Efficient Object Localization and Pose Estimation with 3D Wireframe Models

Erdem Yörük René Vidal
Center for Imaging Science, Johns Hopkins University, Baltimore, MD 21218, USA

Abstract

We propose a new and efficient method for 3D object localization and fine-grained 3D pose estimation from a single 2D image. Our approach follows the classical paradigm of matching a 3D model to the 2D observations. Our first contribution is a 3D object model composed of a set of 3D edge primitives learned from 2D object blueprints, which can be viewed as a 3D generalization of HOG features. This model is used to define a matching cost obtained by applying a rigid-body transformation to the 3D object model, projecting it onto the image plane, and matching the projected model to HOG features extracted from the input image. Our second contribution is a very efficient branch-and-bound algorithm for finding the 3D pose that maximizes the matching score. For this, 3D integral images of quantized HOGs are employed to evaluate in constant time the maximum attainable matching scores of individual model primitives. We applied our method to three different datasets of cars and achieved promising results with testing times as low as less than half a second.

1. Introduction

View-invariant object detection and fine resolution 3D pose estimation are fundamental problems in computer vision. However, the fact that objects often appear in cluttered scenes, occlude each other, and exhibit immense variability in their appearance, shape and pose, makes both problems incredibly challenging. Since the early days, the computer vision community advocated that a 3D representation of objects was key to properly addressing these challenges [19, 4, 1, 15, 3]. The main reasons are that 3D object models are independent of the viewpoint and bear global 3D shape information that can disambiguate local 2D detections.

Depending on the level of detail in both the 3D object representation and the accommodated viewpoints, recent approaches can be roughly divided into three main families. Methods in the first family combine multiple single-view 2D detectors into 3D [22, 25], whereas approaches in the second family extend these ideas to viewpoint annotated data to better handle large pose variations [20, 21, 24]. In general, these methods follow the intuition that 2D features and regions can be effectively used to describe certain views, whose combination can lead to a global decision about the object pose and class label. On the other hand, methods in the third family try to build explicit 3D representations of objects, for a detailed 3D inference from 2D input [26, 14, 2, 16, 11, 10]. In such approaches, object models have more expressive power, but this comes at the expense of additional challenges. First, acquiring a generic 3D model for an object category remains an active research problem on its own. Second, entertaining such models with the huge pose space remains a computational bottleneck, especially, when inference is done from a single 2D image.

Paper contributions. In this paper, we address these challenges by making the following two important contributions: (i) For representing objects, we propose to learn explicit 3D wireframe models from object blueprints, which are vis-a-vis complementary and orthographic sketches of the object in a few canonical views, and (ii) for estimating a fine grained 3D pose, we layout a very efficient optimization scheme using a Branch and Bound (BB) algorithm.

The precision in pose estimation is intimately coupled with the precision in the 3D object representation. On the one hand, such a representation can be readily obtained via off-the-shelf CAD models (*e.g.*, [18]). This would save efforts on training, but would require access to dedicated CAD scans for every other category of interest. On the other hand, object models can be learned generically from actual images (*e.g.*, [20]), which is easily generalizable to different classes. However, the quality of the learned models is prone to accurate manual annotation, background subtraction, correspondence matching and 3D reconstruction. Alternatively, methods can benefit from both ends, where CADs are used for guiding the learning process (*e.g.*, [14]). In that spectrum, we opt for a promising alternative, which, to the best of our knowledge, has not been tested for the subject matter. In particular, we propose an efficient model construction scheme from object blueprints, which, in the absence of a CAD model, can be produced more easily than a whole 3D scan, can be found fairly abundantly on the web for most of the commercial object categories, can be seamlessly triangulated to a volumetric shape without the need of

correspondence matching, and can provide models as precise and detailed as CADs. Reconstructed and averaged from several blueprint exemplars of the target category, our object representation is eventually a set of 3D edge primitives at significant boundaries of the object's texture and geometry, combined also with local surface normals, so as to handle pose dependent self occlusions.

Given this 3D wireframe model, and an input image, we pose the 3D object localization and pose estimation problem as one of matching the edges of the projected 3D model under varying poses, to those of the image summarized by HOG features. Arguably a more difficult challenge remains in the computational aspect of the problem, especially due to the sheer volume of pose hypotheses to be searched, and due to local extrema. To address this challenge, we propose a very fast global optimization using a BB algorithm, which rapidly converges to the pose that maximizes the matching score between the projected 3D model and the input image.

Related work. 3D modeling for view-invariant inference from 2D images has a long tradition in computer vision, which dates back to simple representations like polyhedral shapes [19] and generalized cylinders [4]. Nonetheless, the resurgence of more advanced 3D methods that are especially targeted at joint category and viewpoint classification and pose estimation is fairly recent.

In that context, related work can be grouped into different themes depending on the kind of object representation and the method of inference. In the first theme, models are trained extensively from viewpoint dependent part appearances. For instance, the line of work in [20, 21, 23] decomposes objects into planar part appearances that are related by homographies or affine transformations. Alternatively, [14] proposes to use unsupervised training to acquire separate models for both the 2D appearance and 3D geometry, and combine them at a later stage for probabilistic pose estimation. Another group of work employs a voting-based approach for inference. For example, [2] presents an implicit shape model based on a constellation of features, their locations, and their appearance in multiple views, and uses this model in conjunction with a probabilistic voting procedure to estimate the object pose. Similarly, [8] proposes a nonparametric model and a voting architecture in the pose space for viewpoint estimation. There are also approaches that, rather than dealing with the complex appearance, try to match image discontinuities to geometric cues and salient boundaries of objects under varying pose. In this theme, the line of work in [11, 10] uses 2D and 3D primitives that are stick-like elements, whose projections are represented with Gabor-like filters. With a similar objective, [17] revisits a viewer-centered framework to learn 2D view-dependent shape templates from salient contours in training images, which are grouped using bag-of-boundaries features. Alternatively, the pose estimation problem has also been treated from a purely discriminative perspective. For example, [9] proposes to use a mixture of holistic templates that are discriminatively trained and associated to canonical viewpoints. Again, to be used for the discriminative learning of viewpoints, [18] formulates a 3D deformable part model as an extension of [6] and uses CAD data for training.

Our work departs from existing literature in two significant aspects: First, we use blueprints to acquire explicit wireframe models of objects, hence we do not use dataset-specific training. Second, we propose a very efficient BB algorithm for fine grained pose estimation and localization over the huge space of hypotheses. Although, bounding techniques have already been studied for 2D object detection [13, 12], with existing 3D extensions that use additional depth data [7], to the best of our knowledge, our work is the first to layout a BB algorithm for inferring 3D object poses from a single 2D image.

Paper organization. The paper is organized as follows: In Section 2, we explain our 3D object representation that is obtained from blueprint exemplars. Then, in Section 3, we mathematically state the localization (detection) and fine-grained pose estimation problem, followed by Section 4, where we layout our BB algorithm for inference. Finally, in Section 5, we discuss our experimental evaluation and give our conclusions in Section 6.

2. 3D Object Model

Our approach for view invariant object detection and 3D continuous pose estimation relies on an explicit 3D object class model and the agreement of its pose-varying camera projection with the 2D input image. In this section, we layout our 3D model and its construction from blueprints.

Our object representation is composed of oriented primitives that encode the 3D shape and a coarse appearance information in terms of edges at the boundaries of object faces and at the discontinuities of object texture. In particular, our 3D model is defined as a set of M object primitives

$$\mathcal{M} = \{(\mathbf{p}^m, \mathbf{e}^m, \mathbf{n}_1^m, \mathbf{n}_2^m)\}_{m=1}^{M}, \tag{1}$$

where each primitive is specified by four vectors in $\mathbb{R}^3$: (i) a 3D location $\mathbf{p}$, (ii) a 3D edge direction $\mathbf{e}$, and (iii) two normal vectors $\mathbf{n}_1$ and $\mathbf{n}_2$, all given with respect to some canonical object coordinate system (OCS).

The unit edge and normal vectors describe the local geometric and crude appearance properties of the object. That is, given primitive $(\mathbf{p}, \mathbf{e}, \mathbf{n}_1, \mathbf{n}_2)$, the edge vector $\mathbf{e}$ is along the local 3D discontinuity of color or surface reflectance. On the other hand, unit normal vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ encode the two planar surfaces intersecting at an edge passing through $\mathbf{p}$. As will be discussed later, edge vectors will be projected and matched to intensity discontinuities on the 2D image for validating a pose, whereas surface normals

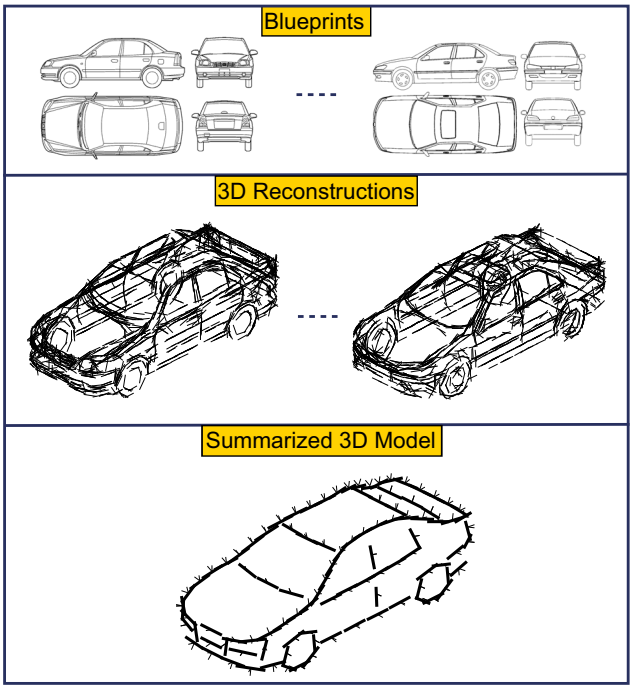

Figure 1: Model construction from blueprints: 2D blueprints (top) are registered to 3D raw models (middle), and then summarized to final model (bottom) with cluster centers found by k-means. Solid black line segments in the final model are salient 3D edges $\mathbf{e}^m$ at $\mathbf{p}^m$, whereas thin black vectors indicate the two normals $\mathbf{n}_1^m$ and $\mathbf{n}_2^m$ of the intersecting shape faces, which are found after a second round of spherical k-means.

will determine the primitive visibility and self occlusions, based on their orientation relative to the camera. By using two normals we will ensure the visibility of shape edges at intersecting object faces from very different views.

2.1. Model Learning from Blueprints

As illustrated in Figure 1-top, a blueprint is a set of edge sketches of the object, which are captured from a few complementary and canonical views (typically frontal, back, side etc.), each obtained by orthographic projection. These are binary images with intensity values set to one, wherever there is a texture or shape edge, and zero elsewhere. In particular, let $\mathcal{I}^{\text{front}}$, $\mathcal{I}^{\text{back}}$, $\mathcal{I}^{\text{left}}$, $\mathcal{I}^{\text{right}}$, $\mathcal{I}^{\text{top}}$ and $\mathcal{I}^{\text{bottom}}$ be the respective 2D sketches from frontal, back, left, right, top and bottom views, which are aligned with each other in terms of scale and projected axes of the OCS. Thanks to shared axes and scales in blueprints, one does not need a correspondence matching for reconstruction: An orthographic ray emanating from one view is identified as a unique row or column in other views, providing the missing depth information as the location of object boundary in that particular row or column. Moreover, since the "on" pixels from each blueprint sketch already correspond to discontinuities in object's shape or texture, their triangulation directly provides a 3D sketch such as the one shown in Figure 1-middle, from which the desired primitives for our 3D object model (1) can easily be extracted.

In order to explicitly describe the 3D reconstruction from blueprint images, we first fix the OCS such that the corresponding x-axis spans from the object's back to front, the y-axis from its right to left, and the z-axis from its bottom to top. Accordingly, we reparametrize the 2D pixel coordinates of edges in each individual view with this common OCS. Below, we give the details of reconstruction for the edge pixels found in the frontal sketch $\mathcal{I}^{\text{front}}$. Reconstructions from other views follows analogously, where source view pixel coordinates and signs of the OCS axes are adjusted accordingly.

Reconstructing 3D locations. Let $\mathbf{p} = (p_x, p_y, p_z)^\top$ be a 3D point on the object, that is visible as an edge in the frontal view $\mathcal{I}^{\text{front}}$ with the corresponding horizontal-vertical pixel coordinate (p_y, p_z). Then, using the left and top views $\mathcal{I}^{\text{left}}$ and $\mathcal{I}^{\text{top}}$, where pixel locations are respectively read off in terms of their xz- and xy- coordinates in the OCS, the component p_x of $\mathbf{p}$, which is missing from $\mathcal{I}^{\text{front}}$, is simply obtained by

$$p_x = \min\big\{\max\{q_x : \mathcal{I}^{\text{left}}(q_x, p_z) = 1\}, \max\{q_x : \mathcal{I}^{\text{top}}(q_x, p_y) = 1\}\big\} \quad (2)$$

Here, the inner $\max$ terms pick the "front-most" edge pixels from $\mathcal{I}^{\text{left}}$ and $\mathcal{I}^{\text{top}}$, within their p_z^{th} and p_y^{th} rows, respectively. Note that, $\mathcal{I}^{\text{right}}$ and $\mathcal{I}^{\text{bottom}}$ could be equivalently used, for which the $\max$-operations would return the same values. The outer $\min$-operation is to account for cavities in the object shape, which may be available in only one of the argument views.

Reconstructing 3D orientations. Let $\mathbf{e}$ and $\mathbf{n}$ be the respective 3D edge and surface normal on the object shape at point $\mathbf{p}$, which is reconstructed via (2) by back-projecting from $\mathcal{I}^{\text{front}}$ to $\mathcal{I}^{\text{left}}$ and $\mathcal{I}^{\text{top}}$. Since the missing component p_x is found at the 2D object boundary in either $\mathcal{I}^{\text{left}}$ or $\mathcal{I}^{\text{top}}$, the surface normal $\mathbf{n}$ will be along the local image gradient in the particular argument view used for $\mathbf{p}$'s reconstruction. That is, we set

$$\mathbf{n} = \begin{cases} \dfrac{\begin{pmatrix}1&0\\0&0\\0&1\end{pmatrix}\nabla\mathcal{I}_\sigma^{\text{left}}(p_x,p_z)}{\|\nabla\mathcal{I}_\sigma^{\text{left}}(p_x,p_z)\|}, & \text{if } \begin{matrix}\max\{q_x:\mathcal{I}^{\text{left}}(q_x,p_z)=1\}\\ <\max\{q_x:\mathcal{I}^{\text{top}}(q_x,p_y)=1\}\end{matrix} \\ \dfrac{\begin{pmatrix}1&0\\0&1\\0&0\end{pmatrix}\nabla\mathcal{I}_\sigma^{\text{top}}(p_x,p_y)}{\|\nabla\mathcal{I}_\sigma^{\text{top}}(p_x,p_y)\|}, & \text{otherwise} \end{cases} \quad (3)$$

where $\mathcal{I}_\sigma^{\text{left}}$ and $\mathcal{I}_\sigma^{\text{top}}$ are the smoothed versions of $\mathcal{I}^{\text{left}}$ and $\mathcal{I}^{\text{top}}$ by a Gaussian kernel with noise scale σ. The 3×2 matrices that pre-multiply the normalized gradients are used to

set the missing component to zero. Recall that our object model defined in (1), requires two normal vectors that characterize the 3D geometry at intersecting object faces. These are indeed locations, where the reconstruction in (3) locally fluctuates between the two cases involved, yielding locally varying normal vectors. Using an additional cluster analysis step, we will group those later into two distinct directions to be consistent with our object model in (1).

Once the normal $\mathbf{n}$ is computed, the 3D edge $\mathbf{e}$ at $\mathbf{p}$ is found directly by using the facts that (i) it should be orthogonal to $\mathbf{n}$, and (ii) its projection on the source view $\mathcal{I}^{\text{front}}$ should be consistent with the 2D edge observed there, *i.e.* it should be orthogonal to the local image gradient in $\mathcal{I}^{\text{front}}$. Thus, we obtain the 3D edge as the cross-product

$$\mathbf{e} = \mathbf{n} \times \left[\begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \frac{\nabla \mathcal{I}^{\text{front}}_{\sigma}(p_y, p_z)}{\|\nabla \mathcal{I}^{\text{front}}_{\sigma}(p_y, p_z)\|} \right] \tag{4}$$

where again $\mathcal{I}^{\text{front}}_{\sigma}$ is the smoothed version of $\mathcal{I}^{\text{front}}$. In our experiments, we take $\sigma = 1$ pixel.

Building a model from multiple exemplars. We perform the above reconstruction steps for each edge pixel from each of the individual views, and repeat this for several different blueprint examples of the category of interest. This process accumulates a large crowd of points and edges in $\mathbb{R}^3$. To summarize them and obtain an average object model, we apply k-means clustering and use the cluster centers as our final 3D representation. For this clustering step, we use the similarity measure $|\mathbf{e}_1^\top \mathbf{e}_2| \exp\{-\frac{\|\mathbf{p}_1 - \mathbf{p}_2\|_2^2}{2\lambda^2}\}$ between any two point-vector pairs $(\mathbf{p}_1, \mathbf{e}_1)$ and $(\mathbf{p}_2, \mathbf{e}_2)$, where λ is a scale parameter for real-world distances between reconstructed points. In our experiments, we take $\lambda = 10$cm.

The centroids are then computed as follows: For points $\mathbf{p}_i$ assigned to a cluster, we take their Euclidean average, whereas for vectors $\mathbf{e}_i$, we first sum their outer-products $\mathbf{E} = \sum_i \mathbf{e}_i \mathbf{e}_i^\top$, and then compute the eigenvector of $\mathbf{E}$ associated to its largest eigenvalue. This will give the average orientation of the vectors being grouped, but without letting those with opposite signs cancel each other. At convergence of this k-means step, we discard clusters, whose cardinality is less than 75% of the average cluster cardinality, in order to remove outliers from the final representation. As a result, the set $\{\mathbf{p}^m, \mathbf{e}^m\}_{m=1}^M$ of surviving M centroids will constitute the location and edge orientation attributes of our 3D object model (1). Finally, within each cluster $m = 1, \ldots, M$, the normal vectors associated to the constituent points are further grouped under two distinct directions, which are found using another round of spherical k-means. This time, absolute dot products are used to compute vector similarity, and averaging is done the same way it is done for edge vectors, *i.e.*, by using the top eigenvectors of summed outer-products. This procedure will provide the two normals $(\mathbf{n}_1^m, \mathbf{n}_2^m)$ assigned to each cluster m, completing our object model. Note that, when the cluster center $\mathbf{p}^m$ is roughly located at the intersection of two different faces of the mean shape, then the corresponding cluster normals $\mathbf{n}_1^m$ and $\mathbf{n}_2^m$ will be necessarily different, due to variance of constituent normal directions within the cluster. As mentioned earlier, this will ensure the primitive visibility under widely differing views, which we will make explicit later.

Figure 1-bottom shows one blueprint model for the "car" category extracted from the blueprints in Figure 1-top. Note that, the final representation is quite regular, since blueprints are inherently precise and consistent with each other.

3. 3D Object Localization and Pose Estimation

In this section, we formulate the problem of view-invariant object localization and 3D pose estimation from a single 2D image $\mathcal{I}$. Given a wireframe model $\mathcal{M}$ composed of 3D edge primitives, this problem can be expressed as finding the transformation θ that maximizes the match between the perspective projection of $\theta \circ \mathcal{M}$ and the 2D edge pattern observed on $\mathcal{I}$. In our case, we quantify the matching score by feature strengths of $\mathcal{I}$ at locations and orientations where $\theta \circ \mathcal{M}$ gets projected (as in the ordinary 2D template matching, which uses dot products). This requires features that are coarse enough to be robust against shape variations in the target object category, but also local enough to reveal a fine resolution pose. In what follows, we describe the features we use, our pose parameterization, and the proposed matching score.

3.1. Image Features

In this paper, we use a variant of HOG features [5], which are obtained with four main modifications. First, we define the orientation bins using unsigned edge orientations rather than gradient orientations, since our 3D model also has 3D edges with unsigned orientations. Second, we accumulate edges with weights set to their gradient magnitudes as usual, but do not locally normalize the extracted histogram as it is originally suggested. Instead, we pass all entries through a sigmoid function to achieve a dynamic range of $[0, 1]$, which also provides a compression effect like gamma-correction. Third, within each spatial bin, we center the values at the corresponding discrete orientations using their local average, and set the ones that become negative to zero. In this way (i) entries of locally dominant and globally strong edge orientations are favored, emphasizing the foreground object, and (ii) flat regions and cluttered regions with dispersed edges get suppressed. Finally, we quantize the resulting histogram to L uniformly spaced levels in $[0, 1]$, which will be exploited during inference for fast bound computation thanks to integral images. To draw the distinction, we henceforth call our features as quantized-HOG or qHOG in short.

3.2. Pose Parametrization

We consider pose θ as a transformation from the OCS to a camera centered world coordinate system (WCS), with 6 degrees of freedom. We assume a spherical imaging surface, and let $\theta \triangleq (a, b, c, d, e, f)$, where rotation parameters a, b and c define, resepctively, the azimuth, elevation and tilt angles of the camera relative to object; d specifies the depth measured from the camera center to the object center; and (e, f) stand for the 2D translation of object's projection on the input image $\mathcal{I}$ relative to image center.

3.3. Objective Function

Let $\mathcal{H}(i, j, k)$ denote the three dimensional qHOG histogram of input image $\mathcal{I}$, where $(i, j) \in \{1, \dots, I\} \times \{1, \dots, J\}$ indexes the spatial grid of location bins, and $k \in \{1, \dots, K\}$ gives the unsigned orientation bin. Then, given object model $\mathcal{M}$ in (1), we formulate the pose estimation problem from $\mathcal{I}$, as finding the maximizer of

$$\mathcal{Q}(\theta) = \sum_{m=1}^{M} \mathcal{V}(\mathbf{p}^m, \mathbf{n}_1^m, \mathbf{n}_2^m|\theta) \times \mathcal{H}\big(i(\mathbf{p}^m|\theta), j(\mathbf{p}^m|\theta), k(\mathbf{p}^m, \mathbf{e}^m|\theta)\big), \quad (5)$$

where, under pose θ, $\mathcal{V}(\mathbf{p}^m, \mathbf{n}_1^m, \mathbf{n}_2^m|\theta) \in \{0, 1\}$ specifies the binary visibility of the m^{th} primitive, as a function of its location and surface normals in the OCS; and indices $\big(i(\mathbf{p}^m|\theta), j(\mathbf{p}^m|\theta), k(\mathbf{p}^m, \mathbf{e}^m|\theta)\big)$ give its camera projected qHOG bin, again written in terms of its 3D location and edge orientation in the OCS.

The objective function $\mathcal{Q}$ counts the scores over visible primitives only, where an individual score is simply taken as the qHOG response at the projected location and orientation of the corresponding primitive. Thus, $\mathcal{Q}$ can be interpreted as the dot product between two qHOG arrays, namely $\mathcal{H}$ itself, and the one that would be obtained from the 2D edge image of the model $\mathcal{M}$ captured under the pose θ.

4. Inference by a Branch & Bound Algorithm

In this section we describe our approach for maximizing the objective function $\mathcal{Q}$ in (5). Doing this in a greedy or exhaustive fashion is intractable due the prohibitively large space of poses and the issue of local maxima. Bounding schemes, on the other hand, can provide a feasible and global approach with guarantees for optimality, provided one can efficiently compute tight bounds on $\mathcal{Q}$.

At a high level, BB algorithms recursively partition the search space and compute bounds on the objective over each partition element. Sorting these subsets in a priority queue by their evaluated bounds, the search is directed to more viable candidates, which are subdivided further until the finest resolution is reached at convergence. We employ the same bounding idea in our 3D pose estimation problem, where we tackle a very large search space with 6 degrees of freedom. Specifically, for any range Θ of poses, we need to compute a sufficiently tight upper bound on the objective $\mathcal{Q}$. Given its form in (5), this can be achieved for each model primitive separately, by upper-bounding both the corresponding visibility and qHOG components, and then summing their products over all primitives. This requires one to know possible visibility states of each primitive as well as their possible locations and orientations in the feature domain, as a function of Θ. The complex dependence of those quantities on the pose (due to the induced transformation on points and edges, combined with the camera projection) necessitates a multi-level analysis. In particular, for achieving an upper bound on $\mathcal{Q}$, we need to evaluate sequentially (i) bounds on pose-transformed 3D primitive locations and orientations in the scene, (ii) bounds on their camera-projected 2D locations and orientations in the feature domain, which can be deduced by projecting bounds found in (i), (iii) upper bounds on primitive visibility, and (iv) upper bounds on qHOG scores within the subsets of the feature domain as determined in (ii). It is important to note that the only part that uses information from the test image is the last step evaluated on qHOG features. Thus, for a given 3D object model $\mathcal{M}$ and a-priori fixed hierarchy of nested pose partitions, steps (i), (ii) and (iii) can be precomputed and stored before doing inference (see Figure 2 for an illustration). Furthermore, in that offline stage, the corresponding bounds can be made arbitrarily tight at the expense of additional computational cost, and also, they can be evaluated with an arbitrary choice of pose parametrization.

4.1. Bounding Mechanism

The core part of the BB algorithm is the bound computation. As mentioned above, for a given range Θ of poses, this process involves analysis at four levels, ranging from 3D point locations to their projections, visibilities and qHOG scores. In particular, our aim is to find sufficiently tight upper bounds on the objective function $\mathcal{Q}$ in (5), over a given 6-dimensional subspace of poses $\Theta = A \times B \times C \times D \times E \times F$, where $A = [\underline{a}, \overline{a}]$, $B = [\underline{b}, \overline{b}]$, $C = [\underline{c}, \overline{c}]$, $D = [\underline{d}, \overline{d}]$, $E = [\underline{e}, \overline{e}]$ and $F = [\underline{f}, \overline{f}]$ are intervals over each of the pose components. In what follows, we will discuss the details of our bounding scheme for a single primitive $(\mathbf{p}, \mathbf{e}, \mathbf{n}_1, \mathbf{n}_2)$ that gets transformed by poses in Θ and gets projected to camera.

(i) Bounds on 3D locations and orientations. We treat this step as finding 3D bounding boxes around the sets $\mathcal{S}_\mathbf{p} = \{\theta \circ \mathbf{p} : \theta \in \Theta\}$ and $\mathcal{S}_\mathbf{e} = \{\theta \circ \mathbf{e} : \theta \in \Theta\}$. Since the 2D translation (e, f) on the image has no effect here, we consider rotation and depth parameters $(a, b, c, d) \in A \times B \times C \times D$, only. Accordingly, we have $\theta \circ \mathbf{p} = \mathbf{R}(a, b, c)\mathbf{p} + \mathbf{t}(d)$ and $\theta \circ \mathbf{e} = \mathbf{R}(a, b, c)\mathbf{e}$, where $\mathbf{R}$ and $\mathbf{t}$ denote 3D rotation and translation, respectively.

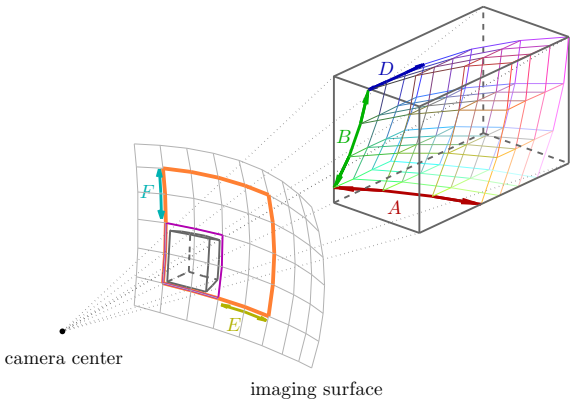

Figure 2: Bounding mechanism for pose transformed primitive locations and their projections. The 3D grid in the scene describes the subvolume of $\mathbb{R}^3$ covered by a point $\mathbf{p}$, when it gets transformed by azimuth angles, elevation angles and depths, from respective intervals A, B and D (camera tilt is omitted for illustration purposes). The surrounding cuboid gives the corresponding 3D bounds, which after camera projection, yields the purple bounding rectangle of HOG-location bins. The translation intervals E and F on the image give the final orange box bounding $\mathbf{p}$'s locations in the feature domain. Bounds on the pose-transformed and projected orientations of an edge $\mathbf{e}$ at $\mathbf{p}$ are found by repeating this process for $\mathbf{p}$ and $\mathbf{p}+\mathbf{e}$ and checking the extreme phases of 2D vectors between the resulting 2D location bounds.

Since $\mathcal{S}_\mathbf{e}$ defines a spherical patch with radius $\|\mathbf{e}\|$, its bounds can be easily found either at the extreme rotations induced by $(\underline{a}, \overline{a}, \underline{b}, \overline{b}, \underline{c}, \overline{c})$, or when it intersects the planes of the WCS. In that latter case, the bounds can be trivially obtained from the extrema of the arc of intersection. For locations, 3D bounds on the corresponding set $\mathcal{S}_\mathbf{p}$ are found in the same way, except now by also extending their z-dimension with extreme depths $\underline{d}$ and $\overline{d}$ (see Figure 2).

(ii) Bounds on projected locations and orientations. In this step, we bound the spatial and angular sub-array of the discrete qHOG domain that is swept by the camera projections of point-edge pairs $\{\theta \circ (\mathbf{p}, \mathbf{e}) : \theta \in \Theta\}$. For that, we project the bounds of the previous step found for 3D locations and orientations. To be specific, let $\{\widehat{\mathbf{p}}_s\}_{s=1}^8$ be the set of eight corners bounding $\mathcal{S}_\mathbf{p}$ and let $\{(\widehat{i}_s, \widehat{j}_s)\}_{s=1}^8$ be their camera projections. Then, the 2D bounding box around the latter, further extended by the extreme 2D translations $(\underline{e}, \underline{f})$ and $(\overline{e}, \overline{f})$ on the image, gives bounds $\big(\underline{i}(\mathbf{p}|\Theta), \underline{j}(\mathbf{p}|\Theta)\big)$ and $\big(\overline{i}(\mathbf{p}|\Theta), \overline{j}(\mathbf{p}|\Theta)\big)$ on projected qHOG-locations (orange rectangle in Figure 2).

For computing bounds on orientations, we follow a similar procedure. Let $\{\widehat{\mathbf{e}}_t\}_{t=1}^8$ be the set of eight corners bounding $\mathcal{S}_\mathbf{e}$. This time, we consider 64 additional points $\{\widehat{\mathbf{p}}_{st} = \widehat{\mathbf{p}}_s + \widehat{\mathbf{e}}_t\}_{s,t=1}^8$, which are found by translating each $\widehat{\mathbf{e}}_t$ by each $\widehat{\mathbf{p}}_s$. In this way, we cover all extreme arrangements of $\theta \circ \mathbf{p}$ and $\theta \circ (\mathbf{p}+\mathbf{e})$ in $\mathbb{R}^3$. Then, letting $(\widehat{i}_{st}, \widehat{j}_{st})$ denote the camera projection of $\widehat{\mathbf{p}}_{st}$, bounds $\underline{k}(\mathbf{p}, \mathbf{e}|\Theta)$ and $\overline{k}(\mathbf{p}, \mathbf{e}|\Theta)$ on qHOG-orientation, are obtained from extreme phases of 2D vectors $\{(\widehat{i}_{st} - \widehat{i}_s, \widehat{j}_{st} - \widehat{j}_s)\}_{s,t=1}^8$.

(iii) Upper bound on visibility. We consider a primitive to be visible, if at least one of its associated surface normals faces towards the camera, *i.e.*, makes a negative inner product with the ray emanating from camera to its location. In the camera centered WCS, this can be written as

$$\mathcal{V}(\mathbf{p}, \mathbf{n}_1, \mathbf{n}_2|\theta) = \mathbf{1}_{\{\min_{s=1,2}\langle \theta\circ\mathbf{p}, \theta\circ\mathbf{n}_s\rangle < 0\}}. \tag{6}$$

An upper bound on $\mathcal{V}$ over the range of poses, is given by $\max_{\theta\in\Theta} \mathbf{1}_{\{\min_{s=1,2}\langle\theta\circ\mathbf{p}, \theta\circ\mathbf{n}_s\rangle<0\}} = \mathbf{1}_{\{\min_{s,\theta}\langle\theta\circ\mathbf{p}, \theta\circ\mathbf{n}_s\rangle<0\}}$. With $\theta\circ\mathbf{p} = \mathbf{R}(a,b,c)\mathbf{p} + \mathbf{t}(d)$ and $\theta\circ\mathbf{n}_s = \mathbf{R}(a,b,c)\mathbf{n}_s$, the inner product inside the indicator becomes $\mathbf{p}^\top\mathbf{n}_s + \mathbf{t}(d)^\top\mathbf{R}(a,b,c)\mathbf{n}_s$. Letting $\underline{v}_s$ denote the lower bound on the z-components of $\{\mathbf{R}(a,b,c)\mathbf{n}_s : (a,b,c) \in A\times B\times C\}$, which can be found by the bounding scheme in (i) for 3D locations, and noting that $\mathbf{t}(d) = (0,0,d)^\top$, the upper bound on visibility is given by

$$\overline{\mathcal{V}}(\mathbf{p}, \mathbf{n}_1, \mathbf{n}_2|\Theta) = \mathbf{1}_{\{\min_{s=1,2}(\mathbf{p}^\top\mathbf{n}_s + \min(\underline{dv}_s, \overline{dv}_s))<0\}}. \tag{7}$$

(iv) Upper bound on qHOG scores. Let $\mathcal{B}(\mathbf{p}, \mathbf{e}|\Theta)$ denote the sub-box of the qHOG domain, specified by the bounds $\{\underline{i}, \overline{i}, \underline{j}, \overline{j}, \underline{k}, \overline{k}\}$ on the spatio-angular projection of the primitive, obtained in step (ii). Then, an upper bound on qHOG scores is given by $\overline{\mathcal{H}}(\mathbf{p}, \mathbf{e}|\Theta) = \max\{\mathcal{H}(i,j,k) : (i,j,k) \in \mathcal{B}(\mathbf{p}, \mathbf{e}|\Theta)\}$. Computing $\overline{\mathcal{H}}$ in a naive way by scanning its entries within $\mathcal{B}$ will take $O(|\mathcal{B}|)$ time. But, depending on the bin resolution, and the size of the evaluated pose range Θ, $|\mathcal{B}|$ can be as large as the whole qHOG domain, especially in the early phases of the BB algorithm. Nonetheless, by exploiting the L-level quantization of $\mathcal{H}$, we can compute its upper bound with a much smaller cost using integral images. Note that, in our case, we are interested in $\mathcal{H}$'s maximum value within $\mathcal{B}$ rather than its sum over $\mathcal{B}$. Thus, we extract integral images not in the usual way, namely once for the whole histogram, but separately for each of its quantization levels. In particular, letting $h_l(\mathcal{B})$ denote the l^{th}-channel integral image of $\mathcal{H}$ over $\mathcal{B}$, the sought after upper bound on qHOG scores is found by

$$\overline{\mathcal{H}}(\mathbf{p}, \mathbf{e}|\Theta) = \max_{l\in\{1,\dots,L\}} l \times \mathbf{1}_{\{h_l(\mathcal{B})>0\}}, \tag{8}$$

which is the largest level l^*, at which $h_{l^*}(\mathcal{B})$ is nonzero. That would mean an entry of $\mathcal{H}$ with value l^* was indeed present inside $\mathcal{B}$, but there was none greater than it. As a result, bounds on the qHOG scores of each primitive can be computed in less than $O(L)$ time. Besides, this constitutes the only bound computation needed online during inference, once the hierarchy of pose partitions is fixed a priori, and the corresponding visibility bounds and $\mathcal{B}$-boxes in the feature domain are precomputed using steps (i)-(iii).

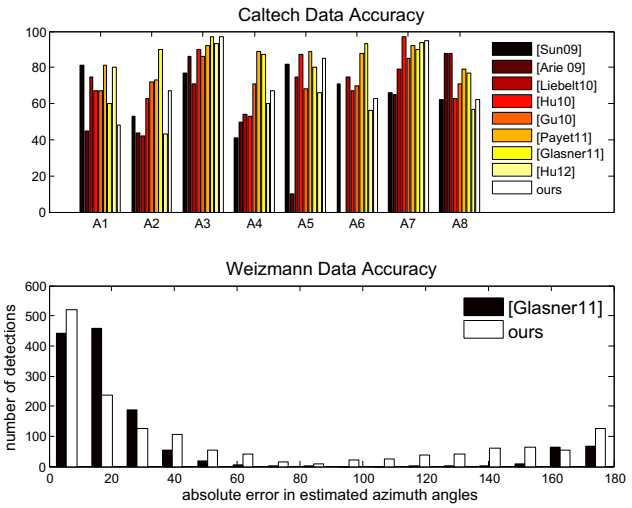

Figure 3: Comparison of our method with the state-of-the-art. Top: Diagonal elements (in %) of confusion matrix on Caltech dataset [20]. Bottom: Histogram of absolute errors in azimuth angles on the Weizmann dateset [8]

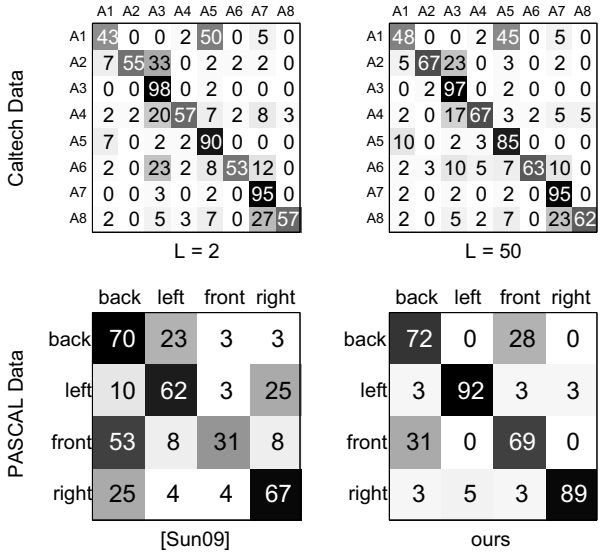

Figure 4: Confusion matrices (in %) of pose estimation. Top-Left: On Caltech data [20] with $L = 2$ quantization levels in qHOG. Top-Right: with $L = 50$. Bottom-Left: Confusion matrix of [24] on PASCAL VOC 2006 car data. Bottom-Right: Our results on the same dataset.

5. Experiments

To show the effectiveness of our approach, we built a 3D model for sedan cars containing a total of $M = 139$ edge primitives (see Figure 1), and experimented it for detection (localization) and pose estimation on the following publicly available data: (i) PASCAL VOC 2006 cars, (i) the Caltech car dataset of [20], and (iii) Weizmann car dataset of [8]. The PASCAL dataset contains car images roughly in four different views, namely front (13 images), back (94 images), left (78 images) and right (64 images), where we used test images, with only one car instance in the scene. The Caltech data includes 10 sets of car images, each set containing 48 images in three different scales, two different heights, and eight different azimuth angles that are roughly multiples of 45 degrees (*i.e.*, front, back, front-left, back-right views and so on) labeled by A1 to A8. The Weizmann dataset on the other hand, has 22 image sets, each containing approximately 70 images with real valued ground truth labels for both azimuth and elevation angles, which enables us to assess error statistics on the estimated pose. We experimented with different levels of quantization L applied to HOG features. $L = 2$ corresponds to a binary valued qHOG, where the upper bound for each primitive takes constant time to compute. As L is increased to 50, the average accuracy of the approach increases modestly (*e.g.*, from an average performance of 69% to 73% on Caltech dataset). As can be seen from Figures 4 and 3, the most significant pose confusion occurs between symmetric views of back (A1) and front (A5) of the car, which is expected, since our approach exclusively relies on shape information rather than appearance. As observed on Caltech data, pose estimation errors also tend to occur more frequently for far objects, which is probably due to our HOG features becoming too coarse to be discriminative in those cases. Inference takes as low as 0.5 seconds with $L = 2$, but it can go up to at most 7 seconds with $L = 50$ (These times are obtained with an unoptimized MATLAB code on an i7-3520M CPU at 2.90GHz). We compared our pose estimations on each dataset (see Figure 5 for example results) with the state of the art. On PASCAL data, we performed significantly better than the compared work by [24] (see Figure 4). On Caltech data, we achieved on the average a better pose estimation than most of the recent methods, except for [17] and [8] (see Figure 3-top), whereas, due to our explicit model fitting approach, our detection (localization) results, which are quantified by the standard 50% VOC overlap criterion, are relatively more accurate than the literature (see Figure 5-left). On Weizmann data, we compared our method with the publishers of the dataset [8], where we report the distribution of absolute error in the azimuth angle (see Figure 3-bottom). Although, we managed to detect significantly more cars within 10 degrees of precision, our erroneous estimations can be slightly more off than [8] with a median error of 23.2°. Given our promising results, it should also be noted that (i) We did not do any training on any of the datasets, nor on real images, as other work do (*e.g.*, in Caltech data, the standard approach is to use first 5 of the total 10 car collections for training, and the last 5 for testing. In that aspect, we report there our results on twice as many examples), and (ii) All of the compared approaches heav-

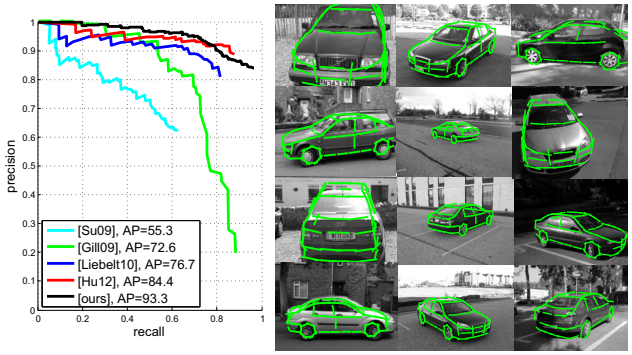

Figure 5: Localization results. Left: precision-recall curves of object detection in Caltech dataset. Right: example detections on PASCAL (first column), Caltech (second column), and Weizmann (third column) datasets. Models under the estimated pose are overlayed to the input image, where only visible edges (identified by (6)) are shown.

ily rely on discriminative training on viewpoint-dependent appearance information of individual object parts, whereas our method almost solely relies on shape (only a few primitives in the front and back of our 3D car model are attributable to texture boundaries between headlights and the car body). Therefore, it is expected that our approach lacks some important discriminative power due to appearance.

6. Conclusion

In this paper, we presented a new and computationally very efficient approach for object localization and fine grained pose estimation. Our method uses an explicit 3D wireframe model of the category of interest, which is generated from blueprint sketches. For pose estimation, model's edge primitives are matched to the HOG pattern on the input image, where optimization is achieved very rapidly (with costs as low as half a second) by a BB algorithm formulated over the 6D pose space. We achieved performances comparable to the state of the art as demonstrated with experiments in three publicly available datasets on car objects. As a future work, we intend to include to our 3D model, additional primitives with attributes like color or flatness of intensity etc, to improve performance. Another important extension is to use our blueprint extracted model as an initial template to guide model learning from actual images.

References

[1] G. J. Agin and T. O. Binford. Computer description of curved objects. *IEEE Trans. Comput.*, 25(4):439–449, 1976.

[2] M. Arie-Nachimson and R. Basri. Constructing implicit 3D shape models for pose estimation. In *ICCV*, 2009.

[3] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

[4] T. O. Binford. Visual perception by computer. In *IEEE Conference on Systems and Control*, 1971.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[6] P. F. Felzenszwalb, D. A. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[7] M. Fritz, K. Saenko, and T. Darrell. Size matters: Metric visual search constraints from monocular metadata. In *NIPS*, 2010.

[8] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *ICCV*, 2011.

[9] C. Gu and X. Ren. Discriminative mixture-of-templates for viewpoint classification. In *ECCV*, 2010.

[10] W. Hu. Learning 3D object templates by hierarchical quantization of geometry and appearance spaces. In *CVPR*, 2012.

[11] W. Hu and S. C. Zhu. Learning a probabilistic model mixing 3D and 2D primitives for view invariant object recognition. In *CVPR*, 2010.

[12] I. Kokkinos. Rapid deformable object detection using dual-tree branch-and-bound. In *NIPS*, 2011.

[13] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(12):2129–2142, 2009.

[14] J. Liebelt and C. Schmid. Multi-view object class detection with a 3D geometric model. In *CVPR*, 2010.

[15] D. Marr and H. K. Nishihara. Representation and recognition of the spatial organisation of three-dimensional shapes. *Proc. of the Royal Society of London*, B200:269–294, 1978.

[16] M. Özuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *CVPR*, 2009.

[17] N. Payet and S. Todorovic. From contours to 3D object detection and pose estimation. In *ICCV*, 2011.

[18] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3D geometry to deformable part models. In *CVPR*, 2012.

[19] L. G. Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology. Dept. of Electrical Engineering, 1963.

[20] S. Savarese and F.-F. Li. 3D generic object categorization, localization and pose estimation. In *ICCV*, 2007.

[21] S. Savarese and F.-F. Li. View synthesis for recognizing unseen poses of object classes. In *ECCV*, 2008.

[22] H. Schneiderman. Feature-centric evaluation for efficient cascaded object detection. In *CVPR*, 2004.

[23] H. Su, M. Sun, F.-F. Li, and S. Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *ICCV*, 2009.

[24] M. Sun, H. Su, S. Savarese, and F.-F. Li. A multi-view probabilistic model for 3D object classes. In *CVPR*, 2009.

[25] A. Thomas, V. Ferrar, B. Leibe, T. Tuytelaars, B. Schiele, and L. Van Gool. Towards multi-view object class detection. In *CVPR*, 2006.

[26] P. Yan, S. M. Khan, and M. Shah. 3D model based object class detection in an arbitrary view. In *ICCV*, 2007.