the guesses according to an evaluation criterion, and returns the top n answers. RANSAC is part of a larger trend to solve perceptual problems by applying successive filters, or sieves, to a set of guesses (or samples), keeping good samples and throwing away the bad ones, until (one hopes) only good samples are left. The key to making such a filtering scheme work efficiently is to generate reasonable guesses early on in the pipeline, and throw away as many bad samples with the early, fast filters, so that the later, discriminative (but slow) filters don't have to sift through as much junk to find the right answers.

Our model fits perfectly at the early stage of such a filtering scheme for pose estimation. By squeezing all the information one can from a single oriented local feature, and then from small sets of oriented local features, we can quickly generate many good guesses for object pose. Later filters—which for example iteratively align a model at a sample pose with the observed point cloud and then compute a fitness score using all the points in the model—can be used to further refine the set of pose estimates, and are well explored in the literature. However, that final alignment stage is omitted in this work in order to focus on our primary contribution—generating good initial sample poses using just a few local oriented features at a time, which is crucial for pose estimation in cluttered scenes, where only a small portion of an object may be visible.

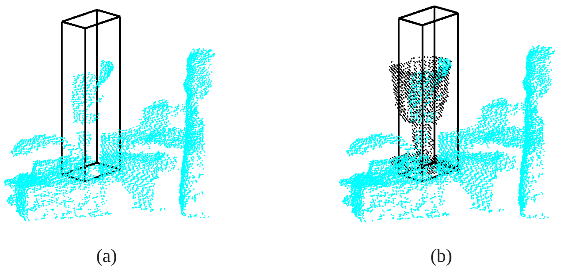


Figure 3. The Single Cluttered Object Pose Estimation (SCOPE) problem: given a model, an observation, and a region of interest (a), we wish to estimate the pose of the object (b).

```
• Given: a model, $M$, and an observed point cloud, $F_{obs}$ with sub-cloud region of interest, $F_{obs,roi}$.
• For $i = 1 \ldots N$
    – 1) Sample a "proposal" oriented feature, $f_p$, at random from sub-cloud $F_{obs,roi}$.
    – 2) Sample an object pose, $(\mathbf{x_i}, \mathbf{q_i})$ from $p_M(\mathbf{x}, \mathbf{q} | f_p)$.
    – 3) Sample $k$ "validation" features, $\{f_{v_{1\ldots k}}\}$ using a random walk in the full point cloud $F_{obs}$, starting from $f_p$.
    – 4) Set sample weight $w_i \leftarrow p_M(\mathbf{x_i}, \mathbf{q_i} | \{f_{v_{1\ldots k}}\})$.
• Return: the top $n$ samples $(\mathbf{x_i}, \mathbf{q_i})$ ranked by weight, $w_i$.
```

Table II
MC-SCOPE

A. *Problem Statement*

There are two primary pose estimation problems we address in this paper. The first problem, which we call "Single Object Pose Estimation" (SOPE), is to estimate an object's pose given a point cloud observation, F_{obs}, of part of the object (Figure 2). For this task, we have a model (i.e. we know the object's identity) and a segmentation (i.e. we know which points in the observation correspond to the model), and the goal is to return samples from the distribution over possible object poses $(\mathbf{x}, \mathbf{q}) \in \mathbb{R}^3 \times \mathbb{S}^3$ given the observation, $p(\mathbf{x}, \mathbf{q} | F_{obs})$. The SOPE problem is well suited to "separable" scenes, for example when one or more objects are separated on a table or a shelf.

The second problem is "Single Cluttered Object Pose Estimation" (SCOPE). In this task, we have a model, an observed point cloud containing several objects, and a rough location (e.g. a bounding box) in which to look for the desired object. The goal once again is to return a set of samples from the posterior pose distribution, $p(\mathbf{x}, \mathbf{q} | F_{obs})$, of the desired object.

B. *MC-SOPE Algorithm*

Our solution to the SOPE problem, MC-SOPE, is shown in table I. It starts by sampling a pose using the information from one "proposal" oriented local feature f_p which is chosen at random from the observed point cloud, F_{obs}. Then, it uses k "validation" features, $f_{v_1}, \ldots, f_{v_k}$, to compute a weight/score for the proposed object pose. After N poses have been proposed and weighted, the algorithm returns the top n samples, ranked by weight. As we will see in section III, sampling from $p_M(\mathbf{x}, \mathbf{q} | f_p)$ and computing the validation density $p_M(\mathbf{x}, \mathbf{q} | \{f_{v_{1\ldots k}}\})$ are both $O(1)$ operations when k

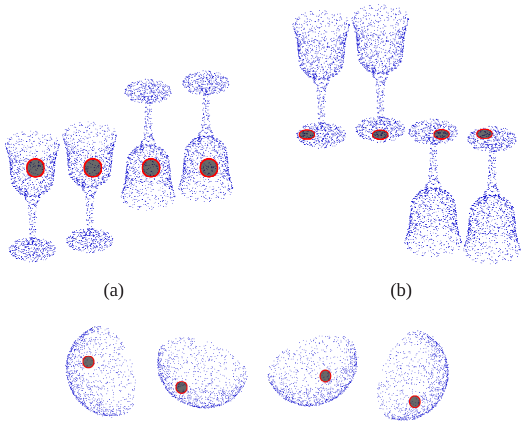


Figure 4. Informative oriented local (surface patch) features. Observing just a single patch of an object can often constrain the set of feasible object poses to a narrow range of possibilities. In the top row, two different patches on a wine glass are shown, along with four possible object poses consistent with each patch. In the bottom row, a bowl feature is shown, along with four object poses consistent with the observed patch. Both of the the observed glass features constrain the orientation of the glass to be aligned with the z-axis, while the bowl feature places much weaker constraints on orientation.

is a small constant (we use $k = 5$ in our experiments), so the entire algorithm is $O(N)$, allowing us to generate large numbers of good samples quickly. This is in contrast to traditional RANSAC-type methods for model fitting and pose estimation, which typically use *all* the observed features in F_{obs} to rank each proposal, resulting in a running time of $O(N \cdot |F_{obs}|)$.

C. *MC-SCOPE Algorithm*

Our solution to the SCOPE problem, MC-SCOPE, is shown in table II. It is nearly identical to MC-SOPE, with the main difference being that MC-SCOPE samples validation points using a random walk (since the observed point cloud contains more than one object and the region of interest, $F_{obs,roi}$ may not be a perfect segmentation of the desired object). The random walk is performed by taking a random step according to an isotropic normal distribution with stdev. ϵ (3cm in all our experiments), then finding the closest point in the point cloud to the new location.

III. Local Feature Distributions

Both pose estimation algorithms in the previous section require a model for $p_M(\mathbf{x}, \mathbf{q}|f_p)$ and $p_M(\mathbf{x}, \mathbf{q}|\{f_{v_{1...k}}\})$—the likelihood of an object pose, given one or more observed oriented local features. Although the oriented local features on an object may not be unique, they can still be quite informative. For example, in figure 4, observing a curved patch of the surface on the cup or a flat patch on the base tells us that the object is aligned with the z-axis, at a narrow range of positions. In contrast, oriented local features on the surface of a hemispherical bowl tell us very little about the bowl's orientation, and constrain the bowl's position to a bowl-shaped set of locations, centered on the observed patch.

Our approach to computing $p_M(\mathbf{x}, \mathbf{q}|f_i)$ will be to flip it around using Bayes' rule,

$$p_M(\mathbf{x}, \mathbf{q}|f_i) \propto p_M(f_i|\mathbf{x}, \mathbf{q}) p_M(\mathbf{x}, \mathbf{q})$$

where $p_M(\mathbf{x}, \mathbf{q})$ is a prior probability and $p_M(f_i|\mathbf{x}, \mathbf{q})$ is the likelihood of observing f_i given the object pose $(\mathbf{x}, \mathbf{q})$. We will gain further purchase on $p_M(f_i|\mathbf{x}, \mathbf{q})$ by examining the components of an oriented local feature, f_i.

A. *Oriented Local Features*

We describe an oriented local feature f_i with three components—

1) a shape descriptor, $\mathbf{s_i}$,
2) a position, $\mathbf{x_i} \in \mathbb{R}^3$, and
3) a quaternion orientation, $\mathbf{q_i} \in \mathbb{S}^3$.

The shape descriptor $\mathbf{s_i}$ should be invariant to position and orientation, while describing the local surface geometry of the feature. After testing different 3D shape descriptors, we selected the Fast Point Feature Histogram (FPFH)[18], due to its favorable performance in the presence of noise and missing data. The position $\mathbf{x_i}$ and orientation $\mathbf{q_i}$ are relative to a fixed model coordinate frame; they describe the rigid body transform which maps model coordinates to local feature coordinates. Unit quaternions are used to describe rotations because they avoid the topological degeneracies of other representations, and fit perfectly into the Bingham mixture distributions we develop in this work.

In this paper, an oriented local feature is a descriptor for the pose and shape of a local surface patch on a 3D point cloud. Given a surface patch (i.e. a local set of 3D points $\mathbf{P}$ on a model M) with estimated surface normals $\mathbf{U}$ at every point, we compute the transform $(\mathbf{x_i}, \mathbf{q_i})$ by picking an arbitrary (fixed) model coordinate frame, and estimating the local coordinate frame whose origin is at the center of patch $\mathbf{P}$, with axes given by $\mathbf{u}$, the estimated surface normal at the center of patch P, $\mathbf{v}$, the direction of maximal surface curvature[1], and $\mathbf{w} = \mathbf{u} \times \mathbf{v}$. Then $(\mathbf{x_i}, \mathbf{q_i})$ is the transform which takes the model frame to the local frame. An example of a local coordinate frame is shown on the left side of figure 6.

B. *Local Feature Likelihood*

The next step in modeling $p_M(f_i|\mathbf{x}, \mathbf{q})$ is generate a "vocabulary" Ω of local shape types for the given model. We do this using K-Means clustering (with $|\Omega| = 10$ in this paper) on the set of all local shape descriptors, $\{\mathbf{s_i}\}$ in the model point cloud, yielding a shape vocabulary, $\{\Omega_j\}$. We then segment the model point cloud into $|\Omega|$ clusters, where cluster j contains all the oriented local features in the model whose closest vocabulary shape is Ω_j. Examples of three feature clusters on a wine glass, with their local coordinate frames, is shown in the top row of figure 5.

For a newly observed oriented local feature, f_i, we compute the local feature likelihood, $p_M(f_i|\mathbf{x}, \mathbf{q})$ by expanding f_i into

[1] The direction of maximal surface curvature, or *principal curvature*, is estimated using principal components analysis on the neighborhood of local surface patch normals, projected into the tangent space to the central normal vector; the principal curvature is the eigenvector corresponding to the largest eigenvalue, and is only defined up to sign.

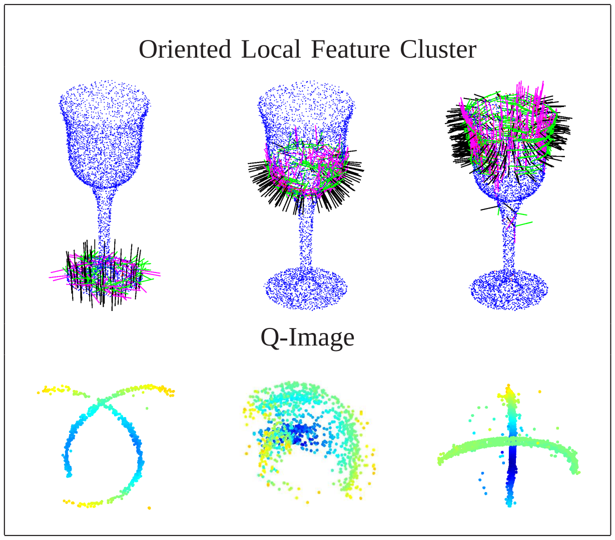


Figure 5. Cluster Q-Images.

components $(\mathbf{s_i}, \mathbf{x_i}, \mathbf{q_i})$, and classifying $\mathbf{s_i}$ into one of the shape vocabulary clusters, Ω_j, yielding

$$\begin{aligned} & p_M(f_i|\mathbf{x},\mathbf{q}) \\ & \quad \propto p_M(\mathbf{x_i},\mathbf{q_i}|\mathbf{x},\mathbf{q},\Omega_j)p_M(\Omega_j|\mathbf{x},\mathbf{q}) \\ & \quad = p_M(\mathbf{q_i}|\mathbf{x},\mathbf{q},\Omega_j)p_M(\mathbf{x_i}|\mathbf{q_i},\mathbf{x},\mathbf{q},\Omega_j)p_M(\Omega_j|\mathbf{x},\mathbf{q}) \end{aligned} \tag{1}$$

where we can drop the dependence on $\mathbf{x}$ in the first term (since the orientation of a local feature on an object shouldn't depend on the object's position in space), and where $p_M(\Omega_j|\mathbf{x},\mathbf{q})$ is assumed to be uniform.[2] Thus,

$$p_M(f_i|\mathbf{x},\mathbf{q}) = p_M(\mathbf{q_i}|\mathbf{q},\Omega_j)p_M(\mathbf{x_i}|\mathbf{q_i},\mathbf{x},\mathbf{q},\Omega_j). \tag{2}$$

To model $p_M(\mathbf{q_i}|\mathbf{q},\Omega_j)$, we consider the set of all local feature orientations on the model from feature cluster Ω_j, which we visualize using the "Q-Image" transform[3] for three features types in the bottom row of figure 5. We then fit a Bingham Mixture Model (BMM)—defined in section IV-D—to the set of all rotations mapping model axes into local axes, so that

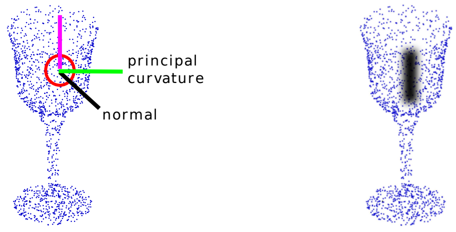
$$p_M(\mathbf{q_i}|\mathbf{q},\Omega_j) = p(\mathbf{q_i}\mathbf{q}^{-1};B_j) \tag{3}$$

where B_j are parameters of the BMM for feature cluster j.

To model $p_M(\mathbf{x_i}|\mathbf{q_i},\mathbf{x},\mathbf{q},\Omega_j)$, we use a non-parametric technique, called local likelihood, to fit a distribution to the set of all translations mapping model origin to rotated local feature origin; that is,

$$p_M(\mathbf{x_i}|\mathbf{q_i},\mathbf{x},\mathbf{q},\Omega_j) = p_M(\mathbf{q}^{-1}(\mathbf{x_i}-\mathbf{x})|\mathbf{q_i}\mathbf{q}^{-1},\Omega_j). \tag{4}$$

Local Likelihood is a technique introduced by Tibshirani and Hastie [21] to model the parameters of a conditional probability distribution, $p(Y|X)$, as a smoothly varying function of X. In our case, we wish to model the conditional local feature pose distribution $p(\mathbf{x}|\mathbf{q})$ within a particular feature class, Ω_j (where $\mathbf{x}|\mathbf{q}$ is shorthand for $\mathbf{q}^{-1}(\mathbf{x_i}-\mathbf{x})|\mathbf{q_i}\mathbf{q}^{-1}$ in equation 4). To do so, we choose a particular parametric model for $p(\mathbf{x}|\mathbf{q}=\mathbf{q}')$ (multivariate normal distribution), with parameters (μ,Σ). To fit parameters, we use a locally-weighted data log-likelihood function, $\ell()$, with Gaussian kernel $K(\mathbf{q},\mathbf{q_i})$ to weight the contribution of each $(\mathbf{x_i}|\mathbf{q_i})$ in the model point cloud according to how close $\mathbf{q_i}$ is to $\mathbf{q}$,



(a) local feature coordinate frame

(b) possible feature positions

Figure 6. Given that an oriented local feature with shape type Ω_j has the coordinate frame shown in (a), the range of possible feature positions is restricted to lie on the dark band of the cup in (b).

$$\ell(\mu,\Sigma;\mathbf{q},X,Q) = \sum_{i=1}^{n} K(\mathbf{q},\mathbf{q_i})\ell(\mu,\Sigma;\mathbf{x_i}). \tag{5}$$

We use a local likelihood model in this work for its flexibility, since finding a good parametric model for $p(\mathbf{x}|\mathbf{q})$ which works for all $\mathbf{q}$'s is quite difficult. An example of $p(\mathbf{x}|\mathbf{q})$ for a particular $\mathbf{q}$ is shown on the right side of figure 6.

C. Multi-Feature Likelihood

Given multiple observed oriented local features, $\{\mathbf{f_i}\} = \{(\mathbf{s_i},\mathbf{x_i},\mathbf{q_i})\}$, $i = 1\ldots k$, the idea for the multi-feature likelihood model is that we consider the average feature log-likelihood to be a random variable drawn from an exponential distribution with parameter λ.

$$p(\{\mathbf{f_i}\}|\mathbf{x},\mathbf{q}) = \lambda \exp\left[\frac{\lambda}{k}\sum_{i=1}^{k}\log p(\mathbf{f_i}|\mathbf{x},\mathbf{q})\right] \tag{6}$$

If the local features were independent given the object pose, then λ would be equal to 1. However, since features may overlap, we smooth the log-likelihood contribution from each feature by setting $\lambda < 1$ (0.5 in all of our experiments).

IV. The Bingham Distribution

The Bingham distribution is an antipodally symmetric probability distribution on a unit hypersphere. Its probability density function (PDF) is

$$f(\mathbf{x};\Lambda,V) = \frac{1}{F}\exp\{\sum_{i=1}^{d}\lambda_i(\mathbf{v_i}^T\mathbf{x})^2\} \tag{7}$$

where $\mathbf{x}$ is a unit vector on the surface of the sphere $\mathbb{S}^d \subset \mathbb{R}^{d+1}$, F is a normalization constant, Λ is a vector of concentration parameters, and the columns of the $(d+1)\times d$ matrix V are orthogonal unit vectors. By convention, one

[2] $p_M(\Omega_j|\mathbf{x},\mathbf{q})$ is a view-dependent term; we would need to know the sensor pose in addition to the object pose in order to utilize view-based statistics. Adding this is a direction for future work.

[3] The *Q-Image* of a 3D point cloud is defined as the set of all quaternion local feature orientations; we visualize the Q-Image in 3D using the axis-angle format of 3D rotation.

typically defines Λ and V so that $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_d \leq 0$. Note that a large (negative) λ_i indicates that the distribution is highly peaked along the direction $\mathbf{v_i}$, while a small (negative) λ_i indicates that the distribution is spread out along $\mathbf{v_i}$.

The Bingham distribution is derived from a zero-mean Gaussian on $\mathbb{R}^{d+1}$, conditioned to lie on the surface of the unit hypersphere $\mathbb{S}^d$. Thus, the exponent of the Bingham PDF is the same as the exponent of a zero-mean Gaussian distribution (in principal components form, with one of the eigenvalues of the covariance matrix set to infinity).

The Bingham distribution is most commonly used to represent uncertainty in axial data on $\mathbb{S}^2$. In geology, it is often used to encode preferred orientations of minerals in rocks [14]. Higher dimensional, complex forms of the Bingham distribution are also used to represent the distribution over 2-D planar shapes [5]. In this work, we use the Bingham on $\mathbb{S}^3$ as a probability distribution over 3-D quaternion rotations. Since the unit quaternions $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation in 3-D space, the antipodal symmetry of the Bingham distribution correctly captures the topology of quaternion rotation space.

A. The Normalization Constant

The primary difficulty with using the Bingham distribution in practice lies in computing the normalization constant, F. Since the distribution must integrate to one over its domain ($\mathbb{S}^d$), we can write the normalization constant as

$$F(\Lambda) = \int_{x \in \mathbb{S}^d} \exp\{\sum_{i=1}^{d} \lambda_i (\mathbf{v_i}^T \mathbf{x})^2\} \qquad (8)$$

In general, there is no closed form for this integral, which means that F must be approximated. Typically, this is done via series expansion [3], [12], although saddle-point approximations [13] have also been used.

Following Bingham [3], we note that $F(\Lambda)$ is proportional to a hyper-geometric function of matrix argument, with series expansion

$$\begin{aligned} F(\Lambda) &= 2 \cdot {}_1F_1(\frac{1}{2}; \frac{d+1}{2}; \Lambda) = \\ & 2\sqrt{\pi} \sum_{\alpha_1,\ldots,\alpha_d=0}^{\infty} \frac{\Gamma(\alpha_1 + \frac{1}{2}) \cdots \Gamma(\alpha_d + \frac{1}{2})}{\Gamma(\alpha_1 + \cdots + \alpha_d + \frac{d+1}{2})} \cdot \frac{\lambda_1^{\alpha_1} \cdots \lambda_d^{\alpha_d}}{\alpha_1! \cdots \alpha_n!} \end{aligned} \qquad (9)$$

For practical usage, we precompute a lookup table of F-values over a discrete grid of Λ's, and use interpolation to quickly estimate normalizing constants on the fly.

B. Parameter Estimation

Following Bingham [3], we estimate the parameters V and Λ given a set of N samples, $\{\mathbf{x_i}\}$, using a maximum likelihood approach. Finding the maximum likelihood estimate (MLE) $\hat{V}$ is an eigenvalue problem—the MLE mode of the distribution is equal to the eigenvector of the scatter matrix $S = \frac{1}{N}\sum_i \mathbf{x_i}\mathbf{x_i^T}$ corresponding to the largest eigenvalue, while the columns of $\hat{V}$ are equal to the eigenvectors corresponding to the 2nd through $(d+1)$th eigenvalues of S.

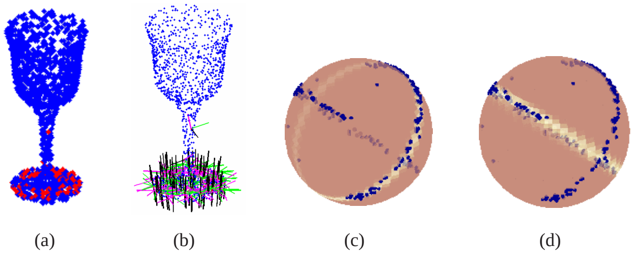


Figure 7. Fitting a Bingham Mixture Model (BMM). (a) A cluster of similar oriented local features. (b) The local coordinate frames of the oriented local features. (c-d) The local coordinate frames (only the axis part of the axis-angle format is visualized), along with the fitted Binghams (represented by the light bands on the two spheres).

The maximum likelihood estimate $\hat{\Lambda}$ is found by setting the partial derivatives of the data log likelihood function with respect to Λ to zero, yielding

$$\frac{1}{F(\Lambda)} \frac{\partial F(\Lambda)}{\partial \lambda_j} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{v_j}^T \mathbf{x_i})^2 = \mathbf{v_j}^T S \mathbf{v_j}, \qquad (10)$$

for $j = 1, \ldots, d$. Just as we did for $F(\Lambda)$, we can pre-compute values of the gradient of F with respect to Λ, ∇F, and store them in a lookup table. Using a kD-tree, we can find the nearest neighbors of a new sample $\nabla F / F$ in $O(d \log M)$ time (where M^d is the size of the lookup table), and use their indices to find Λ via interpolation (since the lookup tables for F and ∇F are indexed by Λ).

Notice that the maximum likelihood estimates for V and Λ are both computed given only the scatter matrix, S. Thus, S is a sufficient statistic for the Bingham distribution. In fact, there is a beautiful result from the theory of exponential families which says that the Bingham distribution is the *maximum entropy* distribution on the hypersphere which matches the sample inertia matrix (scatter matrix) $S = E[\mathbf{x}\mathbf{x}^T]$ [17]. This gives us further theoretical justification to use the Bingham distribution in practice, if we assume that all of the relevant information about the data is captured in the inertia matrix.

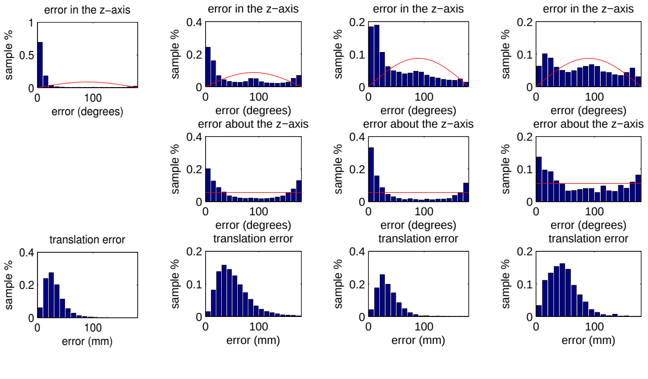
C. Sampling

Because of the complexity of the normalization constant, sampling from the Bingham distribution directly is difficult. Therefore, we use a Metropolis-Hastings sampler, with target distribution given by the Bingham density, and proposal distribution given by the projected zero-mean Gaussian[4] in $\mathbb{R}^{d+1}$ with covariance matrix equal to the Bingham's sample inertia matrix, S. Because the proposal distribution is very similar to the target distribution, the sampling distribution from the Metropolis-Hastings sampler converges to the true Bingham sampling distribution after just a few iterations.

D. Bingham Mixture Models (BMMs)

The Bingham distribution is a natural, maximum entropy model for second order distributions on a hypersphere, $\mathbb{S}^d$. However, the Q-Image in a local feature distribution model will almost always be explained best by a more complex

[4] To sample from the projected Gaussian, we first sample from a Gaussian with covariance S, then project the sample onto the unit sphere.



(a) IKEA (b) Household (c) PR2 (d) PR2, SCOPE

Figure 8. Sample error distributions (bar plots), compared to random guessing for orientation (solid red lines). (a-c) MC-SOPE/uncluttered, (d) MC-SCOPE/cluttered

distribution, as one can see in figures 1 and 5. For this purpose, we introduce the Bingham Mixture Model (BMM), with PDF

$$f_{BM}(x;\mathbf{B},\boldsymbol{\alpha}) = \sum_{i=1}^{k} \alpha_i f(x;B_i). \tag{11}$$

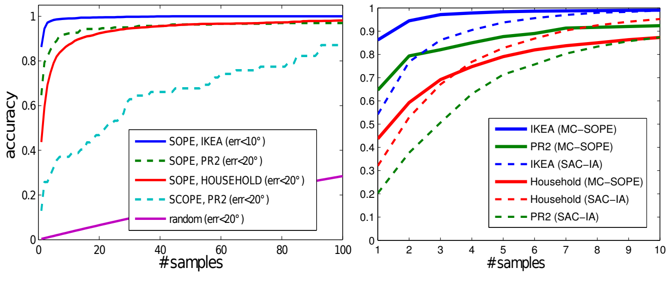
where the B_i's are the component Bingham parameters and the α_i's are weights.

E. BMM Parameter Estimation via Sample Consensus

To fit a mixture model to a set of data points we must estimate the cluster parameters, θ_i, the cluster weights, α_i, and the number of clusters, k. We have found iterative parameter estimation—the standard technique for fitting mixture models—for BMMs to be highly susceptible to getting stuck in local minima[5], so instead we use a greedy algorithm based on the sample consensus framework which works well in practice, but for which there are unfortunately very few theoretical guarantees. The BMM sample-consensus (BMM-SAC) algorithm starts by fitting M Binghams to M random sets of 4 points each. The Bingham B^* which fits the data best under a capped loss function (where outliers all contribute a minimum likelihood, ℓ_{min}, so as not to give them too much influence on the total fitness) is added to the mixture if it is a good enough fit (i.e. has data likelihood $> \ell_{thresh}$). Then, the inlier points with respect to B^* (those points with likelihood greater than ℓ_{min}) are removed from the training set, and the algorithm searches for another Bingham to add to the mixture. When no further Binghams can be found, the remaining points are considered outliers, and a uniform mixture component is added[6]. A reasonable choice for the outlier threshold, ℓ_{min}, is given by the uniform probability density,

$$\ell_{min} = p_{unif}(\mathbf{x}) = 1/A(d), \tag{12}$$

where $A(d)$ is the surface area of the hypersphere $\mathbb{S}^d$.



(a) (b)

Figure 9. (a) Search orientation accuracy of MC-SOPE and MC-SCOPE on each dataset. Each point, (X, Y), in the figure shows the percentage Y of trials for which a "good" pose sample was found in the top X samples. (b) Comparison to the proposal sample distribution of SAC-IA (orientation error thresholds are the same as in (a)).

V. Experimental Results

We tested our pose estimation algorithms on three different datasets—one containing 41 rotationally-symmetric IKEA objects (mostly dishes), one containing 29 rotationally-asymmetric household items, such as soap bottles, tools, and food containers, and one containing 8 manipulable objects for a PR2 robot. Each dataset consisted of a set of 3D point clouds which were scanned in from real objects; however, novel point cloud views of objects (for both training and testing) were generated in three different ways. For the IKEA dataset, we used ray-tracing with additive Gaussian noise to generate 41 views of each object. For the Household dataset, we used a 3-D simulator to emulate output from a texture-projection stereo camera pair. Then we ran a stereo vision algorithm on the simulated images to estimate pixel depth, and extracted point clouds from the rectified depth images for 71 views per object. For the PR2 dataset, we captured point clouds from the Willow Garage PR2 robot's projected light stereo cameras. To generate 3D models, we drove the robot around each object, placed in varying orientations on a table, collecting 34-40 views of each object; point clouds from all the views were then aligned using an occupancy-grid approach. The PR2 dataset also contained 11 point clouds of cluttered scenes with varying configurations of the 8 objects on a tabletop in front of the robot; we painstakingly hand-labelled every scene with each object's true pose for testing purposes.

To fit local feature models for the objects in each dataset, we calculated normals, local shape descriptors (FPFHs), and principal curvatures for the points in each point cloud. After downsampling to reduce the number of oriented local features in each model to around 2000 points, we clustered features across all views by their FPFH using KMeans with $k = 10$. We then fit a Bingham Mixture Model (BMM) to the local coordinate frames (Q-Image) in each cluster. For the purposes of testing the MC-SOPE algorithm, we used cross-validation with 5 training and testing set pairs. For each testing set, we used the corresponding (non-overlapping) training set to learn FPFH clusters and BMMs. We then ran MC-SOPE on each point cloud with $N = 1000$ proposal samples in the testing sets, returning the top $n = 100$ pose samples for each view.

We ran the MC-SCOPE algorithm on the objects in each

[5] This is probably due to the great flexibility of the Bingham distribution, and the non-Euclidean nature of the hypersphere.

[6] Recall that a Bingham with all its concentration parameters λ_i set to zero is a uniform distribution on $\mathbb{S}^d$; thus, the uniform component of a BMM is also a Bingham distribution.

scene of the cluttered PR2 dataset with $N = 5000$ proposal samples, returning the top $n = 100$ pose samples for each object. In figure 10, we show examples of top pose samples for one of the scenes.

In figure 8, average error histograms across all MC-SOPE tests are shown for the three datasets, and for the cluttered PR2 dataset for the MC-SCOPE trials. For the IKEA dataset, we show orientation error in the z-axis, since the IKEA objects are all symmetric about the z-axis. For the Household and PR2 datasets, we show orientation error *in* the z-axis, and also *about* the z-axis for samples with z-axis error < 20 degrees. Error in translation (XYZ) for samples with z-axis error < 20 degrees is shown for all datasets.

In figure 9, we show the search orientation accuracy of MC-SOPE and MC-SCOPE on each dataset. Each point, (X, Y), in figure 9 shows the percentage Y of trials for which a "good" pose sample was found in the top X samples. A "good" sample is defined as z-axis error < 10 degrees for the IKEA dataset, and both z-axis error < 20 degrees and about-z-axis error < 20 degrees for the Household and PR2 datasets. Position accuracy is not measured in figure 9, since we saw in figure 8 that most samples with small orientation error have translation error within a few centimeters (and the average model size is about 10 cm). In figure 9(b), we also compare performance on the SOPE/uncluttered datasets to the proposal[7] sample accuracy of SAC-IA, the RANSAC-based alignment method in the original FPFH paper[18]. Our method, MC-SOPE, is either comparable or better to the proposal method of SAC-IA on all three datasets, with the biggest improvement being on the PR2 dataset (which has the most sensor noise). Further gains in MC-SOPE may be acheived by incorporating some of the feature selection/pruning methods in SAC-IA, which we have not yet explored in this work.

VI. Related Work

Many modern techniques for pose estimation of a rigid 3D object are correspondence-based, relying on the presence of unique visual features on the object's surface so that a rigid-body transform can be found which minimizes the sum of squared distances between corresponding points [10], [18], [16]. Unfortunately, many common household objects—such as dishes, tools, or furniture—do not have such uniquely-identifiable visual features (or they have too few unique features to lock down a pose). Furthermore, these correspondence-based techniques are typically limited to recognition and pose estimation of specific objects, rather than classes of objects.

Current alternatives to the correspondence-based approach tend to be limited in some way. Spherical harmonics can be used in an attempt to bring the object into a "standard" reference frame [4], and tend to work better than pure moment matching techniques. However, such standardization approaches may fail when a unique reference frame can't be found, and are sensitive to noise and occlusions. Brute force techniques using Extended Gaussian Images [9], [11] or registration [2] must exhaustively search the space of 3D object rotations, and are currently too slow for real-time applications. The generalized Hough transform [1] uses local geometric information to vote for object parameters (e.g. position and orientation), and is robust to occlusions. However, it suffers from quantization noise which can make parameter searches in high dimensions difficult [8]. Geometric hashing [15] tries to alleviate this quantization difficulty by voting among a discrete set of basis point tuples, but it has a worst case running time of $O(n^4)$ for n 3D model points. Classification-based techniques, such as [19], which classify object observations by viewpoint, appear to work well when the object is perfectly segmented out of the scene, but are sensitive to clutter and occlusions.

Our method can be most accurately described as a hybrid between the Generalized Hough transform and RANSAC, in that we try to model the relationship between the poses of local features and the pose of the model in a random sampling framework. However, unlike the Hough transform, our model is continuous, which allows us to avoid discretization issues and perform extremely fast, constant time inference.

VII. Conclusion

Robust recognition and pose estimation of common household objects is an important capability for any personal service robot. However, real-world scenes often contain a great deal of clutter and occlusion, making the estimation task difficult. Therefore, it is necessary to infer as much as possible from small object features, such as the oriented surface patches we considered in this work. Although our results on both uncluttered and cluttered datasets demonstrated the applicability of our approach to 3D point cloud models, the full power of 3D oriented local feature distributions will only be seen when more feature types (based on color, edges, local image appearance, etc.) are added to the model. Then, the pose of objects with unique visible features, such as the logo on a mug, will be locked down immediately, while observations of non-unique features such as the ones considered in this paper will still lead to a much smaller set of possible object poses that the algorithm will ultimately need to consider.

We also believe the Bingham distribution is a valuable but underutilized tool in robotics for modeling rotational uncertainty, and plan to release an open-source library to allow others to easily incorporate the Bingham distribution and its mixture model into their algorithms.

References

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

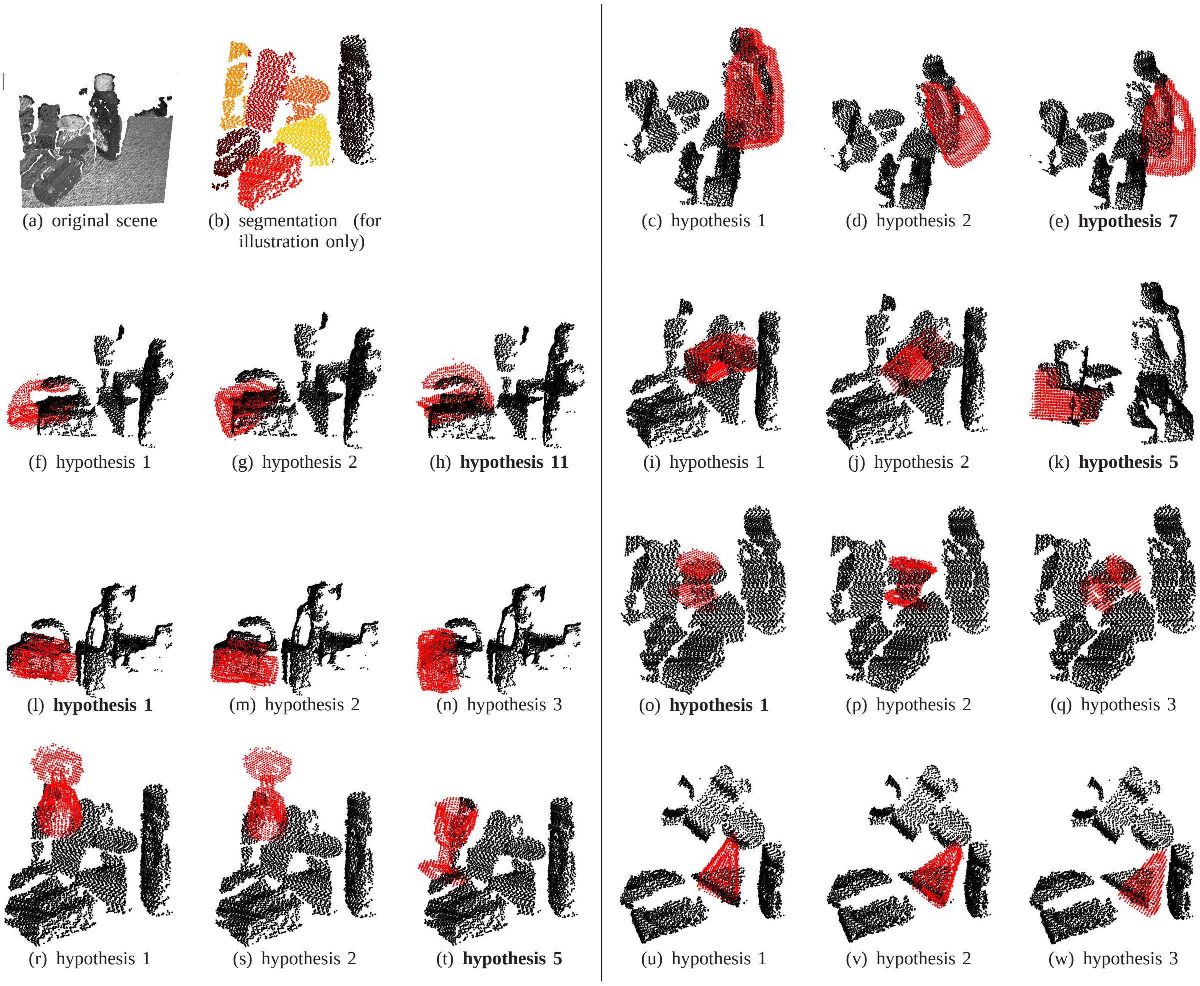
[2] P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.

[3] Christopher Bingham. An antipodally symmetric distribution on the sphere. *The Annals of Statistics*, 2(6):1201–1225, 1974.

[4] Gilles Burel and Hugues Henoco. Determination of the orientation of 3D objects using spherical harmonics. *Graph. Models Image Process.*, 57(5):400–408, 1995.

[5] I. Dryden and K. Mardia. *Statistical Shape Analysis*. John Wiley and Sons, 1998.

[7] The validation part of SAC-IA, which ranks each pose sample by computing an inlier percentage over the entire observed point cloud, is left out of the experiment for a fair comparison of how each algorithm does using only a few local features at a time.



(a) original scene (b) segmentation (for illustration only) (c) hypothesis 1 (d) hypothesis 2 (e) **hypothesis 7**

(f) hypothesis 1 (g) hypothesis 2 (h) **hypothesis 11** (i) hypothesis 1 (j) hypothesis 2 (k) **hypothesis 5**

(l) **hypothesis 1** (m) hypothesis 2 (n) hypothesis 3 (o) **hypothesis 1** (p) hypothesis 2 (q) hypothesis 3

(r) hypothesis 1 (s) hypothesis 2 (t) **hypothesis 5** (u) hypothesis 1 (v) hypothesis 2 (w) hypothesis 3

Figure 10. Top sample poses for 7 objects found by the MC-SCOPE algorithm for a cluttered scene. When a "good" pose sample is found within the top 20 samples, the caption under the best sample is shown in bold.

[6] Wendelin Feiten, Pradeep Atwal, Robert Eidenberger, and Thilo Grundmann. 6D pose uncertainty in robotic perception. In *Advances in Robotics Research*, pages 89–98. Springer Berlin Heidelberg, 2009.

[7] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[8] W. E. L. Grimson and D. P. Huttenlocher. On the sensitivity of the hough transform for object recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(3):255–274, 1990.

[9] B.K.P. Horn. Extended gaussian images. *Proceedings of the IEEE*, 72(12):1671–1686, 1984.

[10] Andrew Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433 – 449, May 1999.

[11] S.B. Kang and K. Ikeuchi. The complex egi: A new representation for 3-d pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:707–721, 1993.

[12] John T. Kent. Asymptotic expansions for the bingham distribution. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 36(2):139–144, 1987.

[13] A. Kume and Andrew T. A. Wood. Saddlepoint approximations for the bingham and Fisher-Bingham normalising constants. *Biometrika*, 92(2):465–476, June 2005.

[14] Karsten Kunze and Helmut Schaeben. The bingham distribution of quaternions and its spherical radon transform in texture analysis. *Mathematical Geology*, 36(8):917–943, November 2004.

[15] Y. Lamdan and H.J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Computer Vision., Second International Conference on*, pages 238–249, 1988.

[16] S. Linnainmaa, D. Harwood, and L.S. Davis. Pose determination of a three-dimensional object using triangle pairs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:634–647, 1988.

[17] K. V. Mardia. Characterizations of directional distributions. In *Statistical Distributions in Scientific Work*, volume 3, pages 365–385. D. Reidel Publishing Company, Dordrecht, 1975.

[18] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast Point Feature Histograms (FPFH) for 3D Registration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 12-17 2009.

[19] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, and John Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 10/2010 2010.

[20] Min Sun, Bing-Xin Xu, Gary Bradski, and Silvio Savarese. Depth-encoded hough voting for joint object detection and shape recovery. In *ECCV*, Crete, Greece, 2010.

[21] Robert Tibshirani and Trevor Hastie. Local likelihood estimation. *Journal of the American Statistical Association*, 82(398):559–567, 1987.