Seeing 3D Objects in a Single 2D Image

Diego Rother
Johns Hopkins University
diroth@gmail.com

Guillermo Sapiro
University of Minnesota
guille@umn.edu

Abstract

A general framework simultaneously addressing pose estimation, 2D segmentation, object recognition, and 3D reconstruction from a single image is introduced in this paper. The proposed approach partitions 3D space into voxels and estimates the voxel states that maximize a likelihood integrating two components: the object fidelity, that is, the probability that an object occupies the given voxels, here encoded as a 3D shape prior learned from 3D samples of objects in a class; and the image fidelity, meaning the probability that the given voxels would produce the input image when properly projected to the image plane. We derive a loop-less graphical model for this likelihood and propose a computationally efficient optimization algorithm that is guaranteed to produce the global likelihood maximum. Furthermore, we derive a multi-resolution implementation of this algorithm that permits to trade reconstruction and estimation accuracy for computation. The presentation of the proposed framework is complemented with experiments on real data demonstrating the accuracy of the proposed approach.

1. Introduction

A unified framework to address the problems of pose estimation, 2D segmentation, object recognition, and 3D reconstruction from a single image is introduced in this paper. At the core of the framework lies a probabilistic *graphical model* that integrates the information in the input image, with prior 3D knowledge about the class of the object (e.g., "mugs," or "cups"), and with its pose (i.e., the transformation that maps the object to its current pose). The prior 3D knowledge about the object's class is encoded in the form of a *3D shape prior*, e.g., [1], a simple probabilistic representation of the distribution of mass of a class of objects in 3D space.

Given a class of objects and a pose (what we call a *hypothesis* about the state of the world), we derive an extremely efficient algorithm to maximize the log-likelihood of this graphical model, and prove that this algorithm returns the globally optimal solution. The maximum log-likelihood provides an estimate of the suitability of the hypothesis as an explanation for the input image. In addition to the maximum log-likelihood, the proposed optimization algorithm yields a 3D reconstruction of the object in the scene, and a 2D segmentation of its silhouette in the image plane. The 3D reconstruction is represented by the states (*Empty* or *Full*) of the voxels into which the 3D space is partitioned, while the 2D segmentation is represented by the states (*Background* or *Foreground*) of the pixels in the image plane.

In general the true hypothesis (i.e., the class and pose of the object) is not known, and must be estimated as part of the framework. Towards this end we show that using a slightly modified version of the basic algorithm, bounds for a hypothesis' log-likelihood can be obtained at lower resolutions with significantly less computations. Exploiting these bounds, we propose a *branch and bound* strategy to efficiently sift through the hypotheses, selecting the guaranteed optimal.

The remainder of this paper is organized as follows. Section 2 places the current work in the context of prior relevant work. Section 3 introduces the proposed graphical model and the efficient inference algorithms. Section 4 presents experimental results obtained with the proposed framework, and Section 5 concludes with a discussion of the key contributions and directions for future research. Additional details, including the computer code and data used, videos, and proofs of the stated theoretical results, are included as supplemental material.

2. Prior work

Three-dimensional reconstruction from a single image is an ill posed problem. Therefore, all approaches to solve this problem must rely on some form of prior knowledge about the scene or object to be reconstructed. These approaches differ mainly on the *representation* that is selected for the reconstruction, and the *encoding scheme* used for the prior knowledge. Three main representations for the reconstruction have been proposed: model-based, surface-based, and volumetric representations.

Model-based representations, in general, consist of a parametric model of the class of objects to be represented. Reconstructions are obtained by finding the parameters of the model that produce the best fit between the projection of the model, and the input image. When it relies on model-based representations, the 3D reconstruction problem is also referred to as model-based tracking (e.g., [2-4]), or pose estimation (e.g., [5]). For the class of "walking people," undoubtedly the most widely studied class, many models have been proposed (see [6] for a survey), such as articulated bodies [2, 5, 7-8], generalized cylinders [4], silhouettes [3], and models capable of producing (at a higher computational cost) visually appealing reconstructions [9-10]. Model-based representations have also been proposed to reconstruct objects from other classes, such as polyhedra [11], and trees/grasses [11].

Model-based representations are best suited to represent objects for the particular class they were designed for, and

2009 IEEE 12th International Conference on Computer Vision (ICCV)
978-1-4244-4419-9/09/$25.00 ©2009 IEEE

are difficult to extend to other classes. Prior knowledge in this case is encoded in the *design* of the model (e.g., which parts an articulated model has, and how they are connected). This is the main source of difficulty preventing the extension of these models to other classes beyond the one they were designed for. In contrast, more general representations (volumetric or surface-based) that can learn about a class of objects from exemplars (as our approach does), can be trained on new classes without having to "redesign" the representation anew each time.

Surface-based representations for reconstruction typically model the object's surface as a polygonal mesh or triangulated surface. The recent works by Saxena *et al.* [12] and Hoiem *et al.* [13] are examples of the use of this representation on the related problem of *scene* reconstruction from a single image. In these works, a planar patch in the reconstructed surface is defined for each superpixel in the input image. The 3D orientation of these patches is inferred using a learned probabilistic graphical model that relates these orientations to features of the corresponding superpixels. Prior knowledge in this case is encoded in the learned relationship between superpixel features and patch 3D orientations. While the particular representations chosen by Saxena *et al.* and Hoiem *et al.* might be well suited to represent a scene, they are not well suited to represent objects, since only one side of the object can be represented with them. In contrast, our method is well suited to represent bounded objects (e.g., mugs, cars, etc.), but it is not well suited to represent entire elaborated scenes (in particular outdoor scenes, on which the methods mentioned above excel).

The third kind of approach for reconstruction relies on *volumetric representations*. In these representations, 3D space (or a smaller volume containing the whole object) is partitioned into voxels, and the reconstruction is given by the set of *Full* voxels. This set is estimated as the set that best fits the input image according to some metric (see for example [14] for a review of methods that use *multiple* images). The work by Snow *et al*. [15] is perhaps the closest to ours among the volumetric approaches, since it defines a prior on possible reconstructions. However the prior used by Snow *et al.* simply rewards smoothness and has no further information about the object class itself, as our approach does. An important additional difference is that Snow *et al.* use data from *multiple* views.

The present work is also related to the work by Franco and Boyer [16], since both define a probabilistic graphical model that relates a volumetric representation to the observed images. Our work improves upon that work on a number of critical aspects. Firstly, Franco and Boyer do not consider prior knowledge in the reconstruction (as we do), limiting their method to *multi*-view reconstructions. Secondly, we use a more accurate law to transform voxel states into pixel states, which explicitly models the interaction between voxels in the same ray, and is invariant to the voxel grid resolution (Section 3.2). Lastly, we are able to efficiently compute the voxel states (the reconstruction) that *jointly* maximize the likelihood, while Franco and Boyer only compute the voxel states that *independently* maximize the marginals.

To the best of our knowledge, ours is the only framework that uses a *volumetric representation*, augmented with *class-specific prior knowledge*, to efficiently solve the problems of pose estimation, 2D segmentation, object recognition, and 3D reconstruction from a single image, with a guarantee of convergence to the global optimum.

3. Definitions, models, and algorithms

This section presents the key components and variables of the proposed statistical graphical model, together with the inference algorithms developed to solve the problems outlined above.

3.1. Definition of system variables

The main goal of this work is to obtain a 3D reconstruction of an object from a single image of the object. This 3D reconstruction, the output of our system, is represented by the states $V = \{v_i\}_{i=1}^{M}$ ($v_i \in \{Empty, Full\}$) corresponding to the regions of space called *voxels*. Along with the 3D reconstruction, the proposed system produces a segmentation (or labeling) of the input image into foreground and background. The segmentation is given by the states $Q = \{q_j\}_{j=1}^{N}$ ($q_j \in \{Background, Foreground\}$) of the pixels in the image. The third kind of random variables in the system consists of the pixel values $C = \{c_j\}_{j=1}^{N}$ on the input image. In this work these values are assumed to be in RGB color space, but many other pixelwise features could be used instead.

The geometric relationship between pixels and voxels is determined by the *camera matrix*, the second input required by our system (Figure 1). We do not adopt the traditional definition of cubic voxels. Instead, voxels are defined to be the 3D space between concentric and equispaced spherical shells, that project to single pixels (see Figure 1). All the

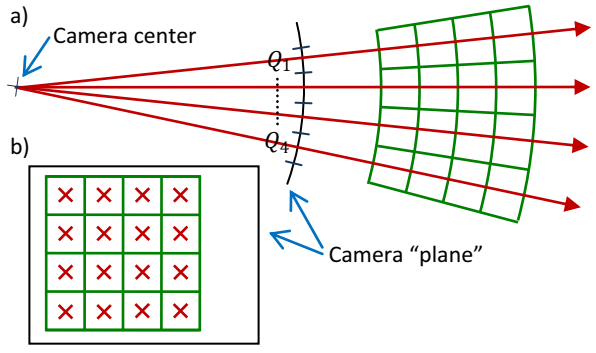

Figure 1: *Definitions of pixels, voxels, and rays, and geometric relationship among them in a plane normal (a) and parallel (b) to the camera plane. Voxels (in green) are aligned with the pixels so that their projections perfectly coincide, and they are delimited on the remaining two sides by concentric and equispaced spherical shells (centered at the camera center). A ray originates at the camera center and passes through a pixel and its corresponding voxels (ray centers are marked in red).*

voxels that project to a pixel j are referred to as pixel j's voxels, and are denoted by the set $R(j)$. The *ray* of a pixel is the 3D space in the solid angle subtended by the pixel. A pixel and all its voxels are said to share a ray. Two important properties follow from these definitions: firstly, all voxels are "seen" at a single pixel in the camera plane; and secondly, the length r of all the intersections between voxels and the rays' center lines, are equal. These properties will be used in the next section to simplify the inference.

The factor graph [17] in Figure 2 depicts the three kinds of random variables introduced above (shown as circles) and the independence assumptions among them. The color c_j on the j-th pixel (in level 6 in Figure 2) is assumed to depend only on the pixel state q_j (in level 4), which in turn depends only on the states of its voxels (in level 2).

The third and last inputs required by the system are the *color models* for the background and foreground, $p(c_j|q_j = B)$ and $p(c_j|q_j = F)$, respectively. These models are represented by the factor in level 5 of Figure 2. The background model for a pixel is the normal distribution with parameters estimated from a video of the background alone, while the foreground model for a pixel, since no information regarding the foreground is available, is assumed to be the uniform distribution. While these simple color models proved to be effective, more sophisticated color models can be straightforwardly substituted and improvements are expected.

Having defined the main variables in the system, in the next section we proceed to introduce the log-likelihood that will be optimized to compute the output (the state of all the voxels and pixels) from the input (the single image).

3.2. The basic inference algorithm

In this section we present the basic algorithm proposed to estimate the pixel and voxel states that jointly best explain the input image, while respecting the object's prior 3D knowledge. To this end, we first define the likelihood for a given hypothesis (i.e., for a given class and a given pose), and then present an efficient and exact algorithm to find its maximum. In sections 3.4 and 3.5 we will describe how to use this basic algorithm when the hypothesis is not known.

The system's likelihood is derived using the independence assumptions depicted in Figure 2:

$$L(Q,V) = \left(\prod_{j=1}^{N} p(c_j|Q_j).P(Q_j|V_{R(j)}) \right).P(V|K). \quad (1)$$

Three kinds of factors appear in this expression. The first kind of factors has the form $p(c_j|Q_j)$, and as explained in the previous section, is given by the background and foreground color models.

The second kind of factors, $P(Q_j|V_{R(j)})$, are termed *projection factors* (in level 3 of Figure 2). Each factor is the probability of obtaining a particular pixel state (i.e., B or F) given the states of the pixel's voxels. Intuitively, the higher the number of *Full* voxels in a ray, the higher the probability should be of the *Foreground* pixel state. To model this dependency, we adopt a law motivated by the Beer-Lambert law in optics [18],

$$P(Q_j = B|V_{R(j)}) = \prod_{i \in R(j)} \mathrm{E}(V_i)^r, \quad (2)$$

where $\mathrm{E}(V_i)$ is the probability that voxel V_i does not occlude the background. This probability does not depend on the particular voxel in the ray, and is of course greater when the voxel is *Empty* than when it is *Full* (we used in all experiments $\mathrm{E}(E) = 0.995$, and $\mathrm{E}(F) = 0.005$). Note that since the length of intersection of voxels and rays r appears as an exponent in (2), this expression is invariant to the number n of voxels in a ray and their depth. Furthermore, since r is constant, (2) does not depend on *which* voxels are *Full*, just on *how many*. By defining n_F to be the number of *Full* voxels in the ray, the logarithm of (2) can be written as,

$$\log P(B|n_F) = r.n_F.\log \mathrm{E}(F) + r.(n - n_F).\log \mathrm{E}(E). \quad (3)$$

This expression, consequence of having defined the spherical shells to be equispaced, will allow us to considerably simplify the optimization algorithm (at the end of this section).

The third factor, $P(V|K)$, is the *3D shape prior* (in level 1 of Figure 2). It is the probability that a given set of voxel states, V, is a valid reconstruction for an object of class K. This probability is based on previous 3D observations of objects of class K, and it is where the 3D knowledge of the object is encoded. Given the large number of variables in V (in the millions in the experiments presented in Section 4), it is intractable to learn a full joint probability $P(V|K)$.

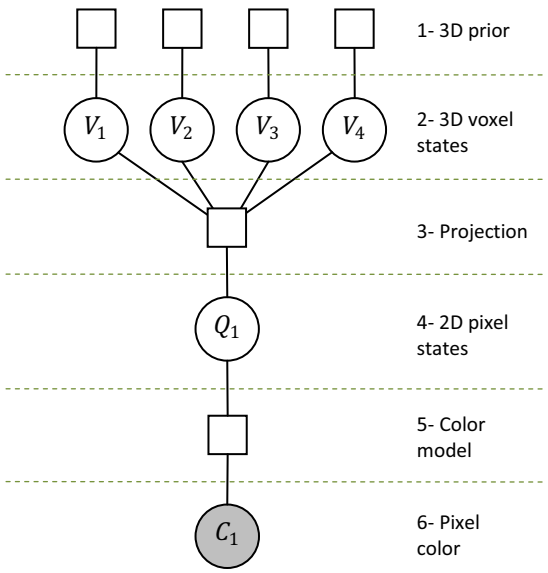

Figure 2: *Factor graph for a ray, its corresponding pixel, and the voxels it intersects. A factor graph,* [17]*, has a variable node (circle) for each variable, and a factor node (square) for each factor in the system's joint probability. Factor nodes are connected to the variable nodes of the variables in the factor. Observed variables are shaded.*

Instead, we make the simplifying assumption of conditional independence (conditioned on the class) between voxels,

$$P(V|K) = \prod_{i=1}^{M} P(V_i|K). \tag{4}$$

In the remaining of this section and the next, we assume that the values of the factors on the right hand side of (4) are given; in Section 3.4 we will show how to learn these factors from real 3D object data.

This assumption of conditional independence, together with the fact that each voxel projects to a single pixel (recall Figure 1), yields a loop-less graph (Figure 2) that allows us to efficiently estimate the states of the voxels and pixels in each ray independently of the states in other rays. This leads to the following expression for the maximum log-likelihood in the *j*-th ray,

$$G_j\left(Q_j, V_{R(j)}\right) = \max_{q_j, v_1, \ldots, v_n} \left\{ \log p\left(c_j|Q_j\right) + \log P\left(Q_j|V_{R(j)}\right) + \sum_{i \in R(j)} \log P(V_i|K) \right\}. \tag{5}$$

Consequently, the global maximum log-likelihood, G, can be written as the sum, over all rays, of the maximum per ray log-likelihood:

$$G = \log L(Q,V) = \sum_{j=1}^{N} G_j\left(Q_j, V_{R(j)}\right). \tag{6}$$

In summary, then, we will proceed as follows to estimate a hypothesis' log-likelihood: first, we compute the ray log-likelihood (and the optimal pixel and voxel states) for each ray using (5); then, we add the contributions of all rays to obtain the global hypothesis' maximum log-likelihood. Before describing the proposed algorithm to solve (5), let us introduce the following definitions (using notation borrowed from the belief propagation literature [17]):

$$\begin{gathered} \mu_{Q_j}(q_j) \triangleq \log\left[\frac{p\left(c_j|q_j\right)}{p\left(c_j|B\right) + p\left(c_j|F\right)}\right], \\ \mu_{V_i}(v_i) \triangleq \log P(v_i|K), \quad \delta_{V_i} \triangleq \mu_{V_i}(F) - \mu_{V_i}(E), \\ S(n_F) \triangleq \sum \left\{ n_F \text{ largest } \delta_{V_i}\text{'s} \right\} \quad 0 \le n_F \le n \end{gathered} \tag{7}$$

With these definitions and using (3), (5) can be rewritten as

$$G_j = \max_{q_j, v_1, \ldots, v_n} \left\{ \sum_{i=1}^{n} \mu_{V_i}(E) + \mu_{Q_j}(q_j) + \log P\left(q_j|n_F\right) + \sum_{i=1}^{n} \delta_{V_i}.v_i \right\}. \tag{8}$$

Since the projection term only depends on the number n_F of *Full* voxels in the ray, this expression is efficiently optimized by the following algorithm (for clarity, the computation of the corresponding voxel states is not shown):

$MaxG = -\infty$.
for each n_F in $\{0, \ldots, n\}$ and q_j in $\{B, F\}$ do:
    $NewG = \mu_{Q_j}(q_j) + \log P\left(q_j|n_F\right) + S(n_F)$.
    If $(NewG > MaxG)$ then $MaxG = NewG$.
end.
$MaxG = MaxG + \sum_{i=1}^{n} \mu_{V_i}(E)$.

The reader can verify that this algorithm computes the maximum ray log-likelihood in $O(n.\log n)$ operations (recall that *n* is the number of voxels in the ray). If there are n^2 rays to be processed, then there are $N = n^3$ voxels, and the overall complexity of the algorithm to compute a hypothesis' log-likelihood is $O(N.\log N)$. This is the basic algorithm we propose to integrate prior 3D knowledge encoded in the 3D shape prior, with the current observations in the input image. In the next section we describe how to compute bounds for the log-likelihood even more efficiently, by performing it at multiple resolutions.

3.3. Multiresolution computation

In the previous section we presented a probabilistic formulation and an optimization algorithm to recover the 3D structure of an object for a particular hypothesis (this was implicit in our assumption that the 3D shape prior $P(V_i|K)$ was given). However, in general the hypothesis (i.e., the class and pose of the object in the image) is not known, and it should be estimated as part of the optimization. To estimate the hypothesis, in Section 3.5 we propose a method that computes the log-likelihood (as explained in the previous section) for a large number of hypotheses. For this reason, it is imperative to be able to discard unpromising hypotheses with the least possible amount of computation. In this section we show how to bound a hypothesis' log-likelihood by computing it at lower resolutions, and hence, with less computation. In Section 3.5 we show how to exploit these bounds to efficiently select the best hypothesis.

We start by introducing additional notation and definitions. Let the *scale* of a voxel or pixel be indicated by a superscript, as in V_i^s or Q_j^s. A scale of *s* indicates that the voxel has been downsampled *s* times by two (see Figure 3). In other words, voxel V_i^s contains 8^s of the original voxels (at scale 0). Let $d^s(i)$ be this set of voxels. Analogously, the pixel Q_j^s contains 4^s of the original pixels and $d^s(j)$ is this set of pixels. Superscripts '+' and '-', e.g., as in μ^+ or μ^-, indicate that a quantity is an upper or lower bound, respectively, or that it is used to compute these bounds.

Using this notation and definitions, we define new quantities related to those defined in (7), however at scale *s*:

$$\begin{gathered} \mu_{Q_j^s}^-\left(q_j^s\right) \triangleq \sum_{k \in d^s(j)} \mu_{Q_k}(q_k)/|d^s(j)|, \\ \mu_{Q_j^s}^+\left(q_j^s\right) \triangleq \max_{k \in d^s(j)} \left\{ \mu_{Q_k}(q_k) \right\}, \end{gathered} \tag{9}$$

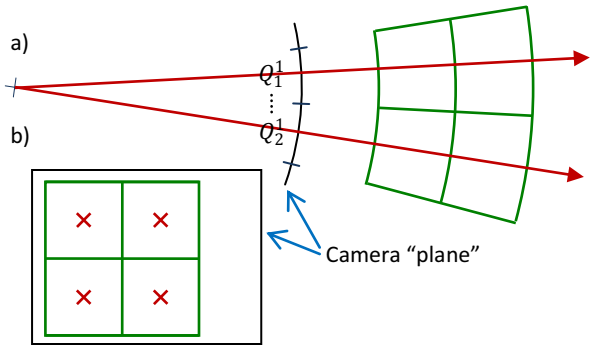

Figure 3: *Same scene as in Figure 1, downsampled by 2 (scale = 1). The number of pixels has been reduced by 4^1 and the number of voxels has been reduced by 8^1 (half of the voxels are not visible in each view).*

$$\mu_{V_i}^{-s}(E) \triangleq 2^s \sum_{k \in d^s(i)} \mu_{V_k}(E) / |d^s(i)|,$$

$$\mu_{V_i}^{+s}(E) \triangleq 2^s \max_{k \in d^s(i)} \{\mu_{V_k}(E)\},$$

$$\delta_{V_i}^{-s} \triangleq \frac{2^s}{|d^s(i)|} \sum_{k \in d^s(i)} \delta_{V_k}, \quad \delta_{V_i}^{+s} \triangleq 2^s \max_{k \in d^s(i)} \{\delta_{V_k}\}.$$

As shown in the supplemental material, substituting the '-' quantities in (9) for the corresponding original quantities in (8), yields a lower bound G^{-s} for the hypothesis' log-likelihood (after scaling by 4^s to account for the reduced number of rays at this scale). Analogously, substituting the '+' quantities in (9) for the corresponding original quantities in (8), yields an upper bound G^{+s} for the hypothesis' log-likelihood (after scaling by 4^s). In summary, the hypothesis' log-likelihood at scale 0 (the original scale) is bounded by the hypothesis' log-likelihoods computed at higher scales (lower resolutions):

$$4^s . \sum_{j=1}^{N/4^s} G_j^{-s} \leq \sum_{j=1}^{N} G_j \leq 4^s . \sum_{j=1}^{N/4^s} G_j^{+s} \qquad (10)$$

Since at scale s the number of pixels and voxels along each dimension is reduced by 2^s (compared to the original resolution), the overall complexity of the algorithm at this scale is reduced by (approximately) 8^s. However, implicit in this calculation is the assumption that the quantities in (9) can be computed from the original quantities in (7) in constant time. We now show how to do this.

Computing the '-' quantities in (9) involves adding a number of terms that increases exponentially with the scale. To compute this in constant time, we rely on *integral images* [19], an image representation precisely proposed to compute sums in rectangular domains (or analogous 3D regions) in constant time. To accomplish this, integral images store in each pixel (or voxel) the cumulative sum of the values in pixels (or voxels) with lower indices.

Similarly, computing the '+' quantities in (9) entails finding the maximum of a number of terms that increases exponentially with the scale. To compute this in constant time, we rely on a *max-pyramid*, a structure having layers with copies of the original (2D or 3D) image at different resolutions (Figure 4). The lowest layer (the base of the pyramid) contains a copy of the original image (the quantities defined in (7)). Each pixel (or voxel) in the layers above, contains the maximum of the 4 corresponding pixels (or 8 corresponding voxels) in the layer below. Maxima are computed using the max pyramid in constant time, simply by looking up the value at the appropriate scale (layer) in the pyramid.

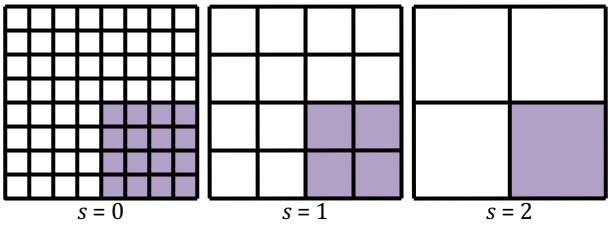

Figure 4: *The max-pyramid, a structure to compute maxima efficiently. The value of the pixel marked at scale 2 (in violet) is the maximum of the four pixels marked at scale 1, which in turn were each computed as the maximum of the four pixels below it at scale 0 (the original scale).*

To use both integral images and max-pyramids, it is necessary to precompute the auxiliary quantities needed by the look up algorithms. In both cases, this is done only once (not for every hypothesis) during an initial stage.

3.4. 3D shape priors

It was assumed in previous sections that the shape prior $\{P(V_i|K)\}$ was given. In this section we describe how to learn this shape prior from data, for a particular hypothesis consisting of a class and an *affine transformation* or pose.

Suppose that we have a sample of (3D) solids belonging to the class of objects K, and that all these solids are registered with respect to a common local coordinate system (LCS). Then, the (empirical) probability $U_K(\vec{x})$ that a 3D point $\vec{x}$ in this LCS is inside an unknown solid of class K is well defined, and given by the number of solids in the sample that contain the point, divided by the total number of solids in the sample [1]. This is the shape prior of the class, for points in the LCS. Using this simple method, shape priors for the classes "mugs," "cups," "bottles," and "plates" (containing 35, 15, 20, and 12 elements respectively), were obtained. Videos of the shape priors obtained for each class are included as supplemental material. These priors, together with the 3D models of all the objects, can be obtained from the authors by request.

To compute the shape prior $P(V_i|K)$ from U_K, let us first define $\overrightarrow{X_i}$ to be the center of voxel V_i, with respect to the world coordinate system (WCS), and let T be an affine transformation that maps points in the LCS into the WCS. This *transformation* was previously referred to as *pose*, and is one of the unknowns that our system must determine. Assuming that the voxel (at scale 0) is sufficiently small, the shape prior can be simply defined as,

$$P(V_i|K) \triangleq U_K\left(T^{-1}\overrightarrow{X_i}\right) \quad (11)$$

For an arbitrary transformation T^{-1} that maps voxels in the WCS into the LCS, there is no guarantee that the faces of the transformed voxel will be parallel to the axes of the LCS (see Figure 5). Therefore, it is not possible to use the techniques described in Section 3.3 (integral images and max-pyramids) to compute the bounds in (9). Instead, we use the following procedure to estimate the upper bound of a quantity μ inside a voxel V_i in the LCS:

- First, we compute the limits of a box B that contains V_i and whose sides are parallel to the axis of the LCS (dotted box in Figure 5).
- Then, we use the length of the longest side of B to select the layer λ to inspect in the max-pyramid, so that no more than a predefined number of voxels (we use 50) has to be inspected.
- Finally, we search the maximum value of μ in all the voxels at layer λ that intersect the box B.

It can be shown that this procedure finds an upper bound for the maximum (but not necessarily *the* maximum) of μ inside V_i, in constant time.

A similar procedure, but relaying on a min-pyramid instead of a max-pyramid, is used to compute the lower bounds in (9). These bounds however, are not as close to those in (9) as the upper bounds are, since in this case the *minimum* is used instead of the *average* (as required in (9)). Note that these looser bounds only apply to the voxel quantities $\mu^{\pm}_{V_i^s}$ and $\delta^{\pm}_{V_i^s}$; the pixel quantities $\mu^{\pm}_{Q_j^s}$ can still be efficiently and tightly bound with the techniques described in the previous section.

Thus far we have defined a formula and algorithm to compute the maximum log-likelihood of a *single* "explanation of the state of the world," or *hypothesis*, for the scene. In the next section we present an algorithm to efficiently sift through a large set of different hypotheses.

3.5. Selection of the best hypothesis

The formulas and algorithms presented in previous sections provide the means to check the suitability of a *single* hypothesis, while permitting to choose the desired accuracy (and hence the amount of computation). In this section we describe an algorithm to *simultaneously* explore *all* the hypotheses, and select the optimal one. This algorithm can be viewed as a focus-of-attention mechanism [19], which refines log-likelihood bounds and discards hypotheses as soon as they are proved non-optimal.

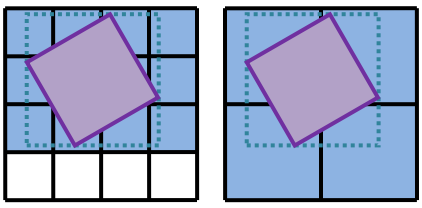

Figure 5: *Computing bounds for a voxel (in violet) using min- (or max-) pyramids at two different scales. Voxels inspected at each scale (in blue) are those that intersect the bounding box (dotted). At the lowest scale (left) the bounds are tighter at the expense of more computation (more voxels have to be inspected).*

The first step of the procedure is to sample the space of possible hypotheses (given each by an affine transformations and a class), and bound the log-likelihood of each hypothesis at the highest scale (lowest resolution) using the algorithm described in previous sections. During the whole procedure, the value of the maximum lower bound among all hypotheses is tracked, and hypotheses are discarded as soon as its upper bound falls below this value (i.e., when a hypothesis is proven non-optimal). Hypotheses are refined (see below) in order of decreasing *margin*, defined as the hypothesis' current upper bound minus the current maximum lower bound. This procedure guarantees that computation, at any time, is only spent in plausible hypotheses.

Recall that according to (6), a hypothesis' log-likelihood is computed as the sum of the log-likelihoods of its rays. To bound a hypothesis' log-likelihood, and to be able to progressively refine these bounds, a *max-heap* [20] containing the rays whose contribution was already added to the hypothesis' log-likelihood is kept (one heap *per* hypothesis is kept). The *key* used to insert elements in this heap is the difference between the ray's upper and lower bounds (the *margin*), so that the ray with the greatest margin can be accessed efficiently.

As mentioned above, hypotheses start with bounds computed at the lowest resolution (i.e., their heap initially contains a single ray intersecting a single voxel at the lowest resolution). Subsequently, when a bound refinement request is received, the following steps are performed:

- The ray R with the greatest margin is removed from the heap.
- R is subdivided into four subrays which cover approximately equal areas in the camera plane.
- The bounds for each subray are computed.
- The hypothesis' bounds are updated by subtracting from it R's bounds and adding the bounds of the subrays.
- The subrays are inserted into the heap.

These steps guide the computation to be spent in the rays that might provide the greatest reduction in the hypothesis' margin, resulting in many rays that are not processed at the lowest scale, since a hypothesis is selected before they are needed.

This concludes the presentation of the proposed framework. In the next section we describe the experiments performed to validate and test the approach.

4. Experimental results

To validate the framework, we first tested the pose estimation error obtained when the prior of an object (i.e., the prior of the class consisting just of itself) was used to estimate its *own* pose. Twenty images of cups and twenty images of mugs, from three different viewpoints (see examples on the left of Figure 7), were hand segmented and

analyzed with the proposed framework. Only images of mugs in which the orientation could be unambiguously estimated (i.e., the silhouette of the handle could be seen) were considered. Hypotheses were given by the Cartesian product {-5cm, -4cm, …, 5cm} × {-5cm, -4cm, …, 5cm} × {0º, 15º, …, 345º}. The result of this product is a 3D grid where each grid point represents a hypothesis. The first two coordinates of each grid point correspond to the position and the third corresponds to the orientation. The results of these experiments are summarized in Table 1.

Table 1		
	Mean Translation Error (cm)	Mean Rotation Error (º)
Cups	0.50	-
Mugs	0.45	0

These errors are within the precision of the camera calibration and the measurement of the ground truth pose parameters, proving the effectiveness of the framework to estimate the pose of *known* objects.

As explained in Section 3.5, most computation is spent deciding between the best hypotheses, in other words, hypotheses that are furthest from the true hypothesis are discarded first and with less computation. This assertion is quantified in Figure 6, which shows the computation time required to decide between two hypotheses.

Next we proceeded to test the more challenging case in which the object in the scene is not known, just its class is. Twenty-four images of cups, mugs, bottles, and plates, from three different viewpoints (see examples on the left of Figure 7) were analyzed. In this case images of the background were used to automatically compute the foreground probability (μ_{Q_j} as defined in (7), and shown in the middle column of Figure 7).

To measure the quality of the reconstructions, we define the *reconstruction error* of a reconstruction S with respect to the ground truth reconstruction S_{GT} as

$$E_R(S) \triangleq 100.\frac{|S_{GT} - S| + |S - S_{GT}|}{2.|S_{GT}|}, \qquad (12)$$

where '$-$' is the usual set difference and '$|\cdot|$' is the volume measure.

The objects in each class were clustered using *k*-means and the error defined in (12) (see details in the supplemental material). For each image analyzed, hypotheses were defined using the priors constructed for each cluster, excluding the object in the image (i.e., an object was *not* used in the prior for its own reconstruction).

The affine transformations allowed in this case combined: 1) horizontal translations $\vec{t}$ (in the X-Y axes); 2) rotations of φ degrees around the vertical (Z) axis; 3) scaling of S_Z percent in the Z direction; and 4) scaling of S_{XY} percent in the X and Y directions (the same constant is used in both directions). Recovering the pose in this case meant recovering all these parameters. Accordingly, the hypotheses in this case were given by the Cartesian product {-5cm, -4cm, …, 5cm} × {-5cm, -4cm, …, 5cm} × {0º, 15º, …, 345º} × {80%, 90%, …, 120%} × {80%, 90%, …, 120%} (the coordinates of each grid point are, in order: $\vec{t}$, φ, S_Z, and S_{XY}).

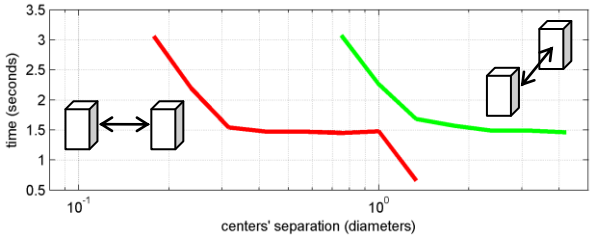

Figure 6: *Computation time required to select the correct hypothesis between two hypotheses separated a given distance, in a direction approximately parallel (red) and approximately perpendicular (green) to the camera plane. See text for details.*

Table 2 summarizes the results of these experiments. The columns in the table correspond, from left to right, to the mean reconstruction error (E_R), and the mean error in the transformation parameters ($E_{\vec{t}}$, E_φ, E_{S_Z}, $E_{S_{XY}}$). Note that even in the case of *unknown* objects, the pose estimation errors are within the precision of the camera calibration and the measurement of the ground truth parameters.

Snapshots of the reconstructions obtained are shown in the rightmost column of Figure 7, and videos of these reconstructions are included as supplemental material. Note that very good reconstructions were obtained, even without explicitly modeling the obvious object symmetries. The segmentations corresponding to the experiments in Figure 7 are also included as supplemental material.

Table 2					
	E_R (%)	$E_{\vec{t}}$ (cm)	E_φ (º)	E_{S_Z} (%)	$E_{S_{XY}}$ (%)
Cups	9.9	0.5	-	5.0	7.5
Mugs	4.9	-	5.0	6.7	6.7
Bottles	7.7	0.9	-	5.7	11
Plates	14	1.1	-	16	4.3

The framework was also tested on a simple instance of the object recognition problem. Objects of the classes "mugs," "cups," "bottles," and "plates," were automatically classified into the class (one of these four) that produced the highest likelihood. On the 23 images tested, 100% correct recognition was obtained. Details of this test are included in the supplemental material.

5. Conclusions and future work

In this work we used a simple statistical representation encoding the 3D shape of a class of objects, and presented a very efficient framework that gives very good results on several important problems in computer vision, including pose estimation, 2D segmentation, object recognition, and 3D reconstruction from a single image. Besides its generality, this framework has the following additional desirable properties: 1) it considers prior knowledge about the objects in the scene; 2) it is guaranteed to find the globally optimal solution; 3) it is very efficient and its computational com-

plexity only depends on the question to be answered (e.g., on the similarity between the objects or poses to be discriminated) and not on the arbitrary resolution of the input image; and 4) it encodes, in a principled way, the uncertainty resulting from the finite resolution of the input image (via the log-likelihood bounds).

While very accurate results were obtained for relatively simple object classes with these simple shape priors, more complex classes will require more complex priors. One possibility to construct these more complex priors, while retaining the optimality properties of the current representation, is to consider objects as the union of other objects (i.e. *parts*), which themselves can be represented as the union of other objects or using the shape priors introduced in this article.

Additional modifications to this framework that are expected to produce significant speedups, include: 1) developing new techniques to compute tighter bounds for the likelihood (sections 3.3 and 3.4); 2) designing new algorithms to allocate computation during the refinement of the hypotheses' bounds (Section 3.5); and 3) implementing the framework on a GPU, an architecture for which it is ideal.

Acknowledgements

This work was carried out while DR was at the University of Minnesota. We thank Lorena Rother and Adriana Telias for their help during the acquisition of the shape priors. Support for this work came from ARO, NSF, NGA, ONR, and DARPA.

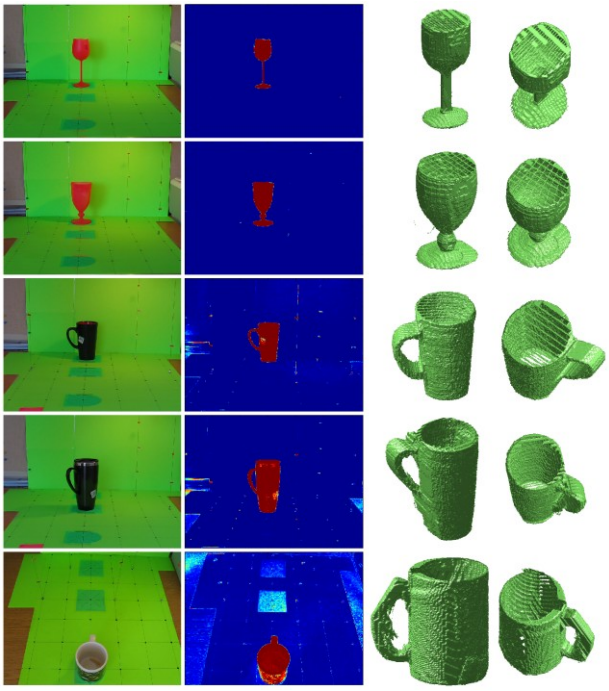

Figure 7: *Reconstructions obtained for objects of the classes "cups" and "mugs". Left) Original images. Middle) Foreground probabilities. Right) Two views, different from the views on the left column, of the reconstructions obtained (artifact holes are a product of the conversion to the surface to display, not of the framework presented in this work). Videos of the reconstructions are included as supplemental material.*

References

[1] Rother, D., Patwardhan, K., Aganj, I., and Sapiro, G., "3D Priors for Scene Learning from a Single View." *S3D Workshop (at CVPR),* 2008.

[2] Bowden, R., Mitchell, T. A. and Sarhadi, M., "Reconstructing 3D Pose and Motion from a Single Camera View." *Proc. British Machine Vision Conf.,* 1998.

[3] Agarwal, A. and Triggs, B., "Learning to Track 3D Human Motion from Silhouettes." *Proc. 21st Int'l Conf. on Machine Learning,* 2004.

[4] Isard, M. and MacCormick, J., "BraMBLe: A Bayesian Multiple-Blob Tracker." *ICCV,* 2001.

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

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

[7] Ramanan, D., "Learning to Parse Images of Articulated Objects." *Advances in Neural Information Processing Systems,* 2006.

[8] Mikic, I., Trivedi, M., Hunter, E., and Cosman, P., "Human Body Model Acquisition and Tracking Using Voxel Data." *IJCV*, 2003.

[9] Balan, A. O., Sigal, L., Black, M. J., Davis, J. E., and Haussecker, H. W., "Detailed Human Shape and Pose from Images." *CVPR,* 2007.

[10] Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., and Davis, J., "SCAPE: Shape completion and animation of people." *SIGGRAPH,* 2005.

[11] Han, F. and Zhu, S. C., "Bayesian Reconstruction of 3D Shapes and Scenes From A Single Image." *Proc. IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis,* 2003.

[12] Saxena, A., Sun, M. and Ng, A. Y., "Make3D: Learning 3D Scene Structure from a Single Still Image." *PAMI,* 2008.

[13] Hoiem, D., Efros, A.A. and Hebert, M., "Automatic Photo Pop-up." *SIGGRAPH,* 2005.

[14] Dyer, C. R., "Volumetric scene reconstruction from multiple views." L. S. Davis. *Foundations of Image Understanding.* Kluwer, Boston, 2001.

[15] Snow, D., Viola, P. and Zabih, R., "Exact Voxel Occupancy with Graph Cuts." *CVPR*, 2000.

[16] Franco, J. S. and Boyer, E., "Fusion of Multi-View Silhouette Cues Using a Space Occupancy Grid." *ICCV,* 2005.

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

[18] Lakowicz, J. R., *Principles of Fluorescence Spectroscopy.* Springer, New York, 2006.

[19] Viola, P. and Jones, M., "Rapid Object Detection Using a Boosted Cascade of Simple Features." *CVPR,* 2001.

[20] Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein, C., *Introduction to Algorithms.* MIT Press, Cambridge, MA, 2001.