A Hypothesize-and-Bound Algorithm for Simultaneous Object Classification, Pose Estimation and 3D Reconstruction from a Single 2D Image

Diego Rother
diroth@gmail.com

René Vidal
rvidal@jhu.edu

Johns Hopkins University
3400 N Charles St., Baltimore MD 21218, USA

Abstract

We consider the problems of 3D reconstruction, pose estimation and object classification, from a single 2D image*. In sharp contrast with state of the art methods that solve each of these problems separately or iteratively, we propose a mathematical framework that solves these problems jointly and simultaneously. Since the joint problem is ill posed unless "prior knowledge" is considered, the proposed framework incorporates "prior knowledge" about the 3D shapes of different object classes. This knowledge is used to define a function $L(H)$ that encodes how well each hypothesis H (object class and pose) "explains" the input image. To efficiently maximize $L(H)$ without having to* exactly *evaluate it for each hypothesis H, we propose a H&B algorithm that computes and refines upper and lower bounds for $L(H)$ at a much lower cost. In this way sub-optimal hypotheses are disregarded with little computation. The resulting algorithm integrates information from the 2D image and the 3D prior, is efficient, and is guaranteed to find the optimal solution.*

1. Introduction

It is in general easy for humans to "perceive" three dimensional (3D) objects, even when presented with a single two dimensional (2D) image alone. This ability to "perceive" the world in 3D is essential to interact with the environment and arguably to "understand" the observed scenes. By trying to replicate this ability we come to appreciate the tremendous complexity of this problem, and the marvelous proficiency of our own visual system, capable of "understanding" scenes containing large numbers of objects, exhibiting great variability in shape and appearance.

To tackle this problem, namely 3D reconstruction from a single 2D image, we propose a method that exploits:

Shape cues. For many object classes, *shape* (rather than *appearance*) is a very *good indicator of object class*. For instance, consider how you would describe a 'mug'. In addition, the shapes of objects in the foreground *can often be extracted reliably* from videos using existing background modeling techniques (*e.g.*, [6, 5]).

3D knowledge. In 3D representations, as opposed to 2D ones, the "knowledge" is not tied to a particular viewpoint. Consequently, there is no need to acquire/store different training examples, and/or learn different parameters, for each viewpoint: The resulting methods can in principle *handle any viewpoint*. In addition, 3D representations allow us to *easily impose 3D constraints* (*e.g.*, that an object is resting on a supporting plane, not "flying" above it).

Synergy. To use "3D knowledge" about an object in its 3D reconstruction, we need to know its class (*e.g.*, if an object is a 'mug' it must have a handle somewhere, even if we do not see it). Moreover, to incorporate our knowledge in the right spatial locations, we need to know the object's location/pose (*e.g.*, if the mug is in a known pose, its handle must be in a certain position). To classify the object and estimate its pose, on the other hand, the information in the 3D reconstruction is helpful. Therefore *we solve the tasks of 3D reconstruction, object classification and pose estimation, simultaneously*, rather than in any particular order.

1.1. The general approach

To solve the aforementioned tasks *simultaneously*, while exploiting shape cues and prior 3D knowledge about the object classes, we define a probabilistic graphical model that encodes the relationships among the variables: class K, pose T, input image f, and 3D reconstruction v (described in detail in §3). Because of the large number of variables in this graphical model (the dimensions of f and v are very high), and due to the existence of a huge number of loops in the graph, standard inference methods are either very inefficient or not guaranteed to find the optimal solution.

For this reason, we use the hypothesize-and-verify paradigm to solve our problem: one hypothesis H is defined for every possible "state of the world," and our goal is to select

2011 IEEE International Conference on Computer Vision Workshops
978-1-4673-0063-6/11/$26.00

the hypothesis that best "explains" the input image. In other words, our goal is to select the hypothesis H_* that solves

$$H_* = \operatorname*{argmax}_{H \in \mathbb{H}} L(H), \tag{1}$$

where $\mathbb{H}$ is the set of all possible hypotheses, referred to as the *hypothesis space*, and $L(H)$ is a function, referred to as the *evidence*, that quantifies how well each hypothesis "explains" the input.

In the specific problem addressed in this article, the hypothesis space $\mathbb{H}$ contains every hypothesis H_{ij} defined by each possible object class K_i and each possible object pose T_j (*i.e.*, $H_{ij} \triangleq (K_i, T_j)$). By selecting the hypothesis H_{ij} that solves (1), the hypothesize-and-verify approach simultaneously estimates the class K_i and the pose T_j of the object in the image. As we shall later see, the 3D reconstruction v is estimated during the computation of the evidence. The evidence $L(H)$ for a hypothesis H, on the other hand, is derived from the system's joint probability, which is obtained from the graphical model mentioned above.

Since the number of hypotheses in the set $\mathbb{H}$ is potentially very large, it is essential to evaluate $L(H)$ very efficiently. For this purpose, we exploit an algorithm of the class *hypothesize-and-bound* (H&B) introduced in [10]. This class of algorithms is described next.

1.2. Hypothesize-and-bound algorithms

H&B algorithms have two parts. The first part consists of a *bounding mechanism* (BM) to compute lower and upper bounds, $\underline{L}(H)$ and $\overline{L}(H)$, respectively, for the evidence $L(H)$ of a hypothesis H. These bounds are much cheaper to compute than the evidence itself, and are often enough to discard hypotheses (a hypothesis H_1 can be *safely* discarded if $\overline{L}(H_1) < \underline{L}(H_2)$ for some hypothesis H_2). On the other hand, these bounds are not as "precise" as the evidence itself, since they only define an *interval* where the evidence for a hypothesis is guaranteed to be. Nonetheless, the interval's width can be made arbitrarily small by investing additional computational cycles into the refinement of its bounds. In other words, the *BM can dynamically trade computation for precision*.

The second part of a H&B algorithm is a focus of attention mechanism (FoAM) to sensibly and dynamically allocate the available computational cycles among the different hypotheses whose bounds are to be refined [10, §3]. Initially the FoAM "orders" the BM to compute rough and inexpensive bounds for each hypothesis. Then, during each iteration, the FoAM selects one hypothesis and calls the BM to refine its bounds. This process continues until either a hypothesis is proved optimal, or a group of hypotheses cannot be further refined or discarded. Such hypothesis or group of hypotheses maximize the evidence, *regardless of the exact values of the evidences (which do not need to be computed)*. At each iteration, hypotheses are carefully selected to minimize the total number of cycles spent.

In principle H&B algorithms can be applied to any optimization problem in which it is possible to efficiently bound the evidence. Thus, the key contributions of this article are: 1) to define an evidence function for solving several vision tasks; and 2) to derive tight bounds on the evidence using the theory of semi-discrete shapes of [10]. In what follows we review related work (§2), formally define the problem (§3), describe the BM (§4), show experimental results (§5), and conclude with a discussion (§6).

2. Prior work

The problem of *simultaneous* 3D reconstruction, pose estimation, and object classification from an image has only recently been approached. To the best of our knowledge, the work in [9] is the only one that uses H&B algorithms for solving *exactly* the same problem. There are, however, two main differences between our work and that of [9]. First, we use the mathematical theory of shapes and summaries described in [10] to compute much tighter bounds for the evidence $L(H)$, and hence reduce the computational load. Second, the present model corrects a bias in [9]. Namely that given two similar 3D shapes with equal projection on the camera plane, the framework in [9] "prefers" the shape located further away from the camera, while the current framework has no preference for either shape.

There are a number of works in related areas. [12], for example, proposed a method to simultaneously categorize an object, estimate its pose, and obtain a *crude* 3D reconstruction from a single image. The reconstruction consists of a few planar faces (or "parts") linked together by homographic transformations. [13] and [4], on the other hand, focus on the related problem of *scene* reconstruction from an image. In these works the reconstructed surface contains a planar patch for each superpixel in the input image. The 3D orientation of these patches is inferred from the superpixels' appearance using a probabilistic graphical model.

Since the early days of computer vision many methods have been proposed to reconstruct and estimate the pose of *specific* object classes from an image. These methods, however, only handled somewhat artificial object classes not frequently found in the real world (*e.g.*, polyhedral shapes [7] and generalized cylinders [1]). More recently a number of methods have been proposed to deal with more realistic, but still very specific, object classes (*e.g.*, trees/grasses [3] and people [16, 14]). In general these methods consist of a parametric model of a class, and a procedure to fit the projection of the model to an image. These approaches are best suited to reconstruct objects from the class they were designed for, and are in general difficult to extend to other classes.

In contrast, more general representations that can learn about an object class from exemplars (as our approach

does), can be trained on new classes without having to redesign the representation anew each time. One example of such a general representation is found in [11], which uses a level set formulation and 3D shape priors in order to simultaneously segment an object in an image and estimate its pose (they do not, however, address classification or reconstruction). The shape prior model is a set of principal components learned from the signed distance functions computed from each 3D shape in the training set.

3. Problem formulation

In this section we formally define our problem of interest. Let $f : \Theta \to \mathbb{R}^c$ ($c \in \mathbb{N}$) be an image produced as the noisy projection onto 2D of a single 3D object (Fig. 1a). This object is assumed to belong to a known set of classes. Given this input image and the 3D shape priors (defined later), the problem is to estimate the class K of the object, its 3D pose T, recover a 2D segmentation q of the object in the image, and estimate a 3D reconstruction v of the object in 3D space. The relationships among these variables are depicted in the factor graph of Fig. 1b.

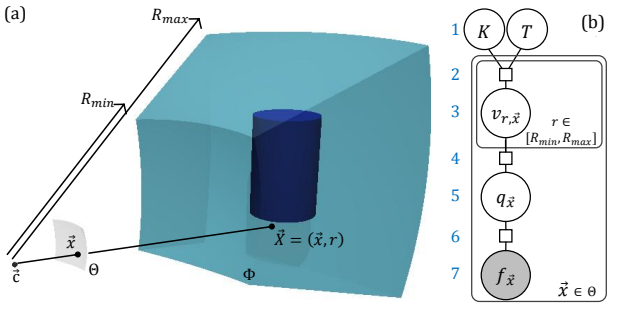

Figure 1. (a) Problem setup. The camera is defined by its center $\vec{c}$ and a patch $\Theta \subset S^2$ in the unit sphere. The set Φ consists of the points $\vec{X} \in \mathbb{R}^3$ that project to Θ and satisfy $R_{min} \leq \|\vec{X} - \vec{c}\| \leq R_{max}$. A single object (represented by the blue cylinder) is assumed to be in Φ. (b) Factor graph [2] proposed to solve the problem. Circles represent *variables* and squares represent *factors* in the system's joint probability. Factors are connected to the variables they contain. Observed variables are shaded. Variables and factors inside a *plate* are instantiated for each element in the set indicated in the lower right corner of the plate.

It follows from the independence assumptions depicted in the graph that the joint probability of f, q, v, and $H \triangleq (K, T)$, is given by

$$p(f, q, v, H) = p(f|q)p(q|v)p(v|H)p(H). \qquad (2)$$

Thus, given f, our goal is to find the values of H, q, and v, that maximize (2). In doing so we will be estimating K, T, q, and v, *simultaneously*. Each term in (2) will be formally defined in §3.2-3.4. In order to do this, in the next section we briefly review the required concepts from [10].

3.1. Review: Shape representations and likelihoods

We represent the shapes q and v as instances of what we call *continuous shapes*, as defined next.

Continuous shape. Given a set $\Omega \subset \mathbb{R}^d$, a set $S \in \Omega$ is a *continuous shape* if: 1) it is open, and 2) its boundary has zero measure. Alternatively, a continuous shape S can also be regarded as the *function* $S : \Omega \to \{0, 1\}$ defined as the indicator function of the *set* S (*i.e.*, $S(\vec{x}) = 1 \iff \vec{x} \in S$).

In order to define the terms $p(f|q)$ and $p(v|H)$ involving the continuous shapes q and v in (2), we define the likelihood of a continuous shape S by extending the definition of the likelihood of a *discrete shape* $\hat{S}$, defined next.

Discrete shape. Given a partition $\Pi(\Omega) = \{\Omega_1, \ldots, \Omega_n\}$ of a set $\Omega \subset \mathbb{R}^d$ (*i.e.*, a collection of sets such that $\bigcup_i \Omega_i = \Omega$, and $\Omega_i \neq \Omega_j$ for $i \neq j$), the *discrete shape* $\hat{S}$ is defined as the function $\hat{S} : \Pi(\Omega) \to \{0, 1\}$. A continuous shape S can be *produced* from a discrete shape $\hat{S}$ (denoted $S \sim \hat{S}$) as $S(\vec{x}) = \hat{S}(\Omega_i)\ \forall \vec{x} \in \Omega_i$, for $i = 1, \ldots, n$.

Let $\hat{B} = \{\hat{B}_1, \ldots, \hat{B}_n\}$ be a family of independent Bernoulli random variables characterized by the success rates $p_{\hat{B}}(i) \triangleq P(\hat{B}_i = 1)$. We call this a *discrete Bernoulli Field* (BF). The log-likelihood $\log P(\hat{B} = \hat{S})$ of the discrete shape $\hat{S}$ according to the discrete BF $\hat{B}$ is computed as

$$\sum_{i=1}^{n} \log P(\hat{B}_i = \hat{S}(\Omega_i)) = Z_{\hat{B}} + \sum_{i=1}^{n} \hat{S}(\Omega_i)\delta_{\hat{B}}(i), \qquad (3)$$

where $Z_{\hat{B}} \triangleq \sum_{i=1}^{n} \log\left(1 - p_{\hat{B}}(i)\right)$ and $\delta_{\hat{B}}(i) \triangleq \log\left(\frac{p_{\hat{B}}(i)}{1 - p_{\hat{B}}(i)}\right)$.

To compute the likelihood of a continuous shape, we first define *continuous* BFs by analogy with *discrete* BFs. A *continuous BF* is a collection of independent Bernoulli random variables $\{B_{\vec{x}}\}$, where a variable $B_{\vec{x}}$ is defined for each point $\vec{x} \in \Omega$ (rather than for each index $i \in \{1, \ldots, n\}$). The success rates of the BF are given by the function $p_B(\vec{x}) \triangleq P(B_{\vec{x}} = 1)$. A continuous BF B can be *produced* from a discrete BF $\hat{B}$ (denoted $B \sim \hat{B}$), by defining its success rates as $p_B(\vec{x}) = p_{\hat{B}}(\Omega_i)\ \forall \vec{x} \in \Omega_i$, for $i = 1, \ldots, n$.

The log-likelihood $\log P(B = S)$ of a continuous shape S according to a continuous BF B, is then defined to be

$$\frac{1}{u_\Omega} \int_\Omega \log P(B_{\vec{x}} = S(\vec{x}))\, d\vec{x} = \frac{1}{u_\Omega}\left[Z_B + \int_\Omega S(\vec{x})\delta_B(\vec{x})\right], \qquad (4)$$

where $Z_B \triangleq \int_\Omega \log\left(1 - p_B(\vec{x})\right) d\vec{x}$, $\delta_B(\vec{x}) \triangleq \log\left(\frac{p_B(\vec{x})}{1 - p_B(\vec{x})}\right)$ and u_Ω is a constant referred to as the *equivalent unit size* (explained below).

The following proposition shows that, under certain conditions, continuous and discrete log-likelihoods (in (4) and (3), respectively) coincide. For this reason we said that (4) *extends* (3).

Proposition 3.1 (Rationale for continuous likelihoods) *Let $\Pi(\Omega)$ be a partition of a set Ω such that $\forall \omega \in \Pi(\Omega)$, $|\omega| = u_\Omega$, and let $\hat{B}$ and $\hat{S}$ be a discrete BF and a discrete shape, respectively, defined on $\Pi(\Omega)$. Finally, let B be a continuous BF and let S be a continuous shape such that $B \sim \hat{B}$ and $S \sim \hat{S}$. Then, the log-likelihoods of the continuous and the discrete shapes are equal,* i.e.*,*

$$\log P(B = S) = \log P(\hat{B} = \hat{S}). \tag{5}$$

Proof. *Immediate from the definitions.* ■

Note that the equivalent unit size u_Ω "scales" the value in brackets in (4) according to the resolution of the partition $\Pi(\Omega)$, making it comparable to (3). We are now ready to define the terms on the right hand side of (2).

3.2. Image term: $p(f|q)$

The 2D segmentation q that we want to estimate (level 5 of Fig. 1b) is represented as a 2D continuous shape defined on the image domain Θ. This segmentation q states whether each point $\vec{x} \in \Theta$ is deemed to be in the `Background` (if $q(\vec{x}) = 0$), or in the `Foreground` (if $q(\vec{x}) = 1$).

We assume that the state $q(\vec{x})$ of a point $\vec{x} \in \Theta$ *cannot* be observed directly, but rather it defines the pdf of a feature $f(\vec{x})$ at $\vec{x}$ that *is* observed (level 7 of Fig. 1b). For example, $f(\vec{x})$ could simply indicate color, depth, texture class, or in general any feature directly observed at $\vec{x}$, or computed from other features observed in the neighborhood of $\vec{x}$.

We also suppose that if a point $\vec{x}$ belongs to the `Background`, its feature $f(\vec{x})$ is distributed according to the pdf $p_{\vec{x}}(f(\vec{x})|q(\vec{x}) = 0)$, while if it belongs to the `Foreground`, $f(\vec{x})$ is distributed according to $p_{\vec{x}}(f(\vec{x})|q(\vec{x}) = 1)$. This feature $f(\vec{x})$ is assumed to be independent of the feature $f(\vec{y})$ and the state $q(\vec{y})$ in every other point $\vec{y} \in \Theta$, given $q(\vec{x})$. In the experiments in §5, a *different* pdf was learned for each point $\vec{x}$ in the background (through background subtraction), and a *single* pdf was learned for all the points in the foreground.

The *image term*, $\log p(f|q)$ (level 6 of Fig. 1b), is thus given by

$$\log p(f|q) \triangleq \frac{1}{u_\Theta} \int_\Theta \log p_{\vec{x}}(f(\vec{x})|q(\vec{x}))\, d\vec{x}, \tag{6}$$

where the equivalent unit size in Θ, u_Θ, is a constant to be defined. If we define a continuous BF B_f with success rates

$$p_{B_f}(\vec{x}) \triangleq \frac{p_{\vec{x}}\left(f(\vec{x})|q(\vec{x}) = 1\right)}{p_{\vec{x}}(f(\vec{x})|q(\vec{x}) = 0) + p_{\vec{x}}(f(\vec{x})|q(\vec{x}) = 1)}, \tag{7}$$

it follows that (6) is equal, up to a constant, to the log-likelihood of the shape q according to the BF B_f, *i.e.*, $\log p(f|q) = \log P(B_f = q) + C_1$. Therefore, using (4), the image term can be written as

$$\log p(f|q) = \frac{1}{u_\Theta}\left[Z_{B_f} + \int_\Theta q(\vec{x}) \delta_{B_f}(\vec{x})\, d\vec{x}\right] + C_1. \tag{8}$$

3.3. 3D shape prior term: $P(v|H)$

While the segmentation q is a *2D continuous shape* on the 2D image domain Θ, the reconstruction v (level 3 of Fig. 1b) is a *3D continuous shape* on the set $\Phi \subset \mathbb{R}^3$. Hence v states whether each point $\vec{X} \in \Phi$ is deemed to be `In` the reconstruction (if $v(\vec{X}) = 1$), or `Out` of it (if $v(\vec{X}) = 0$). The 3D point $\vec{X}$ is expressed in the *world coordinate system* (WCS) defined on the set Φ.

Since our problem of interest is ill posed unless some form of prior knowledge about the shape of the objects is incorporated, we assume that the object class K (level 1 of Fig. 1b) is one out of N_K distinct possible object classes, each one characterized by a 3D shape prior B_K encoding our prior geometric knowledge about the object class. This knowledge is stated with respect to an *intrinsic 3D coordinate system* (ICS) defined for each class. In other words, all the objects of the class are assumed to be in a canonical pose (*i.e.*, normalized) in this ICS.

Each shape prior B_K is encoded as a BF, such that for each point $\vec{X}'$ in the ICS of the class, the success rates $p_{B_K}(\vec{X}') \triangleq P(v'(\vec{X}') = 1|K)$ indicate the probability that the point $\vec{X}'$ would be `In` the 3D reconstruction v' defined in the ICS, given the class K of the object (Fig. 2).

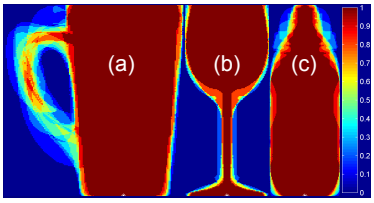

Figure 2. Vertical cuts through the 3D BFs corresponding to the object classes mugs (a), cups (b), and bottles (c). Colors indicate the success rates $p_{B_K}(\vec{X}')$.

In order to translate the quantities v' and p_{B_K} (defined in the ICS), to the corresponding quantities v and p_{B_H} (defined in the WCS), we define the transformation T that maps each point $\vec{X}'$ in the ICS to the point $\vec{X} \triangleq T(\vec{X}')$ in the WCS (level 1 of Fig. 1b). This transformation, referred to as the *pose*, is an unknown to be estimated. Thus, we can now write the entities in the WCS in terms of those in the ICS, as $v(\vec{X}) = v'(T^{-1}(\vec{X}))$ and $p_{B_H}(\vec{X}) \triangleq P(v(\vec{X}) = 1|K, T) = p_{B_K}(T^{-1}(\vec{X}))$ (level 2 of Fig. 1b).

Therefore, from (4), the *shape prior term*, $\log p(v|H)$, is given by

$$\log p(v|H) \triangleq \frac{1}{u_\Phi(T)} \int_\Phi \log p_{B_H}(\vec{X})\, d\vec{X} = \frac{1}{u_\Phi(T)}\left[Z_{B_H} + \int_\Phi v(\vec{X}) \delta_{B_H}(\vec{X})\, d\vec{X}\right], \tag{9}$$

where $u_\Phi(T)$ is the equivalent unit size in Φ. We define this quantity to be $u_\Phi(T) \triangleq |J(T)|/\lambda$, where $|J(T)|$ is the Jacobian of the transformation T, and $\lambda > 0$ is an arbitrary constant. This will prevent the system from having a bias towards either smaller objects closer to the camera, or bigger objects farther from the camera (see details in [8]).

3.4. Projection term: $P(q|v)$

The segmentation q and the reconstruction v are certainly related, as we expect q to be (at least) "close" to the projection of v on Θ. Thus, the *projection term* $p(q|v)$ (level 4 of Fig. 1b) encodes the probability of obtaining a segmentation q in Θ, given that a reconstruction v is present in Φ. In order to define this term, we first need to understand the relationship between the sets Θ and Φ (Fig. 1a).

Points in the 3D space Φ are mapped by the *camera transformation* into the 2D camera retina Θ (we consider a spherical retina, *i.e.*, $\Theta \subset S^2$). Given a point $\vec{c} \in \mathbb{R}^3$, referred to as the *camera center*, and a point $\vec{x} \in \Theta$, the set of points $R(\vec{x}) \triangleq \{\vec{X} \in \Phi : \vec{X} = \vec{c} + r\vec{x},\ r \in [0, \infty)\}$ *projects* to $\vec{x}$ and is referred to as the *ray* of $\vec{x}$. We will often refer to points $\vec{X} \in \Phi$ by their projection $\vec{x}$ in Θ and their distance r from the camera center, as in $\vec{X} = (\vec{x}, r)$. The set Φ can thus be formally defined as $\Phi \triangleq \{(\vec{x}, r) \in \mathbb{R}^3 : \vec{x} \in \Theta, R_{min} \leq r \leq R_{max}\}$.

We can now define the projection term. To reflect our intuition, we want it to have the following properties:

1. $p(q(\vec{x})|v)$ should depend only on the state of v in $R(\vec{x})$, denoted as $v_{R(\vec{x})}$, and not on the state of v in other points of Φ. This follows from the camera projection model.
2. $p(q(\vec{x}) = 1|v_{R(\vec{x})})$ should be higher when the measure of the set $v_{R(\vec{x})}$ is greater. This property says that reconstructions that are intuitively "better" will be assigned higher probabilities.
3. The projection term should be *scale invariant*, meaning that it remains agnostic as to whether the object in the scene is small and close or large and far. This simply reflects the fact that (standard) cameras cannot estimate absolute size.

These properties are satisfied by the following projection term (see a formal derivation of this term in [8]),

$$\log p(q(\vec{x})|v) \triangleq \log p(q(\vec{x})|\ell_v(\vec{x})) = \begin{cases} \alpha \ell_v(\vec{x}), & \text{if } q(\vec{x}) = 0, \\ \log\left(1 - e^{\alpha \ell_v(\vec{x})}\right), & \text{if } q(\vec{x}) = 1, \end{cases} \tag{10}$$

where

$$\ell_v(\vec{x}) \triangleq \int_0^\infty \frac{v(\vec{c} + r\vec{x})}{r}\, dr \tag{11}$$

is a measure of the "mass" in the ray $R(\vec{x})$, and $\alpha < 0$ is a constant to be determined. Therefore, the projection term is given by

$$\log p(q|v) \triangleq \frac{1}{u_\Theta} \int_\Theta \log p\left(q(\vec{x})|\ell_v(\vec{x})\right) d\vec{x}, \tag{12}$$

where u_Θ is the equivalent unit size in (8).

3.5. Definition of the evidence: $L(H)$

In this section we put together the image term, the shape prior term and the projection term, derived in previous sections, to obtain an expression for the evidence $L(H)$ of a hypothesis H. Substituting (8), (9), and (12) into (2), we get an expression for the joint log-probability of the system, $\log p(f, q, v, H)$. Our goal can now be formally stated as to solve $\sup_{q,v,H} \log p(f, q, v, H)$, which is equivalent to solving $\max_H L'(H)$ with

$$L'(H) \triangleq \sup_{q,v} \left[\log p(f, q, v, H)\right]. \tag{13}$$

Instead of computing (13) directly, however, we will first derive an expression that is equal to it (up to a constant and a change of scale), but is simpler to work with.

Hence, we disregard the terms C_1 and Z_{B_f}/u_Θ in $\log p(f, q, v, H)$ that do not depend on H, q, or v, disregard the term $\log P(H)$ that is assumed to be equal for all hypotheses, rearrange terms, multiply by u_Θ, and substitute the result in (13), to arrive at the final expression for $L(H)$,

$$L(H) \triangleq \sup_{q,v} \left\{ \int_\Theta \left[q(\vec{x}) \delta_{B_f}(\vec{x}) + \log P(q(\vec{x})|\ell_v(\vec{x})) \right. \right. \tag{14}$$
$$\left. \left. + \frac{u_\Theta}{u_\Phi(T)} \int_{R_{min}}^{R_{max}} r^2 v(\vec{x}, r) \delta_{B_H}(\vec{x}, r)\, dr \right] d\vec{x} + \frac{u_\Theta Z_{B_H}}{u_\Phi(T)} \right\}.$$

In the process of finding the hypothesis $H = (K, T)$ that maximizes $L(H)$ in (14), a segmentation q and a reconstruction v are also obtained. Such a reconstruction v is "a compromise" between the shape prior of the estimated class K transformed by the estimated pose T, and the observed features in the image f. As explained before, however, computing the evidence $L(H)$ for each hypothesis H using (14) would be prohibitively expensive. For this reason we compute bounds for it and use a H&B algorithm, instead, to select the best hypothesis. In the next section we describe the mechanism to compute those bounds.

4. Bounding mechanism

We will now show how to efficiently compute bounds for $L(H)$ in (14). To compute these bounds we will rely on a special type of partition, the *standard partition*, and the concept of *summaries* introduced in [10]. Before deriving the bounds, we define these two concepts and prove a lemma.

4.1. Pre-requisites

Standard partition. Let $\Theta \subset S^2$, $\mathcal{R} \triangleq [R_{min}, R_{max}]$ and $\Phi = \Theta \times \mathcal{R} \subset \mathbb{R}^3$. Let $\Pi(\Theta) = \{\Theta_1, \ldots, \Theta_{N_\Theta}\}$ be a partition of Θ and let $\Pi(\mathcal{R}) = \{[r_0, r_1), \ldots, [r_{N_r - 1}, r_{N_r})\}$ be a partition of $\mathcal{R}$ with $r_0 \triangleq R_{min}$, $r_{N_r} \triangleq R_{max}$ and $r_i \triangleq \beta r_{i-1} = \beta^i r_0$ $(\beta > 1)$. The *standard partition* for (Θ, Φ) is defined as $(\Pi(\Theta), \Pi(\Phi))$, where $\Pi(\Phi) \triangleq \Pi(\Theta) \times \Pi(\mathcal{R}) = \{\Phi_{1,1}, \Phi_{1,2}, ..., \Phi_{N_\Theta, N_r}\}$ and $\Phi_{j,i} \triangleq \Theta_j \times [r_{i-1}, r_i)$ (Fig. 3).

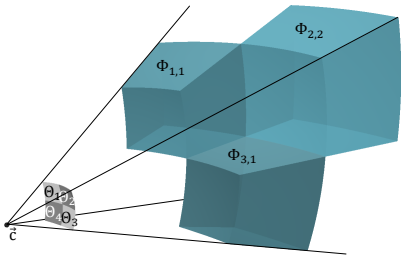

Figure 3. Example of a standard partition for (Θ, Φ), with $\Pi(\Theta) = \{\Theta_1, \Theta_2, \Theta_3, \Theta_4\}$ and $\Pi(\mathcal{R}) = \{[R_{min}, r_1), [r_1, R_{max})\}$. For clarity only the voxels $\Phi_{1,1}$, $\Phi_{2,2}$, $\Phi_{3,1} \in \Pi(\Phi)$ are shown.

Mean-summary. Given a BF B defined on a set Ω and a partition $\Pi(\Omega)$ of this set, the *mean-summary* is the functional $\hat{Y}_B = \{\hat{Y}_{B,\omega}\}_{\omega \in \Pi(\Omega)}$ that assigns the value

$$\hat{Y}_{B,\omega} \triangleq \int_{\omega} \delta_B(\vec{x})\, d\vec{x} \tag{15}$$

to each set $\omega \in \Pi(\Omega)$. The "infinite dimensional" BF is thus "summarized" by just $n \triangleq |\Pi(\Omega)|$ values.

Mean-summaries have two important properties: 1) for certain kinds of sets $\omega \in \Pi(\Omega)$, the values $\hat{Y}_{B,\omega}$ in the summary can be computed in constant time, regardless of the "size" of the sets ω (using integral images [15]); and 2) they can be used to obtain a lower bound for the evidence. We prove next a result needed to obtain this lower bound.

Lemma 4.1 (Mean-summary identity) *Let $\Pi(\Omega)$ be a partition of a set Ω, let $\hat{S}$ be a discrete shape defined on $\Pi(\Omega)$, let B be a BF on Ω, and let $\hat{Y}_B = \{\hat{Y}_{B,\omega}\}$ ($\omega \in \Pi(\Omega)$) be the* mean-*summary of B in $\Pi(\Omega)$. Then, for any continuous shape $S \sim \hat{S}$, it holds that*

$$\int_{\Omega} \delta_B(\vec{x}) S(\vec{x})\, d\vec{x} = \sum_{\omega \in \Pi(\Omega)} \hat{S}(\omega) \hat{Y}_{B,\omega}. \tag{16}$$

Proof. *Immediate from the definitions.* ■

4.2. Derivation of the lower bound

In the next theorem we derive an expression to bound $L(H)$ from below. This derivation is based on the observation that the set of continuous shapes that are equivalent to *any* discrete shape defined on a partition Π, denoted as $\mathbb{S}_\Pi$, is a subset of the set of all continuous shapes $\mathbb{S}$ (*i.e.*, $\mathbb{S}_\Pi \subset \mathbb{S}$). Hence, the supremum in (14) for $q, v \in \mathbb{S}_\Pi$, is less than the supremum for $q, v \in \mathbb{S}$. Moreover, since the continuous shapes in $\mathbb{S}_\Pi$ are constant inside each partition element, evaluating this new supremum is easier.

Theorem 4.2 (Lower bound for $L(H)$) *Let $\mathbf{\Pi} \triangleq (\Pi(\Theta), \Pi(\Phi))$ be a standard partition, and let $\hat{Y}_f = \left\{\hat{Y}_{f,\theta}\right\}_{\theta \in \Pi(\Theta)}$ and $\hat{Y}_H = \left\{\hat{Y}_{H,\phi}\right\}_{\phi \in \Pi(\Phi)}$ be the* mean-*summaries of two unknown BFs in $\Pi(\Theta)$ and $\Pi(\Phi)$, respectively. Let $\psi_{j,k}$ be the set of the indices of the k largest elements of $\left\{\hat{Y}_{H,\Phi_{j,1}}, \hat{Y}_{H,\Phi_{j,2}}, \ldots, \hat{Y}_{H,\Phi_{j,N_r}}\right\}$, and let $\Psi_{j,k}$ be the sum of these elements,* i.e., *$\Psi_{j,k} \triangleq \sum_{i \in \psi_{j,k}} \hat{Y}_{H,\Phi_{j,i}}$. Then, for all BFs B_f with summary Y_f, denoted $B_f \sim \hat{Y}_f$, and for all BFs $B_H \sim \hat{Y}_H$, it holds that $L(H) \geq \underline{L}_{\mathbf{\Pi}}(H)$, where*

$$\underline{L}_{\mathbf{\Pi}}(H) \triangleq \frac{u_\Theta Z_{B_H}}{u_\Phi(T)} + \sum_{j=1}^{N_\Theta} \underline{\mathcal{L}}_{\Theta_j}(H), \quad \textit{and} \tag{17}$$

$$\underline{\mathcal{L}}_{\Theta_j}(H) \triangleq \max_{0 \leq \mathsf{n}_j \leq N_r} \left\{ \max_{\mathsf{q} \in \{0,1\}} \left[\mathsf{q} \hat{Y}_{f,\Theta_j} + |\Theta_j| \log P\left(\mathsf{q} | \mathsf{n}_j \log \beta\right) \right] + \frac{u_\Theta}{u_\Phi(T)} \Psi_{j,\mathsf{n}_j} \right\}. \tag{18}$$

Moreover, the 3D reconstruction and the 2D segmentation corresponding to this bound are given by the discrete shapes $\hat{v}$ and $\hat{q}$, respectively, defined by

$$\hat{v}(\Phi_{j,i}) \triangleq \begin{cases} 1, & \textit{if } i \in \psi_{j,\mathsf{n}_j^*}, \\ 0, & \textit{otherwise}, \end{cases} \quad \textit{and} \tag{19}$$

$$\hat{q}(\Theta_j) \triangleq \underset{\mathsf{q} \in \{0,1\}}{\arg\max} \left[\mathsf{q} \hat{Y}_{f,\Theta_j} + |\Theta_j| \log P\left(\mathsf{q} | \mathsf{n}_j^* \log \beta\right) \right], \tag{20}$$

where n_j^* is the solution to (18).

Proof. *Since the sets of continuous shapes q and v that are, respectively, equivalent to the discrete shapes $\hat{q}$ and $\hat{v}$ are subsets of the sets of all continuous shapes, it holds that $L(H)$ (defined in (14)) is greater than or equal to*

$$\frac{u_\Theta Z_{B_H}}{u_\Phi(T)} + \sup_{\hat{q}, \hat{v}} \left\{ \sup_{\substack{q \sim \hat{q} \\ v \sim \hat{v}}} \int_\Theta \left[\delta_{B_f}(\vec{x}) q(\vec{x}) + \frac{u_\Theta}{u_\Phi(T)} \int_{R_{min}}^{R_{max}} r^2 v(\vec{x}, r) \delta_{B_H}(\vec{x}, r)\, dr + \log P\left(q(\vec{x}) | \ell_v(\vec{x})\right) \right] d\vec{x} \right\}. \tag{21}$$

Since $q \sim \hat{q}$ and $v \sim \hat{v}$, it follows from Lemma 4.1 that

$$\int_\Theta \delta_{B_f}(\vec{x}) q(\vec{x})\, d\vec{x} = \sum_{j=1}^{N_\Theta} \hat{q}\left(\Theta_j\right) \hat{Y}_{f,\Theta_j}, \quad \textit{and} \tag{22}$$

$$\int_\Theta \int_{R_{min}}^{R_{max}} r^2 v(\vec{x}, r) \delta_{B_H}(\vec{x}, r)\, dr\, d\vec{x} = \sum_{j=1}^{N_\Theta} \sum_{i=1}^{N_r} \hat{v}\left(\Phi_{j,i}\right) \hat{Y}_{H,\Phi_{j,i}}. \tag{23}$$

On the other hand, $\ell_v(\vec{x})$ is constant inside each element of $\Pi(\Theta)$, because $\forall \vec{x} \in \Theta_j$,

$$\ell_v(\vec{x}) = \int_{r_0}^{r_{N_r}} \frac{v(\vec{x}, r)}{r}\, dr = \sum_{i=1}^{N_r} \hat{v}\left(\Phi_{j,i}\right) \log\left(\frac{r_i}{r_{i-1}}\right) = \log(\beta) \sum_{i=1}^{N_r} \hat{v}\left(\Phi_{j,i}\right). \tag{24}$$

Then, substituting (22), (23) *and* (24) *into* (21) *yields*

$$\frac{u_\Theta Z_{B_H}}{u_\Phi(T)} + \max_{\hat{q},\hat{v}} \sum_{j=1}^{N_\Theta} \Bigg[\hat{q}(\Theta_j)\, \hat{Y}_{f,\Theta_j} + \\ + |\Theta_j| \log P\left(\hat{q}(\Theta_j) \,|\, \log(\beta) \sum_{i=1}^{N_r} \hat{v}(\Phi_{j,i}) \right) + \\ + \frac{u_\Theta}{u_\Phi(T)} \sum_{i=1}^{N_r} \hat{v}(\Phi_{j,i})\, \hat{Y}_{H,\Phi_{j,i}} \Bigg]. \quad (25)$$

Note that the first (leftmost) term inside the square brackets in (25) *does not depend on the states of the voxels* $\hat{v}(\Phi_{j,i})$*, the third term does not depend on the state of the pixel* $\hat{q}(\Theta_j)$*, and the second term does not depend on which particular voxels are full, but on the number of full voxels* $\mathsf{n}_j \triangleq \sum_{i=1}^{N_r} \hat{v}(\Phi_{j,i})$*. In contrast, the third term does depend on which voxels are full, and given the number* n_j *of full voxels, the third term will be maximum when the* n_j *voxels with the largest summary* $\hat{Y}_{H,\Phi_{j,i}}$ *are full,* i.e.*, it will be equal to* Ψ_{j,n_j}*. Therefore* (25) *is equal to,*

$$\frac{u_\Theta Z_{B_H}}{u_\Phi(T)} + \sum_{j=1}^{N_\Theta} \max_{0\le \mathsf{n}_j \le N_r} \Bigg\{ \max_{\mathsf{q}\in\{0,1\}} \Big[\mathsf{q}\hat{Y}_{f,\Theta_j} + \\ |\Theta_j| \log P(\mathsf{q}|\mathsf{n}_j \log\beta) \Big] + \frac{u_\Theta}{u_\Phi(T)} \Psi_{j,\mathsf{n}_j} \Bigg\}, \quad (26)$$

which can be rearranged to yield (17) *and* (18)*. It is then easy to see that the discrete shapes* $\hat{v}$ *and* $\hat{q}$ *(defined respectively in* (19) *and* (20)*) maximize* (18)*.* ■

Note that this lower bound can be computed in order $O(N_\Theta N_r \log N_r)$, and that it can be refined by refining the corresponding standard partition.

4.3. Derivation of the upper bound

A formula to compute an upper bound for $L(H)$ is derived following a path similar to that in Theorem 4.2: a standard partition is defined, summaries of the BFs B_f and B_H are computed inside each partition element, and these summaries are then used to compute the upper bound. However, a different kind of summaries, namely m-summaries, is used to derive the upper bound. Since the derivation of the upper bound is longer and much more involved than the one for the lower bound, we refer the reader to [8] for the derivation and proofs.

5. Experimental results

In this section we show results obtained with the framework described in previous sections. To demonstrate its unique characteristics, we start with a simple experiment where a *known* object (a bottle) is present in the input image

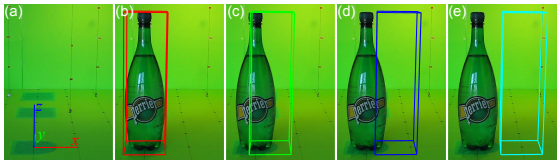

Figure 4. (a) Coordinate axes in the WCS: axes start at the origin and are 10cm long. (b-e) Four hypotheses proposed to "explain" the (same) input image. The support of $p_{B_H}(\vec{X})$ is indicated in each case by the overlaid 3D box.

and our goal is only to estimate its pose. For this purpose we define a set $\mathbb{H}_1$ containing 6771 hypotheses around the true hypothesis, produced by combining 61 translations t_x in the x direction with 111 translations t_y in the y direction, sampled every 0.5cm (see the 3D coordinate axes in Fig. 4a). Fig. 4b-e shows 4 hypotheses from $\mathbb{H}_1$.

We then use the proposed framework to find the hypotheses that maximize $L(H)$. The evolution of the bounds for the four hypotheses of Fig. 4b-e only are depicted in Fig. 5.

Fig. 5 shows that after 22 refinement cycles the red hypothesis is proved to be optimal. When we run the framework on the whole set $\mathbb{H}_1$, however, the set of optimal hypotheses, denoted as $\mathbb{A}$, contained 19 hypotheses. To quantify the quality of this set we define, for each parameter t of the transformations, the *bias of* t and the *standard deviation of* t, respectively as

$$\mu_t \triangleq \frac{1}{|\mathbb{A}|} \sum_{H\in\mathbb{A}} t_H - t_T \quad \text{and} \quad \sigma_t \triangleq \sqrt{\frac{1}{|\mathbb{A}|} \sum_{H\in\mathbb{A}} (t_H - t_T)^2}, \quad (27)$$

where t_H is the parameter corresponding to $H \in \mathbb{A}$, and t_T is the true value of the parameter. The results for this experiment are summarized in Table 1.

Table 1. Pose estimation results for a *known, symmetric* object.

$\lvert\mathbb{A}\rvert$	μ_{t_x} (cm)	σ_{t_x} (cm)	μ_{t_y} (cm)	σ_{t_y} (cm)
19	-0.24	0.34	0.61	1.54

Fig. 6 shows how the computation is distributed among the hypotheses. It can be seen in Fig. 6a that most hypotheses (92.2%) only require 0/1 refinement cycles, while only a few (0.41%) had to be fully refined. This is the source of the computational gain of our algorithm. Fig. 6b shows that

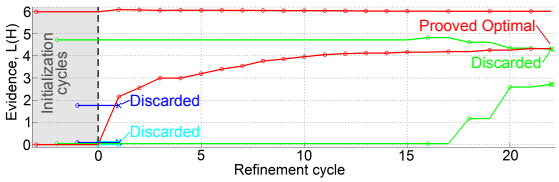

Figure 5. Evolution of the bounds of the four hypotheses shown in Fig. 4b-e. The bounds of one hypothesis are represented by the two lines of the same color, which is also the color of the hypothesis in Fig. 4b-e. Cycles in which a hypothesis is selected for refinement or discarded are indicated by '$\circ$' and '$\times$', respectively.

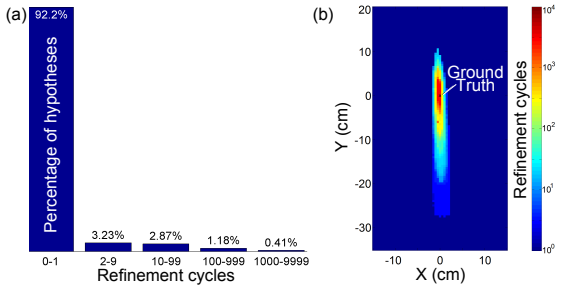

Figure 6. (a) Histogram of refinement cycles allocated per hypothesis. (b) Refinement cycles allocated to each hypothesis in $\mathbb{H}_1$ (note the logarithmic scale).

the hypotheses that require most computation surround the ground truth, however, not isotropically: hypotheses in the ground truth's line of sight are harder to distinguish from it (compare σ_{t_x} and σ_{t_y} in Table 1).

Next we look at the *average* perfromance of the framework on a set of images. These images contained objects from one of five classes: mugs, glasses, cups, bottles or plates. The hypothesis space for this experiment, $\mathbb{H}_2$, contained $45,675$ hypotheses, obtained by combining 5 different classes, translations t_x and t_y in the x and y directions, respectively, rotations ϕ_z around the vertical z axis (for the non-rotationally symmetric class 'mugs' only), uniform scalings s_{xy} in the $x-y$ directions, and scalings s_z in the z direction. The pose estimation errors for this experiment are summarized in Table 2.

Table 2. Pose estimation results for *unknown* objects.

$\lvert\mathbb{A}\rvert$	μ_{t_x} (cm)	σ_{t_x} (cm)	μ_{t_y} (cm)	σ_{t_y} (cm)	μ_ϕ (°)	σ_ϕ (°)	$\mu_{s_{xy}}$ (%)	$\sigma_{s_{xy}}$ (%)	μ_{s_z} (%)	σ_{s_z} (%)
43	0.1	0.3	0.9	2.0	19	48	4.8	8.4	5.7	9.3

In this experiment the class of the object was estimated as well as the pose. In *all cases* the correct class was included in the set $\mathbb{A}$. Apart from hypotheses of this class, the set $\mathbb{A}$ contained 7% of hypotheses from other classes (in 97% of these confusions the class 'glasses,' the one with least training exemplars, was confused with the classes 'mugs' or 'bottles'). Some examples of the 3D reconstructions obtained for the best hypotheses are shown in Fig. 7. The last column shows one case where the best hypothesis is of the wrong class (a bottle instead of a glass). This concludes the presentation of the experiments.

6. Conclusions

This article presented an inference framework to simultaneously tackle the problems of 3D reconstruction, pose estimation and object classification, from a single input image. We showed preliminary results indicating that it is possible to accurately estimate the class and pose of unknown objects, and obtain their 3D reconstructions. Moreover, by deriving tighter bounds than in [9], we were able to reduce the running time of the method and thus handle hypothesis spaces with more degrees of freedom.

Figure 7. 3D reconstructions obtained for the best hypothesis: (1^{st} row) input images; (2^{nd} and 3^{rd} rows) two views of each reconstruction, from two orthogonal directions.

References

[1] T. O. Binford. Visual perception by computer. *Proc. IEEE Conf. on Systems and Control*, 1971. 2

[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006. 3

[3] F. Han and S. C. Zhu. Bayesian reconstruction of 3D shapes and scenes from a single image. *Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis*, 2003. 2

[4] D. Hoiem, A. Efros, and M. Hebert. Automatic photo pop-up. *ACM SIGGRAPH*, 2005. 2

[5] F. D. la Torre and M. J. Black. A framework for robust subspace learning. *IJCV*, 2003. 1

[6] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. *CVPR*, 2004. 1

[7] L. G. Roberts. Machine perception of three-dimensional solids. In *Optical and Electrooptical Information Processing*. MIT Press, 1965. 2

[8] D. Rother, S. Mahendran, and R. Vidal. Hypothesize and bound: A computational focus of attention mechanism for simultaneous 3D shape reconstruction, pose estimation and classification from a single 2D image. www.cis.jhu.edu/~diroth/Research/ICCVsm.pdf, 2011. 4, 5, 7

[9] D. Rother and G. Sapiro. Seeing 3D objects in a single 2D image. *ICCV*, 2009. 2, 8

[10] D. Rother, S. Schütz, and R. Vidal. Hypothesize and bound: A computational focus of attention mechanism for simultaneous n-d segmentation, pose estimation and classification using shape priors. arXiv:1104.2580v2, 2011. 2, 3, 5

[11] R. Sandhu, S. Dambreville, A. Yezzi, and A. Tannenbaum. Non-rigid 2D-3D pose estimation and 2D image segmentation. *CVPR*, 2009. 3

[12] S. Savarese and L. Fei-Fei. 3D generic object categorization, localization and pose estimation. *ICCV*, 2007. 2

[13] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3D scene structure from a single still image. *PAMI*, 2008. 2

[14] L. Sigal and M. J. Black. Predicting 3D people from 2D pictures. *Conf. on Articulated Motion and Deformable Objects*, 2006. 2

[15] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *CVPR*, 2001. 6

[16] J. J. L. Wang and S. Singh. Video analysis of human dynamics - a survey. *Real Time Imaging*, 2003. 2