

Probabilistic Search for Object Segmentation and Recognition

Ulrich Hillenbrand and Gerd Hirzinger

Institute of Robotics and Mechatronics, German Aerospace Center/DLR
P.O. Box 1116, 82230 Wessling, Germany
`Ulrich.Hillenbrand@dlr.de`

Abstract. The problem of searching for a model-based scene interpretation is analyzed within a probabilistic framework. Object models are formulated as generative models for range data of the scene. A new statistical criterion, the truncated object probability, is introduced to infer an optimal sequence of object hypotheses to be evaluated for their match to the data. The truncated probability is partly determined by prior knowledge of the objects and partly learned from data. Some experiments on sequence quality and object segmentation and recognition from stereo data are presented. The article recovers classic concepts from object recognition (grouping, geometric hashing, alignment) from the probabilistic perspective and adds insight into the optimal ordering of object hypotheses for evaluation. Moreover, it introduces point-relation densities, a key component of the truncated probability, as statistical models of local surface shape.

1 Introduction

Model-based object recognition or, more generally, scene interpretation can be conceptualized as a two-part process: one that generates a sequence of hypotheses on object identities and poses, the other that evaluates them based on the object models. Viewed as an optimization problem, the former is concerned with the search sequence, the latter with the objective function. Usually the evaluation of the objective function is computationally expensive. A reasonable search algorithm will thus arrive at an acceptable hypothesis within a small number of such evaluations.

In this article, we will analyze the search for a scene interpretation from a probabilistic perspective. The object models will be formulated as generative models for range data. For visual analysis of natural scenes, that is, scenes that are cluttered with multiple, non-completely visible objects in an uncontrolled context, it is a highly non-trivial task to optimize the match of a generative model to the data. Local optimization techniques will usually get stuck in meaningless, local optima, while techniques akin to exhaustive search are precluded by time constraints. The critical aspect of many object recognition problems hence concerns the generation of a clever search sequence.

In the probabilistic framework explored here, the problem of optimizing the match of generative object models to data is alleviated by the introduction of another statistical match criterion that is more easily optimized, albeit less reliable in its estimates of object parameters. The new criterion is used to define a sequence of hypotheses for object parameters of decreasing probability, starting with the most probable, while the generative model remains the measure for their evaluation. For an efficient generation of the search sequence, it is desirable to make the new criterion as simple as possible. On the other hand, for obtaining a short search sequence for acceptable object parameters, it is necessary to make the criterion as informative as is feasible. As a key quantity for sequence optimization, an estimate of features' posterior probability enters the new criterion. The method is an alternative to other, often more heuristic strategies aimed at producing a short search sequence, such as checking for model features in the order of the features' prior probability [7,12], in a coarse-to-fine hierarchy [6], or as predicted from the current scene interpretation [4,8].

Classic hypothesize-and-test paradigms have been demonstrated in RANSAC-like [5,19] and alignment techniques [9,10]. The method proposed here is more similar to the latter in that testing of hypotheses is done with respect to the full data rather than on a sparse feature representation. It significantly differs from both, however, in that it recommends a certain *order* of hypotheses to be tested, which is indeed the main point of the present study. Other classic concepts that are recovered naturally from the probabilistic perspective are feature-based indexing and geometric hashing [13,20], and feature grouping and perceptual organization [14,15,11].

To keep the notational load in this article to a minimum and to support ease of reading, we will denote all probability densities by the letter p and indicate each type of density by its arguments. Moreover, we will not introduce symbols for random variables but only for the values they take. It is understood that probability densities become probabilities whenever these values are discrete.

2 Generative Object Models for Range Data

Consider the task of recognizing and localizing a number of objects in a scene. More precisely, we want to estimate object parameters (c, \mathbf{p}) from data, where $c \in \mathbb{N}$ is the object's discrete class label and $\mathbf{p} \in \mathbb{R}^6$ are its pose parameters (rotation and translation). Suppose we use a sensor like a set of stereo cameras or a laser scanner to obtain range-data points $\mathbf{d} \in \mathbb{R}^3$ of one view of the scene, that is, $2 + 1/2$ dimensional ($2 + 1/2D$) range data. A reasonable generative model of the range-data points within the volume $\mathcal{V}(c, \mathbf{p})$ of the object c with pose \mathbf{p} is given by the conditional probability density

$$p(\mathbf{d}|c, \mathbf{p}) = \begin{cases} N(c) f[\phi(\mathbf{d}; c, \mathbf{p})] & \text{for } \mathbf{d} \in \mathcal{S}(c, \mathbf{p}), \\ N(c) b & \text{for } \mathbf{d} \in \mathcal{V}(c, \mathbf{p}) \setminus \mathcal{S}(c, \mathbf{p}). \end{cases} \quad (1)$$

The condition is on the object parameters (c, \mathbf{p}) . The set $\mathcal{S}(c, \mathbf{p}) \subset \mathcal{V}(c, \mathbf{p})$ is the region close to visible surfaces of the object c with pose \mathbf{p} , $b > 0$ is a constant that

describes the background of spurious data points, and $N(c)$ is a normalization constant. The surface density of points is described by a function f of the angle $\phi(\mathbf{d}; c, \mathbf{p})$ between the direction of gaze of the sensor and the object's inward surface normal at the (close) surface point $\mathbf{d} \in \mathcal{S}(c, \mathbf{p})$. The function f takes its maximal value, that we may set to 1, at $\phi = 0$, i.e., on surfaces orthogonal to the sensor's gazing direction.

Four comments are in order. First, the normalization constant $N(c)$ generally depends upon both c and \mathbf{p} . However, we here neglect its dependence on the object pose \mathbf{p} . The consequence is that objects will be harder to detect, if they expose less surface area to the sensor. Second, the extension of the set $\mathcal{S}(c, \mathbf{p})$ has to be adapted to the level of noise in the range data. Third, points $\mathbf{d} \in \mathcal{S}(c, \mathbf{p})$ that are close to but not on the surface have assigned surface normals of close surface points. Fourth, given the object parameters (c, \mathbf{p}) , the data points $\mathbf{d} \in \mathcal{V}(c, \mathbf{p})$ can be assumed statistically independent with reasonable accuracy.

Our task is to estimate the parameters c and \mathbf{p} by optimizing the match between the generative model (1) to the range data D . Because of the conditional independence of the data points in $\mathcal{V}(c, \mathbf{p})$, this means to maximize the logarithmic likelihood

$$L(c, \mathbf{p}; D) := \sum_{\mathbf{d} \in D \cap \mathcal{V}(c, \mathbf{p})} L(c, \mathbf{p}; \mathbf{d}) := \sum_{\mathbf{d} \in D \cap \mathcal{V}(c, \mathbf{p})} \ln p(\mathbf{d} | c, \mathbf{p}) \quad (2)$$

with respect to $(c, \mathbf{p}) \in \Omega \subset \mathbb{N} \times \mathbb{R}^6$. A computationally efficient version is obtained by assuming that $\phi \ll 1$, which is true for patches of surface that are approximately orthogonal to the direction of gaze. Fortunately, such parts of the surface contribute most data points; cf. (1). Observing that

$$\ln f(\phi) = -a\phi^2 + O(\phi^4) = 2a(\cos \phi - 1) + O(\phi^4) \quad , \quad (3)$$

with some constant $a > 0$, we may neglect terms of order $O(\phi^4)$ to obtain

$$L(c, \mathbf{p}; \mathbf{d}) \approx \begin{cases} 2a[\mathbf{n}(\mathbf{d}; c, \mathbf{p}) \cdot \mathbf{g} - 1] + \ln N(c) & \text{for } \mathbf{d} \in \mathcal{S}(c, \mathbf{p}), \\ \ln b + \ln N(c) & \text{for } \mathbf{d} \in \mathcal{V}(c, \mathbf{p}) \setminus \mathcal{S}(c, \mathbf{p}), \end{cases} \quad (4)$$

where $\mathbf{n}(\mathbf{d}; c, \mathbf{p})$ is the object's inward surface-normal vector at the (close) surface point $\mathbf{d} \in \mathcal{S}(c, \mathbf{p})$ and \mathbf{g} is the sensor's direction of gaze; both are unit vectors. The constant $\ln b < 0$ is effectively a penalty term for data points that come to lie in the object's interior under the hypothesis (c, \mathbf{p}) . Again, discarding the terms $O(\phi^4)$ in (3) is acceptable as data points producing a large error will be rare; cf. (1).

3 Probabilistic Search for Model Matches

In this section, we derive the probabilistic search algorithm. We first introduce another statistical criterion function of object parameters (c, \mathbf{p}) . The new criterion, to be called *truncated probability* (TP), defines a sequence of (c, \mathbf{p}) -hypotheses. The hypotheses are evaluated by the likelihood (2), starting with

the most probable (c, \mathbf{p}) -value and proceeding to gradually less probable values, as estimated by the TP. The search terminates as soon as $L(c, \mathbf{p}; D)$ is large enough.

We will obtain the TP from a series of steps that simplify from the ideal criterion, the posterior probability of object parameters (c, \mathbf{p}) . Although this derivation cannot be regarded as a thoroughly controlled approximation, it will make explicit the underlying assumptions and give some feeling for how far we have to depart from the ideal to arrive at a feasible criterion.

The TP makes use of the probability of the presence of feature values consistent with object parameters (c, \mathbf{p}) . It is thus an essential property of the approach to treat features as random variables.

3.1 The Search Sequence

Let us define point features as surface points centered on certain local, isolated surface shapes. Examples of such point features are corners, saddle points, points of locally-extreme surface curvature etc. They are characterized by a pair (s, \mathbf{f}) , where $s \in \mathbb{N}$ is the feature's shape-class label and $\mathbf{f} \in \mathbb{R}^3$ is its location. Consider now a random variable that takes values (s, \mathbf{f}) of point features that are related to the objects sought in the data. Its values are statistically dependent upon the range data D . Let us restrict the possible feature values (s, \mathbf{f}) to $s \in \{1, 2, \dots, m\}$ and $\mathbf{f} \in D$, such that only data points can be feature locations. This is equivalent to setting the feature-value probability to 0 for values (s, \mathbf{f}) with $\mathbf{f} \notin D$. The restriction is not really correct as we will loose true feature locations between data points. However, it will greatly facilitate the search by limiting features to discrete values that lie close to the true object surfaces and, hence, include highly probable candidates.

Let us introduce the concept of *groups* of point features. A feature group is a set $G_g = \{(s_1, \mathbf{f}_1), (s_2, \mathbf{f}_2), \dots, (s_g, \mathbf{f}_g)\}$ of feature values with $\mathbf{f}_i \neq \mathbf{f}_j$ for $i \neq j$. A group G_g hence contains *simultaneously possible* values of g features.

The best knowledge we could have about true values of the object parameters (c, \mathbf{p}) is encapsulated in their posterior-probability density given the data D , $p(c, \mathbf{p}|D)$. However, we usually do not know this density, and if we knew it, its maximization would pose a problem similar to our initial one of maximizing the generative model (2). We can nonetheless expand it using feature groups,

$$p(c, \mathbf{p}|D) = \sum_{G_g} p(c, \mathbf{p}|D, G_g) p(G_g|D) , \quad \sum_{G_g} p(G_g|D) = 1 . \tag{5}$$

The summations are over all possible groups G_g of fixed size g . Their enumeration is straightforward but tedious to explicate, so we omit this here. Note that the expansion (5) is only valid, if we can be sure to find g true feature locations among the data points.

Let us now simplify the density (5) to the density

$$p_g(c, \mathbf{p}; D) := \sum_{G_g} p(c, \mathbf{p}|G_g) p(G_g|D) . \tag{6}$$

Unlike the posterior density (5), p_g depends upon the feature-group size g . Formally, p_g is a Bayesian belief network with a g -feature probability distribution at the intermediate node. Maximization of p_g with respect to (c, \mathbf{p}) is still too difficult a task, as all possible feature-group values contribute to all values of (c, \mathbf{p}) -density.

A radically simplifying step thus takes us naturally to the density

$$q_g(c, \mathbf{p}; D) := \max_{G_g} p(c, \mathbf{p}|G_g) p(G_g|D) , \quad (7)$$

where for each (c, \mathbf{p}) only the largest term in the sum of (6) contributes. Note that $q_g(c, \mathbf{p}; D) \leq p_g(c, \mathbf{p}; D)$ for all (c, \mathbf{p}) , such that q_g is not normalized on the set Ω of object parameters (c, \mathbf{p}) . This density will nonetheless be useful for our purpose of guiding a search through Ω , as there only *relative* densities matter.

As pointed out above, for all this to make sense, we need g true feature locations among the data points. To be safe, one could be tempted to set $g = 1$. However, the simplifying step from density p_g to density q_g suggests that the latter will be more informative as to the true value of (c, \mathbf{p}) , if the sum in (6) is dominated for high-density values of (c, \mathbf{p}) by only *few* terms. Now, less than three point features do not generally define a *finite* set of consistent object parameters (c, \mathbf{p}) . The density $p(c, \mathbf{p}|G_g)$ will hence not exhibit a pronounced maximum for $g < 3$. High-density points of p_g arise then from accumulation over *many* g -feature values, i.e., terms in (6), and are necessarily lost in q_g . Groups of size $g \geq 3$, on the other hand, do define finite sets of consistent (c, \mathbf{p}) -values, and $p(c, \mathbf{p}|G_g)$ has infinite density there. Altogether, feature groups of size $g = 3$ seem to be a good choice for our search procedure; see, however, the discussion in Sect. 5.

Let us now introduce the logarithmic *truncated probability* (TP),

$$\text{TP}(c, \mathbf{p}; D) := \lim_{\epsilon \rightarrow 0} \ln \int_{S_\epsilon(\mathbf{p})} d\mathbf{p}' q_3(c, \mathbf{p}'; D) , \quad (8)$$

where $S_\epsilon(\mathbf{p})$ is a sphere in pose space centered on \mathbf{p} with radius $\epsilon > 0$. The integral and limit are needed to pick up infinities in the density $q_3(c, \mathbf{p}; D)$; see below. The TP is truncated in a double sense: finite contributions from q_3 are truncated and q_3 itself is obtained from truncating the sum in (6).

According to the discussion above, it is expected that (c, \mathbf{p}) -values of high likelihood (2) will mostly yield a high TP (8). Our search thus proceeds by evaluating, in that order,

$$L(c_1, \mathbf{p}_1; D), L(c_2, \mathbf{p}_2; D), \dots \quad \text{with}$$

$$\text{TP}(c_1, \mathbf{p}_1; D) \geq \text{TP}(c_2, \mathbf{p}_2; D) \geq \dots , \quad (c_1, \mathbf{p}_1) = \arg \max_{(c, \mathbf{p}) \in \Omega} \text{TP}(c, \mathbf{p}; D) . \quad (9)$$

The search stops, as soon as $L(c_k, \mathbf{p}_k; D) > \Theta$ or all (c, \mathbf{p}) -candidates have been evaluated. In the former case, the object identity c_k and object pose \mathbf{p}_k are returned as the estimates. In the latter case, it is inferred that none of the objects sought is present in the scene. Alternatively, if we know a priori that one of the objects *must* be present, we may pick the object parameters that have

scored highest under L in the whole sequence. The algorithm will be formulated in more detail in Sect. 3.3.

In the density (7) that guides the search, one of the factors is the density of the object parameters conditioned on the values G_3 of a triple of point features. This is explicitly

$$p(c, \mathbf{p} | G_3) = \begin{cases} \sum_{i=1}^{h(G_3)} \gamma_i(G_3) \delta_{c, C_i(G_3)} \delta[\mathbf{p} - P_i(G_3)] \\ \quad + \left[1 - \sum_{i=1}^{h(G_3)} \gamma_i(G_3) \right] \rho_1(c, \mathbf{p}; G_3) & \text{for } G_3 \in \mathcal{C}, \\ \rho_2(c, \mathbf{p}; G_3) & \text{for } G_3 \notin \mathcal{C}. \end{cases} \quad (10)$$

Here $\delta_{c,c'}$ is the Kronecker delta (elements of the unit matrix) and $\delta(\mathbf{p} - \mathbf{p}')$ is the Dirac-delta distribution on pose space; \mathcal{C} is the set of feature-triple values *consistent* with any object hypotheses; $h(G_3)$ is the number of object hypotheses consistent with the feature triple G_3 ; $(C_i(G_3), P_i(G_3))$ are the consistent (c, \mathbf{p}) -hypotheses; $\gamma_i(G_3) \in (0, 1)$ are probability-weighting factors for the hypotheses. Generally we have

$$\sum_{i=1}^{h(G_3)} \gamma_i(G_3) < 1, \quad (11)$$

which leaves a non-vanishing probability $1 - \sum_i \gamma_i > 0$ that three consistent features do not all belong to one of the objects sought. Accordingly, the density $p(c, \mathbf{p} | G_3)$ on Ω contains some finite, spread contribution $\propto \rho_1(c, \mathbf{p}; G_3)$, a “background” density, in the consistent-features case. In the inconsistent-features case, the density is similarly spread on Ω and given by some finite term $\rho_2(c, \mathbf{p}; G_3)$. Clearly, only the values (C_i, P_i) of object parameters where the density (10) is infinite will be visited during the search (9); cf. (8).

The functions $\gamma_i(G_3)$, $C_i(G_3)$, $P_i(G_3)$ are computed by generalized *geometric hashing*. Let $G_3 = \{(s_1, \mathbf{f}_1), (s_2, \mathbf{f}_2), (s_3, \mathbf{f}_3)\}$. As key into the hash table we use the pose invariants

$$(s_1, s_2, s_3, \|\mathbf{f}_1 - \mathbf{f}_2\|, \|\mathbf{f}_2 - \mathbf{f}_3\|, \|\mathbf{f}_3 - \mathbf{f}_1\|), \quad (12)$$

appropriately shifted, scaled, and quantized. Here $\|\cdot\|$ denotes the Euclidean vector norm. The poses $P_i(G_3)$, $i = 1, 2, \dots, h(G_3)$, are computed from matching the triple G_3 to $h(G_3)$ model triples from the hash table. Small model distortions that result from an error-tolerant match are removed by orthogonalization of the match-related pose transforms. The weight $\gamma_i(G_3)$ and the object class $C_i(G_3)$ are drawn along with each matched model triple.

The weights $\gamma_i(G_3)$ can be adapted during a training period and during operation of the system in an unsupervised manner. One simply needs to count the number of instances where a feature triple G_3 draws a correct set of object parameters (C_i, P_i) , i.e., one with $L(C_i, P_i; D) > \Theta$. The adapted $\gamma_i(G_3)$ represent empirical *grouping laws* for the point features. Since it will often be the case that $\gamma_i(G_3) \propto 1/h(G_3)$ for all $i = 1, 2, \dots, h(G_3)$, it turns out that hypotheses drawn by feature triples *less common* among the sought objects tend to be visited *before* the ones drawn by more common triples during the search (9).

In order to fully specify the TP (8) and formulate our search algorithm, it remains to establish a model for the feature-value probabilities $p(G_3|D)$ given the data D ; cf. (7). This is the subject of the next section.

3.2 Inferring Local Surface Shape from Point-Relation Densities

In principle, a generative model of range data on local surface shapes s can be built analogous to (1). Thus, we may construct a conditional probability density $p(D|s, \mathbf{f}, \mathbf{r})$, where $\mathbf{r} \in \mathbb{R}^3$ are the rotational parameters of the local surface patch represented by the point feature (s, \mathbf{f}) . During recognition, however, it would then be necessary to optimize the parameters \mathbf{r} for each $s \in \{1, 2, \dots, m\}$ and $\mathbf{f} \in D$, in order to obtain the likelihood of each feature value (s, \mathbf{f}) . Besides the huge overhead of doing so without being interested in an estimate for \mathbf{r} , this procedure would confront us with a problem similar to our original one of optimizing $p(D|c, \mathbf{p})$. On the other hand, the density $p(D|s, \mathbf{f})$ is an infeasible model because of high-order correlations between the data points D . Neglecting these correlations would throw out all information on surface shape s .

The strategy we pursue here is to capture some of the informative correlations between data points D in a family of new representations $T(\mathbf{c}) = \{\Delta_1, \Delta_2, \dots, \Delta_N\}$, $\mathbf{c} \in \mathbb{R}^3$. Each $T(\mathbf{c})$ represents the geometry of the data D relative to the point \mathbf{c} , and each Δ_i represents geometric relations between multiple data points. A reasonable statistical model is then obtained by neglecting correlations between the Δ_i .

Learning Point-Relation Densities. The point relations we are going to exploit here are between four data points, that is, tetrahedron geometries, where three of the points are selected from a spherical neighborhood of the fourth. For four points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{c} \in \mathbb{R}^3$ we define the map

$$\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{pmatrix} \mapsto \Delta(\mathbf{c}; \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) := \begin{pmatrix} r/R \\ d/\sqrt{r^2 - \frac{4a}{3\sqrt{3}}} \\ 4a/3\sqrt{3}(r^2 - d^2) \end{pmatrix} \in [0, 1] \times [-1, 1] \times [0, 1], \tag{13}$$

where r is the mean distance of the center \mathbf{c} to $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$, R is the radius of the spherical neighborhood of \mathbf{c} , d is the (signed) distance of \mathbf{c} to the plane defined by $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$, and a is the area of the triangle they define¹. Explicitly,

$$r = \frac{1}{3} \sum_{i=1}^3 \|\mathbf{x}_i - \mathbf{c}\|, \tag{14}$$

$$d = \text{sgn}[\mathbf{g} \cdot (\mathbf{x}_1 - \mathbf{c})] \frac{\|(\mathbf{x}_1 - \mathbf{c}) \cdot [(\mathbf{x}_2 - \mathbf{x}_1) \times (\mathbf{x}_3 - \mathbf{x}_1)]\|}{\|(\mathbf{x}_2 - \mathbf{x}_1) \times (\mathbf{x}_3 - \mathbf{x}_1)\|}, \tag{15}$$

$$a = \frac{1}{2} \|(\mathbf{x}_2 - \mathbf{x}_1) \times (\mathbf{x}_3 - \mathbf{x}_1)\|. \tag{16}$$

¹ The quantity introduced in (13) will be denoted by $\Delta(\mathbf{c}; \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$, $\Delta(\mathbf{c})$, or simply Δ to indicate its dependence on points as needed.

As seen in (15), the sign of d is determined by the direction of gaze \mathbf{g} . We can now define the family of *tetrahedron representations* $T(\mathbf{c})$ by

$$T(\mathbf{c}) := \left\{ \Delta(\mathbf{c}; \mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3) \mid \{ \mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3 \} \subseteq D \wedge \max_{i=1,2,3} \|\mathbf{d}_i - \mathbf{c}\| < R \right. \\ \left. \wedge \max_{i=1,2,3} \|\mathbf{d}_i - \mathbf{c}\| - \min_{i=1,2,3} \|\mathbf{d}_i - \mathbf{c}\| < \epsilon \right\}. \quad (17)$$

The parameter $\epsilon > 0$ sets a small tolerance for differences in distance to \mathbf{c} among each data-point triple $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$, that is, ideally all three points have the same distance. In practice, the tolerance ϵ has to be adapted to the density of range-data points D to obtain enough Δ -samples in each $T(\mathbf{c})$. Figure 1 depicts the geometry of the tetrahedron representations.

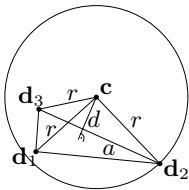


Fig. 1. Transformation of range data points to a tetrahedron representation. For an inspection point \mathbf{c} , the 3-tuples (r, d, a) are collected and normalized [cf. (13)] for each triple of data points $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3 \in D$ with (approximately) equal distance $r \in (0, R)$ from \mathbf{c} .

One of the nice properties of the tetrahedron representation (17) is that for any surface of revolution we get at its symmetry point \mathbf{c} samples $\Delta \in T(\mathbf{c})$ that are confined to a surface in Δ -space characteristic of the original surface shape. For surfaces of revolution, the distinctiveness and low entropy of the distribution of range data D sampled from a surface in 3D is thus completely preserved in the tetrahedron representation. For more general surfaces, it is therefore expected that a high mutual information between Δ -samples and shape classes can be achieved.

Our goal is to estimate *point-relation densities* for each shape class $s = 1, 2, \dots, m$. Samples are taken from tetrahedron representations centered on feature locations \mathbf{f} from a training set \mathcal{F}_s , that is, from $\cup_{\mathbf{f} \in \mathcal{F}_s} T(\mathbf{f})$. Moreover, we add a non-feature class $s = 0$ that lumps together all the shapes we are not interested in. The result are the shape-conditioned densities $p(\Delta|s)$ for $\Delta \in [0, 1] \times [-1, 1] \times [0, 1]$ and $s = 0, 1, 2, \dots, m$. In particular, $p(\Delta|s) = p(\Delta|s, \mathbf{f})$ for $\Delta \in T(\mathbf{f})$.

Estimation of $p(\Delta|s)$ is made simple by the fact that we get $O(n^3)$, i.e., a lot of samples of Δ for each feature location \mathbf{f} with n data points within a distance R . For our application, it is sufficient to count Δ -samples in bins to estimate probabilities $p(\Delta|s)$ for discrete Δ -events as normalized frequencies.

Inferring Local Surface Shape. Let the features' prior probabilities be $p(s)$ for $s = 1, 2, \dots, m$, that is, $p(s) \in (0, 1)$ is the probability that any given data point from D is a feature location of type s . The feature priors are virtually impossible to know for general scenes. We do know, however, that $p(s) \ll 1$ for

$s = 1, 2, \dots, m$, i.e., the overwhelming majority of data points are not feature locations. This must be so for a useful dictionary of point features. It thus makes sense to expand the logarithm of the shapes' posterior probabilities, given l samples $\Delta_1(\mathbf{f}), \Delta_2(\mathbf{f}), \dots, \Delta_l(\mathbf{f}) \in T(\mathbf{f}), \mathbf{f} \in D$,

$$\ln p[s|\Delta_1(\mathbf{f}), \Delta_2(\mathbf{f}), \dots, \Delta_l(\mathbf{f})] = \ln p(s) + \sum_{i=1}^l \{\ln p[\Delta_i(\mathbf{f})|s] - \ln p[\Delta_i(\mathbf{f})|0]\} + O[p(1), p(2), \dots, p(m)] \quad (18)$$

and neglect the terms $O[p(1), p(2), \dots, p(m)]$. Remember that $p(\Delta|0)$ is the Δ -density of the non-feature class. The expression (18) neglects correlations between the $\Delta_i(\mathbf{f}) \in T(\mathbf{f})$.

The sample length l for each representation $T(\mathbf{f})$ will usually be $l \ll |T(\mathbf{f})|$, the cardinality of the set $T(\mathbf{f})$. It is in fact crucial that we do not have to generate the complete representations $T(\mathbf{f})$ for recognition, as this would require $O(n^3)$ time for n data points within a distance R from \mathbf{f} . Instead, it is possible to draw a random, unbiased sub-sample of $T(\mathbf{f})$ in $O(n)$ and $O(l)$ time.

The first term in (18) can be split into

$$\ln p(s) = \ln q(s) + \ln \sum_{s'=1}^m p(s') \quad (19)$$

where $q(s) \in (0, 1)$ are the relative frequencies of shape classes $s = 1, 2, \dots, m$. Now, these frequencies do not differ between features over many orders of magnitude for a useful dictionary of point features². The dependence of (19) on the shape class s is thus negligible compared to the sum in (18) for reasonable sample lengths $l \gg 1$ (several tens to hundreds). The first term in (18) may hence be disregarded, and we are left with the feature's log-probability

$$\begin{aligned}
 \Phi(s, \mathbf{f}; D) &:= \ln p[s|\Delta_1(\mathbf{f}), \Delta_2(\mathbf{f}), \dots, \Delta_l(\mathbf{f})] + \text{const.} \\
 &\approx \sum_{i=1}^l \{\ln p[\Delta_i(\mathbf{f})|s] - \ln p[\Delta_i(\mathbf{f})|0]\} \quad (20)
 \end{aligned}$$

that is, a log-likelihood ratio.

3.3 The Search Algorithm

We still need to determine the TP (8) for guiding the search (9). Let again $G_3 = \{(s_1, \mathbf{f}_1), (s_2, \mathbf{f}_2), (s_3, \mathbf{f}_3)\}$. It is reasonable to assume statistical independence of the point features in G_3 , as long as there are many different objects in the world that have these features³. We hence model the feature-triples' posterior probability $p(G_3|D)$ in (7) as

² A feature that is more than a hundred times less frequent than the others should usually be dropped for computational efficiency.

³ These objects may or may not be in our modeled set.

$$p(G_3|D) = \prod_{i=1}^3 p[s_i|\Delta_1(\mathbf{f}_i), \Delta_2(\mathbf{f}_i), \dots, \Delta_l(\mathbf{f}_i)] \propto \exp \left[\sum_{i=1}^3 \Phi(s_i, \mathbf{f}_i; D) \right]. \quad (21)$$

The TP can then be expressed as

$$\text{TP}(c, \mathbf{p}; D) = \max_{G_3 \in \mathcal{C}} \left[\ln \sum_{i=1}^{h(G_3)} \gamma_i(G_3) \delta_{c, C_i(G_3)} \delta_{\mathbf{p}, P_i(G_3)} + \sum_{i=1}^3 \Phi(s_i, \mathbf{f}_i; D) \right], \quad (22)$$

up to additive constants. The first term under the max-operation is the contribution from feature grouping⁴, the second from single features. In particular, we get

$$\begin{aligned} \text{TP}(c, \mathbf{p}; D) &= -\infty \text{ for} \\ &(c, \mathbf{p}) \notin \{(C_i(G_3), P_i(G_3)) \mid G_3 \in \mathcal{C}, i \in \{1, 2, \dots, h(G_3)\}\} \end{aligned}, \quad (23)$$

that is, for (c, \mathbf{p}) -values not suggested by any feature triple G_3 .

We are now prepared to formulate the search algorithm. The following is not meant to specify a flow of processing, but rather a logic sequence of operations.

1. From all possible feature values $(s, \mathbf{f}) \in \{1, 2, \dots, m\} \times D$, select as feature candidates the values with log-probability $\Phi(s, \mathbf{f}; D) > \Xi$.
2. From the feature candidates, generate all triples G_3 . Use their keys (12) into a hash table to find the associated object hypotheses $(C_i(G_3), P_i(G_3))$ and their weights $\gamma_i(G_3)$ for $i = 1, 2, \dots, h(G_3)$.
3. For the object hypotheses obtained, compute the TPs.
4. Evaluate the object hypotheses by the likelihood L in the order of decreasing TP, starting with the hypothesis scoring highest under TP. Skip duplicate hypotheses. Stop when $L(c, \mathbf{p}; D) > \Theta$ for a hypothesis (c, \mathbf{p}) .
5. Return the last evaluated object parameters (c, \mathbf{p}) as the result, if $L(c, \mathbf{p}; D) > \Theta$. Else conclude that there is none of the sought objects in the data.

Some comments are in order.

- Comment on step 1: The restriction to the most probable feature values by introduction of the threshold Ξ is optional. It is a powerful means, however, to radically speed up the algorithm, if there are many range data points D . If the probability measure on the features is reliable and Ξ is adjusted conservatively, the risk of missing a good set of object parameters is very small; see Sect. 4.1 below.
- Comment on step 2: Feature triples $G_3 \notin \mathcal{C}$, that is, inconsistent triples, are discarded by the hashing process.

⁴ The factor $\delta_{\mathbf{p}, P_i(G_3)}$ is an idealization. Since the pose parameters \mathbf{p} are continuous, the hashing with the key (12) is necessarily error tolerant. The factor is then to be replaced by an integral $\int_{\mathcal{P}_i(G_3)} d\mathbf{p}' \delta(\mathbf{p} - \mathbf{p}')$ for a set $\mathcal{P}_i(G_3)$ of pose parameters.

- Comment on step 3: Computation of TP is cheap as the hashed weights γ_i from step 2 and the feature’s log-probabilities Φ from step 1 can be used; cf. (22).
- Comment on step 4: Duplicate hypotheses may be very unlikely, but this depends on when two hypotheses are considered equal. In any case, skipping them realizes the $\max_{G_3 \in \mathcal{C}}$ -operation in the TP (22), ensuring that each hypothesis is drawn only by the feature triple which makes it most probable.
- Comment on step 5: If the likelihood L does not reach the threshold Θ but we know a priori that at least one of the objects must be present in the scene, the algorithm may return the object parameters (c, \mathbf{p}) that have scored highest under L .

A complete scene interpretation often involves more than one recognized set of object parameters (c, \mathbf{p}) . A simultaneous search for several objects is possible within the current framework, although it will be computationally very costly. It requires evaluation of TPs and likelihoods

$$\text{TP}(c_1, \mathbf{p}_1, c_2, \mathbf{p}_2, \dots ; D) , \quad L(c_1, \mathbf{p}_1, c_2, \mathbf{p}_2, \dots ; D) \quad (24)$$

for a combinatorially enlarged set of object parameters $(c_1, \mathbf{p}_1, c_2, \mathbf{p}_2, \dots)$. A much cheaper, although less reliable, alternative is sequential search, where the data belonging to a recognized object (c, \mathbf{p}) are removed prior to the algorithm’s restart.

4 Experiments

We have performed two types of experiment. One tests the quality of the measure (20) of the features’ posterior probability. The other probes the capabilities of the complete algorithm for object segmentation and recognition. All experiments were run on stereo data obtained from a three-camera device (the ‘Triclops Stereo Vision System’, Point Grey Research Inc.). The stereo data were calculated from three 320×160 -pixel images by a standard least-sum-of-pixel-differences algorithm for correspondence search. The data were accordingly of a rather low resolution. They were moreover noisy, contained a lot of artifacts and outliers, and even visible surfaces lacked large regions of data points, as is not uncommon for stereo data; see Figs. 4 and 5.

4.1 Local Shape Recognition

Only correct feature values are able to draw correct object hypotheses (c, \mathbf{p}) , except from accidental, unprobable events. For an acceptable search length it is, therefore, crucial that correct feature values yield a high log-probability Φ ; cf. (20) and (22). This is even more crucial, if feature candidates are preselected by imposing a threshold $\Phi(s, \mathbf{f}; D) > \Xi$; cf. step 1 of the algorithm in Sect. 3.3.

As point features, we chose convex and concave rectangular corners. The training set contained 221 feature locations that produced 466,485 Δ -samples for the densities $p(\Delta|s)$, $s \in \{\text{“convex corner”}, \text{“concave corner”}\}$. For the non-feature class $s = 0$ we obtained 9,623,073 Δ -samples from several thousand

non-feature locations (mostly on planes, also some edges). The parameters for Δ sampling were $R = 15$ mm and $\epsilon = 1$ mm; cf. Sect. 3.2. The Δ -samples were counted in $15 \times 20 \times 10$ bins, which corresponds to the discretization used for the Δ -space $[0, 1] \times [-1, 1] \times [0, 1]$.

Feature-recognition performance was evaluated on 10 scenes that contained (i) an object with planar surface segments, edges, saddle points, and altogether 65 convex and concave corner points (feature locations); (ii) a piece of loosely folded cloth that contributed a lot of potential distractor shapes. The sample length was $l = 50$ Δ -samples; cf. Sect. 1. Since drawing Δ -samples is a stochastic process, we let the program run 10 times on each scene with different seeds for the random-number generator to obtain a more significant statistics.

Corner points are particularly interesting as test features, since they allow for comparison of the proposed log-probability measure (20) with the classic method of curvature estimation. Let $c_1(\mathbf{f}; D)$ and $c_2(\mathbf{f}; D)$ be the principal curvatures of a second-order polynomial that is least-squares fitted to the data points from D in the R -sphere around the point \mathbf{f} . Because the corners are the maximally curved parabolic shapes in the scenes, it makes sense to use the curvature measures

$$C(s = \text{“convex corner”}, \mathbf{f}; D) := \min[c_1(\mathbf{f}; D), c_2(\mathbf{f}; D)] \quad , \quad (25)$$

$$C(s = \text{“concave corner”}, \mathbf{f}; D) := \min[-c_1(\mathbf{f}; D), -c_2(\mathbf{f}; D)] \quad , \quad (26)$$

as alternative measures of “cornerness” of the point \mathbf{f} for convex and concave shapes, respectively. We thus compare $\Phi(s, \mathbf{f}; D)$ as defined in (20) to $C(s, \mathbf{f}; D)$ on true and false feature values

$$(s, \mathbf{f}) \in \{ \text{“convex corner”}, \text{“concave corner”} \} \times D \quad . \quad (27)$$

In Fig. 2 we show the distribution of Φ - and C -scores for the 65 true feature values in relation to their distribution for all possible feature values, which are more than 100,000. As can be seen, the Φ - and C -distributions for the population of true values are both distinct from, but overlapping with the Φ - and C -distributions for the entire population of feature values. A qualitative difference between Φ - and C -scoring of feature values is more obvious in the *ranking* they produce. In Fig. 3 we show the distribution of Φ - and C -ranks of the true feature values among all feature values, with 1 being the first rank, i.e., highest Φ/C -score, and 0 the last. Both Φ - and C -ranks of true values are most frequently found close to 1 and gradually less frequently at lower ranks. There are almost no true features Φ -ranked below 0.6. The C -ranks, in contrast, seem to scatter uniformly in $[0, 1]$ for part of the true features. These features score under C as poorly as the non-features. If using the C -score, they would in practice not be available for object recognition, as they would only be selected to draw an object hypothesis after hundreds to thousands of false feature values.

The result of the comparison comes not as a complete surprise. Some kind of surface-fitting procedure is necessary for estimation of the principal curvatures [2,3,18]. If not a lot of care is invested, fitting to data will always be very sensitive to outliers and artifacts, of which there are many in typical stereo data. A robust fitting procedure, however, has to incorporate in some way a statistical model of the data that accounts for its outliers and artifacts. We here propose to go all

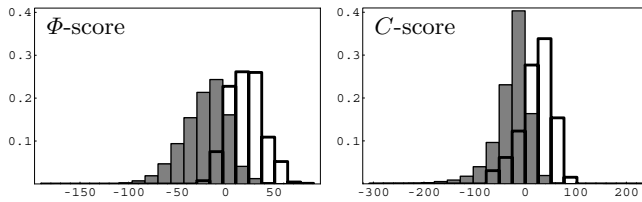


Fig. 2. Score distribution of true (transparent bars, thick lines) and all possible (filled bars, thin lines) feature values as produced by the Φ -score (left) and the C -score (right). Horizontally in each plot extends the score, vertically the normalized frequency of feature values.

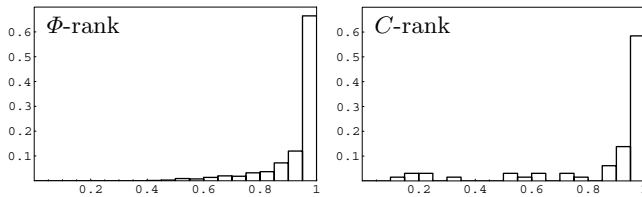


Fig. 3. Rank distribution of true feature values among all possible feature values as produced by the Φ -score (left) and the C -score (right); rank 1 is the highest score in each data set, rank 0 the lowest score. Horizontally in each plot extends the rank, vertically the normalized frequency of true feature values.

the way to a statistical model, the point-relation densities, and to do without any fitting for local shape recognition.

4.2 Object Segmentation and Recognition

We have tested segmentation and recognition performance of the proposed algorithm on two objects. One is a cube from which 8 smaller cubes are removed at its corners; the other is a subset of this object. These two objects can be assembled to create challenging segmentation and recognition scenarios.

The test data were taken from 50 scenes that contained the two objects to be recognized, and additional distractor objects in some of them. A single data set consisted of roughly 5,000 to 10,000 3D points. As an acceptable accuracy for recognition of an object's pose we considered what is needed for grasping with a hand (robot or human), i.e., less than 3 degrees of angular and less than 5 mm of translational misalignment. For a fixed set of parameters of the search algorithm, in all but 3 cases the result was acceptable in this sense. Processing times ranged roughly between 1 and 50 seconds, staying mostly well below 5 seconds, on a Sun UltraSPARC-III workstation (750 MHz).

In Figs. 4 and 5 we show examples of scenes, camera images, range data, and recognition results. Edges drawn in the data outline the object pose recognized first. The other object can be recognized after deletion of data points belonging to the first object.

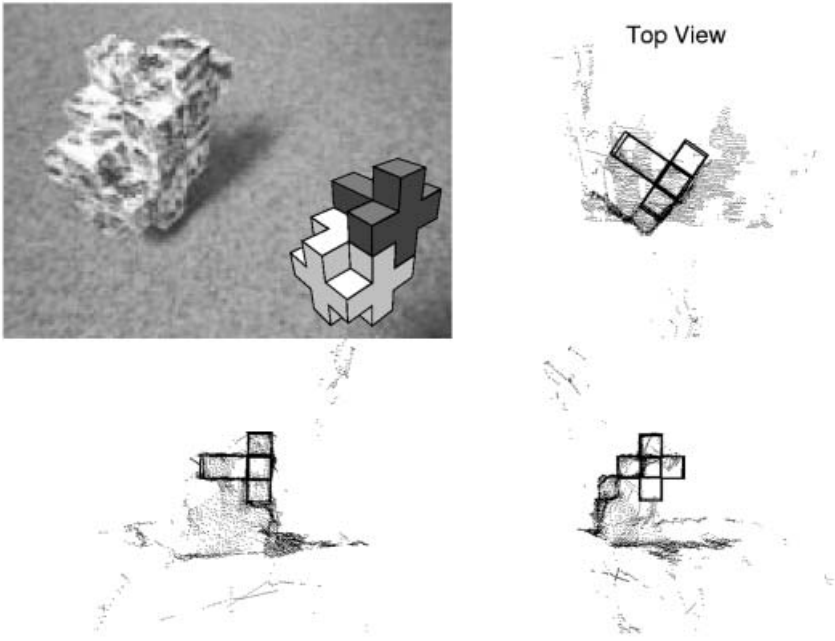


Fig. 4. Example scene with the two objects we used for evaluation of segmentation and recognition performance. The smaller object is stacked on the larger as drawn near the center of the figure. Shown are one of the camera images and three orthogonal views of the stereo-data set. The object pose recognized first is outlined in the data. The length of both objects' longest edges is 13 cm. The recognition time was 1.2 seconds.

5 Conclusion

We have derived from a probabilistic perspective a new variant of hypothesize-and-test algorithm for object recognition. A critical element of any such algorithm is the ordered sequence of hypotheses that is tested. The ordered sequence that is generated by our algorithm is inferred from a statistical criterion, here called the *truncated probability* (TP) of object parameters. As a key component of the truncated probability, we have introduced *point-relation densities*, from which we obtain an estimate of posterior probability for surface shape given range data.

One of the strengths of the algorithm, as demonstrated in experiments, is its very high degree of robustness to noise and artifacts in the data. Raw stereo-data points of low quality are sufficient for many recognition tasks. Such data are fast and cheap to obtain and are intrinsically invariant to changes in illumination.

We have argued in Sect. 3.1 that feature groups of size $g = 3$, i.e., minimal groups, are a good choice for guiding the search for model matches. However, if we can be sure that more than three true object features are represented in the data, it may be worth considering larger groups. For $g > 3$, the density

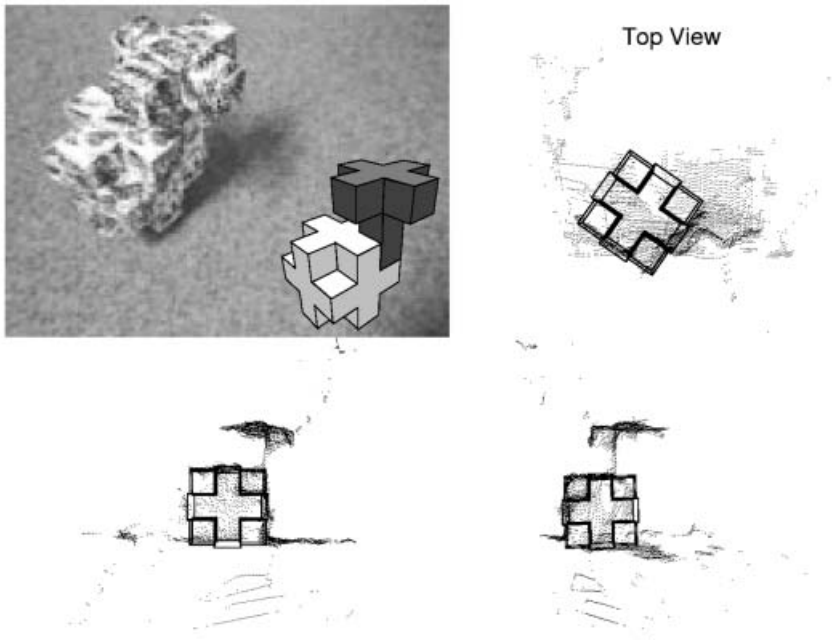


Fig. 5. Example scene with the two objects we used for evaluation of segmentation and recognition performance; cf. Fig. 4. The recognition time was 1.5 seconds.

(10) that enters the TP (8) looks more complicated. In particular, hash tables of higher dimension are needed to accommodate the grouping laws, i.e., the weights $\gamma(G_g)$. If we do without the grouping laws, using larger groups is equivalent to having more than just the largest term of the density (6) contributing to the TP; cf. (7). In the limit of large feature groups, the approach then resembles generalized-Hough-transform and pose-clustering techniques [1,17,16].

One avenue of research that is suggested here is for generalizations of point-relation densities. Alternative representations of range data have to be explored for learning posterior-probability estimates for various surface shapes.

References

- [1] BALLARD, D. H. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognit.* 13 (1981), 111–122.
- [2] BESL, P., AND JAIN, R. Intrinsic and extrinsic surface characteristics. In *Proc. IEEE Conf. Comp. Vision Patt. Recogn.* (1985), pp. 226–233.
- [3] BESL, P., AND JAIN, R. Three-dimensional object recognition. *ACM Comput. Surveys* 17 (1985), 75–145.
- [4] FAUGERAS, O. D., AND HEBERT, M. The representation, recognition, and positioning of 3-D shapes from range data. In *Three-Dimensional Machine Vision*, T. Kanade, Ed. Kluwer Academic Publishers, 1987, pp. 301–353.

- [5] FISCHLER, M., AND BOLLES, R. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Comm. Assoc. Comp. Mach.* 24 (1981), 381–395.
- [6] FLEURET, F., AND GEMAN, D. Graded learning for object detection. In *Proc. IEEE Workshop Stat. Comput. Theo. Vision* (1999).
- [7] GOAD, C. Special purpose automatic programming for 3D model-based vision. In *Proc. ARPA Image Understanding Workshop* (1983).
- [8] GRIMSON, W. E. L. *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press, Cambridge, MA, 1990.
- [9] HUTTENLOCHER, D. P., AND ULLMAN, S. Object recognition using alignment. In *Proc. Intern. Conf. Comput. Vision* (1987), pp. 102–111.
- [10] HUTTENLOCHER, D. P., AND ULLMAN, S. Recognizing solid objects by alignment with an image. *Int. J. Computer Vision* 5 (1990), 195–212.
- [11] JACOBS, D., AND CLEMENS, D. Space and time bounds on indexing 3-D models from 2-D images. *IEEE Trans. Patt. Anal. Mach. Intell.* 13 (1991), 1007–1017.
- [12] KUNO, Y., OKAMOTO, Y., AND OKADA, S. Robot vision using a feature search strategy generated from a 3-D object model. *IEEE Trans. Patt. Anal. Machine Intell.* 13 (1991), 1085–1097.
- [13] LAMDAN, Y., AND WOLFSON, H. J. Geometric hashing: a general and efficient model-based recognition scheme. In *Proc. Intern. Conf. Comput. Vision* (1988), pp. 238–249.
- [14] LOWE, D. G. *Perceptual Organization and Visual Recognition*. Kluwer, Boston, 1985.
- [15] LOWE, D. G. Visual recognition as probabilistic inference from spatial relations. In *AI and the Eye*, A. Blake and T. Troscianko, Eds. John Wiley and Sons Ltd., New York, 1990, pp. 261–279.
- [16] MOSS, S., WILSON, R. C., AND HANCOCK, E. R. A mixture model for pose clustering. *Patt. Recogn. Let.* 20 (1999), 1093–1101.
- [17] STOCKMANN, G. Object recognition and localization via pose clustering. *CVGIP* 40 (1987), 361–387.
- [18] STOKELY, E. M., AND WU, S. Y. Surface parameterization and curvature measurement of arbitrary 3-D objects: Five practical methods. *IEEE Trans. Patt. Anal. Mach. Intell.* 14 (1992), 833–840.
- [19] TORR, P. H. S., AND ZISSERMAN, A. MLESAC: A new robust estimator with application to estimating image geometry. *Comput. Vision Image Understanding* 78 (2000), 138–156.
- [20] WOLFSON, H. J. Model-based object recognition by geometric hashing. In *Proc. Eur. Conf. Comput. Vision* (1990), pp. 526–536.