

Multiview Feature Distributions for Object Detection and Continuous Pose Estimation

Damien Teney*

*Department of Computer Science
University of Bath
Bath, BA27AY, United Kingdom*

Justus Piater

*Intelligent and Interactive Systems
University of Innsbruck
Technikerstrasse 21a, A-6020 Innsbruck, Austria*

Abstract

This paper presents a multiview model of object categories, generally applicable to virtually any type of image features, and methods to efficiently perform, in a unified manner, detection, localization and continuous pose estimation in novel scenes. We represent appearance as distributions of low-level, fine-grained image features. Multiview models encode the appearance of objects at discrete viewpoints, and, in addition, how these viewpoints deform into one another as the viewpoint continuously varies (as detected from optical flow between training examples). Using a measure of similarity between an arbitrary test image and such a model at chosen viewpoints, we perform all tasks mentioned above with a common method. We leverage the simplicity of low-level image features, such as points extracted along edges, or coarse-scale gradients extracted densely over the images, by building probabilistic templates, i.e. distributions of features, learned from one or several training examples. We efficiently handle these distributions with probabilistic techniques such as kernel density estimation, Monte Carlo integration and importance sampling. We provide an extensive evaluation on a wide variety of benchmark datasets. We demonstrate performance on the “ETHZ Shape” dataset, with single (hand-drawn) and multiple training examples, well above baseline methods, on par with a number of more task-specific methods. We obtain remarkable performance on the recognition of more complex objects, notably the cars of the “3D Object” dataset of Savarese *et al.* with detection rates of 92.5% and an accuracy in pose estimation of 91%. We perform better than the state-of-the-art on continuous pose estimation with the “rotating cars” dataset of Ozuysal *et al.* . We also demonstrate particular capabilities with a novel dataset featuring non-textured objects of undistinctive shapes, the pose of which can only be determined from shading, captured here by coarse scale intensity gradients.

Keywords: appearance-based object recognition; object detection; pose estimation; Hough voting; edges and shape models; viewpoint synthesis

*Corresponding author.

Email addresses: d.teney@bath.ac.uk (Damien Teney), justus.piater@uibk.ac.at (Justus Piater)

1. Introduction and related work

This paper is concerned with the joint recognition and pose estimation of object categories in 2D images. Recognizing that these two tasks represent two sides of a same problem, we tackle them in a unified approach. In general, the pose (3D orientation) of objects cannot be inferred from just one type of image information, e.g. silhouette and edges, to cite a common example. Additional visual cues may be necessary, such as the shading onto the object surface. A key point of our contributions is thus to provide techniques generally applicable in this regard, even to low-level, dense and/or non-descriptive image features. To perform *continuous* pose estimation, our object model captures, in addition to the appearance at discrete training viewpoints, the deformations between these, detected from the optical flow between training examples. A measure of similarity between generated views of the object (possibly at an unseen viewpoint) and a test image allows us to perform detection, recognition, and pose estimation in a unified manner. The following paragraphs present the principal motivations and key points of the method, comparing them to existing related work. Parts of these contributions were introduced in earlier publications [1, 2].

1.1. Tasks considered

The recognition of objects in 2D images encompasses a number of tasks, detailed below, which are often considered as separate research problems. They are however closely related, and we handle them all with the same model and methods. Notably, we do not train discriminative models, which is the usual approach for the classification tasks.

Localization The goal is to identify the parts of the test image that belong to the object of interest, versus the parts of the image that correspond to background clutter. The result of localization is typically a set of bounding boxes, which encircle candidate objects in the image, each accompanied with a detection score. We handle this task with an algorithm similar to the generalized Hough voting scheme. The model of the object can be learned from *one* or *several* training examples: we handle both cases identically by modeling distributions of features through kernel density estimation (see Section 2).

Detection One must decide whether the object of interest appears in the test image or not. This task can be performed alone, or by setting a threshold on scores of localizations to obtain binary detection results.

Classification (among objects or among discrete poses) One must determine which object or which viewpoint among learned ones appears in the image. This traditionally involves learning discriminative classifiers. In our method however, we simply build generative models independently for each learned object or viewpoint, and determine the best match from the similarity measured between the test image and one of those models.

Continuous viewpoint (pose) estimation This more challenging task is handled by extending our generative models to also synthesize unseen (untrained) viewpoints.

1.2. Modeling object appearance

The method for performing recognition of objects in 2D images depends heavily on the internal representation chosen to model the appearance of those objects. We are interested in building models of appearance for object *categories* (or “classes”) rather than specific instances, thus capable of recognizing, to some extent, unseen objects that are similar to a category learned from a few

training examples. The goal is for example to train the system with a set of different cars, then to recognize the pose of a new, unseen car. The categories in such a scenario are defined implicitly by the training instances used as examples. In the proposed approach, the appearance of the object under a specific viewpoint is modeled as the distribution of low-level image features, represented, in a non-parametric manner, by the actual image features of one or several training images of the objects under that specific viewpoint. We therefore handle variability in appearance in a probabilistic way; this variability among the training examples can equally come from different objects of a same category, or from variations of appearance of a unique object, e.g. observed under different conditions of illuminations.

1.3. 2D and 3D object models

We choose to model the 2D appearance of the objects without explicit knowledge of their underlying 3D shape. The motive for this choice is to handle more easily and naturally the variability within categories, both in appearance *and shape*. As a result, the model is thus trained with simple example images. Existing methods have used explicit, geometrical, 3D models of objects [3], but the modeling of variations in appearance is generally limited in regards with shape [4, 5]. One exception is the work of Glasner *et al.* [6], which uses structure-from-motion to reconstruct accurate 3D models from the training images. They then account for within-category variability simply by merging multiple exemplars in their non-parametric model, in a fashion very similar to the one we use (with our 2D training examples). One drawback of their approach is the initial need for a large number of views to reconstruct accurate 3D models. In comparison, our exemplar-based model can use an arbitrary number of views, which do not need to overlap, and the model can be incrementally updated as more views become available.

1.4. Object localization and detection

Object localization and detection among clutter is commonly achieved with variants of either the “sliding window” or the “Hough voting” approaches. The former (used e.g. in [7]) uses a binary classifier, which is evaluated on a uniform sample of image locations and scales. Such an exhaustive search may prove computationally expensive, and many heuristics have been proposed to alleviate this issue [8]: salient regions, coarse-to-fine-search, etc. Voting techniques based on the well-known generalized Hough transform [9] provides another way to alleviate the complexity issue. Probabilistic formulations of this voting technique have been proposed through the implicit shape models [10, 11]. Our algorithm for detection uses this voting scheme, applied to low-level, dense image features. Hough voting was extended to discriminative framework by Maji and Malik [12], by computing optimal weights to the image features of the model. They obtained excellent results, further improved by a subsequent verification step, in which the initial detections are rescored by an SVM-based classifier. We reuse this idea of weighting parts of the learned model; the exact procedure is slightly different, and suited to our non-discriminative features. Although not a central element of our contributions, we will show that this weighting often brings substantial improvements.

1.5. Choice of image features

The type of image features used to encode the appearance of the objects is a crucial choice. Some methods historically used of the appearance of the object as a whole [13–15], but with the common downsides of poor robustness to occlusions and a need for large numbers of training views. At the opposite end, *feature-based* methods have relied on “interest points”, precisely located in the images, and characterized by hand-designed descriptors of local appearance, such as SIFT descriptors [16]. Those discrete points can then be matched between the test image and the training examples [3].

While this approach has proved to be highly successful and efficient in many cases, the extraction of such discriminant image features cannot be relied upon in general cases, as it often fails with non-textured objects. The basic approach also does not readily extend to variability within categories. A recent trend is to describe image contents with similar descriptors of appearance over a *dense* grid across the image, such as done by the successful histograms of oriented gradients (HOG) [7], also used within the state-of-the-art detector of Felzenszwalb *et al.* [17]. The idea behind those descriptors is to capture statistics or distributions of primitive characteristics (such as intensity gradients) over local image regions. We believe that this approach is indeed the most generally-applicable one, and is the central motivation for our technique. Similarly to, e.g. HOGs, our “distributions of features” capture local statistics densely over the images, but we do not depend on hand-designed descriptors, and we offer a unique formulation suitable to different types of image features. Another notable difference of our method with HOGs is to use gradients extracted at a coarse scale, intended to capture shape (rather than pure appearance) of smooth surfaces, whereas HOGs were most successful with gradients extracted at a much smaller scale, thus essentially capturing sharp transitions like edges.

Most current, state-of-the-art methods for object recognition rely on the use of image edges (e.g. [18, 19], among many others), seen as an efficient representation of the silhouette and shape of objects. The typical technique basically consists in building intermediate representations such as contour fragments, which can then be matched discriminatively between training and test images, and used e.g. in a Hough voting scheme. Our approach, which leverages the simplicity of low-level, fine-grained image features, can be applied to edges by considering all edge pixels of the image as features. At the cost of higher computational costs, this approach leads to excellent results as well, while satisfying our aim for a general and straightforward formulation.

A large area of research has focused on the modeling and detection of deformable shapes (see [18] for a review). Interestingly, our simple approach proves competitive with some of those techniques, as demonstrated on the ETHZ shape dataset. Although we neither model continuous contours nor their variations explicitly, our low level features (edge points) can encode similar variations to some degree. Another advantage of our our method is its ability to learn shape models similarly from a single or multiple examples, and from only *loosely* segmented images (with a bounding box). Such capabilities are not commonplace in the domain of shape matching, but were also offered by the work of Ferrari *et al.* [18].

Finally, our capability of handling *dense* image features is demonstrated and used with great advantage with intensity gradients, extracted at a coarse-scale over the whole images. Using such gradients provides unique capabilities, as it allows one (1) to effectively handle non-textured objects (see Section 5.6), and, even more importantly, (2) to resolve cases where edges alone would only offer ambiguous information on the presence or the pose of an object in a scene. Indeed, the shading over homogeneous surfaces, captured by such gradients, may sometimes be the sole relevant clue, in particular to identify the exact pose of certain objects, or, for example, to differentiate between hollow versus full objects of similar shapes (see our experiments in Section 5.6).

1.6. Multiview models of appearance and pose estimation

Object recognition with 2D training examples typically uses viewpoint-specific models, e.g. a model for cars seen from the front, and another for cars seen from the side. Recent contributions have included more and more techniques that handle multiple registered training viewpoints. The object in the test image is then matched against one of these viewpoints and allows performing a coarse estimation of its pose (or 3D orientation) also called *pose classification*. We refer to this basic approach as a “nearest-neighbour” pose estimation. Some applications (robotic interaction and grasping for example) require however a more precise estimation of the pose [20, 21]. This

capability was commonly reserved to recognition methods using 3D object models. As discussed above though, they do not cope well with object *categories*, which are clearly very challenging with regards to the task of pose estimation. Few appearance-based methods have been designed to provide this capability [14]. Most recent multiview models of appearance consider the different training viewpoints independently [21–25], while others try to match and link features across viewpoints [26–28]. Savarese *et al.* [27], for example, model an object as a collection of planar parts that can appear in different views. We follow an intermediate approach, by storing independently the image features that make up the different views, but we also store, along with every each image feature, how its appearance varies with respect to the pose of the object. The multiview models mentioned above only performed localization and classification such as “frontal view” or “side view”, whereas we allow precise, *continuous* pose estimation.

Simple techniques have been proposed to improve the precision of nearest-neighbour pose classification. They typically involve voting in the 3D pose space followed by averaging [21] or probabilistic smoothing schemes [1, 25, 29], leading to a precision beyond the resolution of viewpoints given as training examples. While those simple techniques have sometimes given very interesting results, we rather chose, in the work presented here, to explicitly detect, and include in the model, the changes of appearance between the discrete viewpoints seen during training (practically, how image features translate in the image, and thus how the appearance “deforms” between neighbouring viewpoints). This information extends our generative model, which can now synthesize arbitrary, untrained viewpoints. We can then finely optimize the 3D pose, starting from the initial nearest-neighbour estimates. Let us mention the work of Torki and Elgammal [30]. In their radically different approach to appearance-based pose estimation, they learn a direct regression from local image features to the pose of the object. This original approach recovers a precise pose, but cannot handle significant clutter or occlusions, and the accurate pose estimation depends on the (supervised) enforcement of a one-dimensional manifold constraint (corresponding to the 1D rotation of the object in the training examples). It is not clear how that approach would extend to the estimation of the full 3D pose of an object. Other recent works such as [31] have looked further at manifold modeling for appearance-based pose estimation, but with an evaluation limited to fairly simple conditions, and the performance of such methods for detection in cluttered scenes is not obvious.

1.7. Synthesis of novel viewpoints

During an off-line training phase, we use an optical flow algorithm between pairs of images to detect how the appearance of each training object varies between these viewpoints. The image features extracted from one of these images can then be deformed into the other, and the interpolation for intermediate viewpoints is straightforward. We thereby obtain a generative model that synthesizes the appearance of the object in any (possibly unseen) viewpoint. This procedure is related to the technique of *morphing* in computer graphics [32–34], with the difference that we are considering arbitrary numbers of input views, and we do *not* rely on established correspondences between specific landmarks of the input views. This similarly contrasts with the competing method of Savarese *et al.* [35], which does use specific correspondences between nearby views. Our advantage is to handle non-textured objects with little detail. Although some global consistency in the detected deformations is enforced by the optical flow algorithm, each image feature independently stores its possible deformations. This does not limit the model to a particular class of transformations. In comparison, Savarese *et al.* [35] specifically models *affine* transformations of object parts, assuming that objects are made of large planar parts. We also use a *sparse* set of training views (typically spaced about 20° apart on the viewing sphere) and do not require videos or dense sequences of images to track features between frames, as opposed to Sun *et al.* [36].

1.8. Summary of contributions

Our main contributions can be summarized in the following points.

1. We present a general framework for modeling the appearance of objects and object categories, suitable to virtually any type of image features, applicable for detection and recognition without relying on hand-designed local visual descriptors, while still providing performance and efficiency on par with state-of-the-art — arguably more complex — methods.
2. We show how to handle dense, unmatchable image features, such as coarse-scale intensity gradients. This ultimately enables the method to recognize objects without texture, and to handle cases where shading constitutes the sole source of unambiguous visual information.
3. We provide a technique for identifying, and storing, within a multiview model of appearance, how the appearance varies between discrete training viewpoints. This ultimately allows performing *continuous* pose estimation of an object in a novel scene, without relying on an explicit 3D model of the object. This also readily applies to object *categories*, and not only to specific objects.

2. Probabilistic model of appearance

This section presents our model of appearance with a bottom-up description. We start by turning a set of image features of a given image into a “distribution of features”, then use those representations to form our model that includes several viewpoints, and possibly several training examples for each viewpoint. We finally show how to detect and recognize those training views in a novel test image.

2.1. From image features to probability distributions

Our approach is based on a representation of images as continuous probability distributions of image features. The motivation for representing images as distributions is twofold. First, this representation accounts for the inevitable uncertainty of the description of any single image, due to e.g. image noise, quantization errors, uncertainty during feature extraction, etc. Secondly, it also provides, as we will see in the next section, a way of modeling variability in appearance of an object or object category, e.g. given several different examples of this category. It will also give us a more abstract representation of the images that is convenient to manipulate with existing probabilistic techniques, and that generally applies to any type of image features. The approach is first applied and presented for a test image — in which we want to recognize the object of interest — while the next section will then apply it to the training examples.

We start off by extracting, from a given test image, different types of features (detailed in Section 4.1), each type denoted by an index $f = 1 \dots F$. These can be as simple as the pixels belonging to edges (which we call “edge points”), or to the value of the intensity gradients for all pixels of the image (“gradient points”). In general, each feature x is thus characterized by (1) its position in the image, noted $x.pos \in \mathbb{R}^2$ and (2) some appearance attributes, noted $x.app$. In the case of edge points, we use, as an attribute, the local orientation of the edge (an angle in $S_1^+ = [0, \pi[$); in the case of gradient points, we use the orientation and the magnitude of the gradient. The contents of a given test image form thus, for each type f of features, a set $\text{test}^f = \{x_i\}_i$, with $x_i \in \mathcal{A}^f$, the domain of these features. For example with edge points, $\mathcal{A}^{\text{edges}} = \mathbb{R}^2 \times S_1^+$ (see Section 4.1 for details).

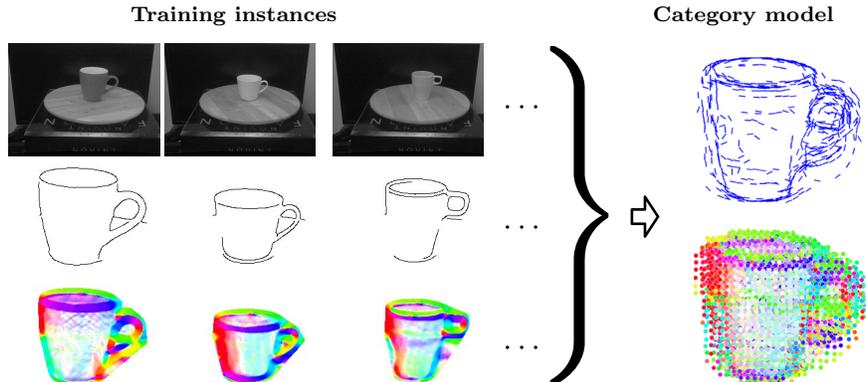


Figure 1: We model the appearance of an object with continuous probability distributions of image features, defined from one or several training examples (left). The model can be visualized by drawing samples (right) from those distributions, in this case as edge points and gradient points (hue/saturation represent respectively orientation and magnitude).

We now show how to turn such a set of discrete image features (from a given test image) into a continuous probability distribution. We define and represent such distributions over the appearance space of image features (\mathcal{A}^f) in a non-parametric manner, through kernel density estimation (KDE). With this procedure, all image features are used as particles supporting simple kernels, the sum of which represents a continuous distribution. Formally, for each type of image features f , we use the set of features test^f extracted from our test image to define the distribution

$$\phi_{\text{test}^f}^f(x) = \sum_{x_i \in \text{test}^f} \text{wt}(x_i) \mathcal{N}(x_i.\text{pos}; x.\text{pos}, \sigma_{\text{pos}}) \mathcal{K}^f(x_i.\text{app}; x.\text{app}), \quad (1)$$

with $x \in \mathcal{A}^f$, \mathcal{N} a Gaussian kernel for the position of the features, \mathcal{K}^f a kernel for their appearance attributes (see Section 4.1), and $\text{wt}(x_i)$ the weight of the feature x_i . Those weights are set uniformly for the features of a test image, i.e. $\text{wt}(x_i) = \frac{1}{|\text{test}^f|} \forall x_i \in \text{test}^f$. This representation with KDE will be reused for the *training* images, where the weights will then take a more complex form (Section 2.4). Practically, Eq. 1 gives us a probability density function that can be easily evaluated for any x . For example, in the case of edge points, we can evaluate the probability of observing a horizontal edge at a specific location in the image.

2.2. Application to object categories and to multiple views

We have represented our test image as continuous distributions of image features. We will now similarly apply that approach to the training images. Two differences are worth mentioning though.

First, we may observe the object of interest under *multiple* viewpoints. Each training image t corresponds to a viewpoint $v_t \in S^2$ (a point on the viewing sphere), and gives, a set of features $\text{train}_{v_t}^f$ for each type of feature f (defined similarly to the sets test^f above). Those multiple viewpoints are considered independently at this point, and they each define distributions $\phi_{\text{train}_{v_t}^f}^f$ as in Eq. 1. Only in Section 3 will we consider multiple viewpoints together, in order to perform continuous pose estimation. As a first step though, we are only interested in recognizing (approximately at least) one of the discrete viewpoints provided as the training examples.

Second, we may be provided with training images of several, different objects (object “instances”) representative of an object *category*. We assume that all training images are aligned and at the same scale, which can be practically done automatically as explained in Section 5. We now want our distributions of features to reflect statistics relevant to all the different training examples. This is straightforward within our formulation with a KDE: for each viewpoint v_t , we simply include, in the set of features train_v^f , the features extracted from *all* training images corresponding to that viewpoint (Fig. 1). The resulting distributions $\phi_{\text{train}_v^f}^f$, as defined earlier, are then representative of the occurrence of image features among all those training examples together, and they constitute our model of appearance of an object *category*. Consequently, the appearance of that category is thus defined implicitly by the instances provided as training examples.

2.3. Use of proposed model for detection and recognition in a new image

We now would like to detect, or recognize the learned object in the test image. The solution to this task consists in the optimal set of in-plane transformations w^* (a translation, rotation and scaling in the image) and viewpoint (out-of-plane transformations) v^* ($\in S^2$), which corresponds to the training viewpoint recognized in the test image. Let us mention, as a side note, that this result (v^*, w^*) presents 6 degrees of freedom (DoF), and that it can be equally described in the image space (as we do) or in the “world” space (as Euclidean coordinates for position and orientation). The latter is usually preferred in the field of robotics, and commonly called the 6-DoF pose of the object. Both representations are however equivalent and interchangeable, provided the calibration of the camera.

We will first present how to measure the visual similarity between the test image and the learned object at a specific viewpoint and in-plane transformations. We will then provide an algorithm to identify the optimal set of such transformations, determining the local maxima of that similarity. At this point, we still consider the training viewpoints independently, and thus perform a “nearest-neighbour” classification of the viewpoint. This will serve as a starting pointer later, for a local optimization procedure to perform *continuous* pose estimation (Section 3).

Let us consider a test image is represented by the distributions of features ϕ_{test}^f , and a specific training view t represented by $\phi_{\text{train}_v^f}^f$. This training view may appear in the test image under any similarity transformations w (in-plane translation, rotation, scaling), trivially applied by a function $\text{transform}_w(x)$. Accounting for such transformations, we measure the similarity between the test and training views with the cross-correlation of the distributions

$$(\phi_{\text{test}}^f \star \phi_{\text{train}_w^f}^f)(w) = \int_{\mathcal{A}^f} \phi_{\text{test}}^f(x) \phi_{\text{train}_v^f}^f(\text{transform}_w(x)) \, dx \quad . \quad (2)$$

To efficiently obtain an approximate evaluation the integral of Eq. 2, we use Monte Carlo integration [37]. This involves drawing samples x_i ($\ell = 1 \dots L$) from the distribution ϕ_{test}^f (see Section 4.3), and computing the following sum:

$$(\phi_{\text{test}}^f \star \phi_{\text{train}_v^f}^f)(w) \approx \frac{1}{L} \sum_i^L \phi_{\text{train}_v^f}^f(\text{transform}_w(x_i)) \quad . \quad (3)$$

We can substitute the distribution $\phi_{\text{train}_v^f}^f$ by its definition with KDE (as in Eq. 1). Assuming this distribution is represented by L' particles x_j (either the original image features extracted from the

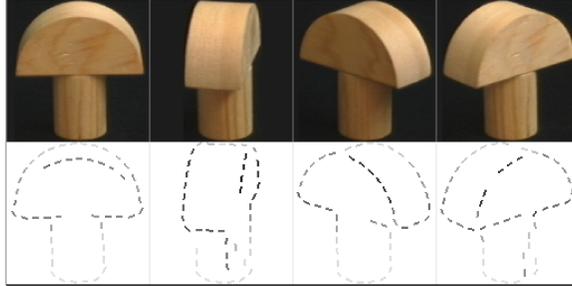


Figure 2: Higher weights (darker colors) are assigned to the image features most informative to both detection and pose estimation. On this toy dataset (18 images of the object rotating around one axis), the cylindrical base, which looks similar in different views, receive lower weights. Non-silhouette edges, which can unambiguously determine a precise pose, receive high weights.

training images, or a resampled set of those as will be discussed in Section 4), we have

$$(\phi_{\text{test}}^f \star \phi_{\text{train}_v}^f)(w) \approx \frac{1}{LL'} \sum_i^L \sum_j^{L'} \text{wt}(x_j) \mathcal{N}(x_i.\text{pos}; \text{transform}_w(x_j.\text{pos}), \sigma_{\text{pos}}) \mathbb{K}^f(x_i.\text{app}; x_j.\text{app}). \quad (4)$$

Now, taking into account several types f of image features ($f = 1 \dots F$), the full similarity measure between two images finally uses the product over f of the expression above, which gives

$$\text{similarity}_{\text{test}, \text{train}_v}(w) = \prod_f (\phi_{\text{test}}^f \star \phi_{\text{train}_v}^f)(w). \quad (5)$$

We now have the core of the proposed method, with equations 4 and 5: we can easily evaluate the likelihood of observing, in the test image, the object under the viewpoint v and in-plane transformations w . The solution to the problem of object localization corresponds the maxima of Eq. 5, i.e.

$$(v^*, w^*) = \arg \max_{v, w} (\text{similarity}_{\text{test}, \text{train}_v}(w)). \quad (6)$$

Our algorithm to solve this maximiation problem is detailed in Section 4.2. It efficiently computes the values of the objective function over all image locations (in-plane translations), with a method similar to a Hough voting using samples drawn from our distributions of features.

2.4. Weighting of image features

We now present how to assign adequate weights to samples drawn from the trained model. The model of appearance presented in Sections 2.1 and 2.2 is merely a convenient way of representing the appearance of object categories. Since our goal is specifically to use this model to detect an object among clutter, and to determine its actual pose, we wish to give more weight its parts that are most informative to those tasks. As will be detailed in the Implementation section (Section 4), we choose to preselect samples offline from the trained model for efficiency. Therefore, the weights associated to these samples can also be computed in a pre-processing step, using the procedure described below.

Weighting training data in the context of object recognition is common among many existing methods [12, 38–40], where it has shown to increase performance significantly. In comparison to

existing methods, our procedure is better suited to non-discriminative low-level image features, and does not rely on large amounts of training examples. It iteratively uses a validation test set to weight each feature relative to how informative it is to discriminate the appearance at a specific pose, versus other poses and against background clutter.

The procedure is performed for each type f of image feature separately; we omit the superscripts f in the following paragraph to lighten the notations. We initially run the algorithm for detection and pose estimation (Section 2.3) with uniform weights on all image features of the training data. The idea is then to decrease the relative weight of those features that lead to incorrect results, from false positive detections (object identified in the background clutter) or from the recognition of incorrect poses (e.g. a car facing right identified as a car facing left). For each training view t (corresponding to a viewpoint v_t), we obtain some incorrect results $\{(v_n, w_n)\}_n$ ($n = 1 \dots N$) to be used as negative examples (typically a pose estimate off by 20° or more, or an overlap of the detection bounding box less than 0.5 with the ground truth). We then update the weights of all image features x_i of the training view t according to a three step rule:

$$\begin{aligned} \text{wt}'(x_i) &= 1 - \frac{1}{N} \sum_n \phi_{\text{train}_{v_t}}(\text{transform}_{w_n^{-1}}(x_i)) \\ \text{wt}(x_i) &\leftarrow \lambda \text{wt}'(x_i) + (1 - \lambda) \text{wt}(x_i) \\ \text{wt}(x_i) &\leftarrow \text{wt}(x_i) / \sum_i \text{wt}(x_i). \end{aligned} \tag{7}$$

The first of these steps evaluates the contribution of the image feature x_i to the negative examples (incorrect results), by simply measuring how well that feature “matches” with the training view superimposed onto the test view (according to the in-plane transformations w_n). The weights are then updated (step 2, with learning rate $\lambda = 0.5$, typically), and normalized as to always sum to 1 (step 3). The effect of these steps is thus to actually decrease the relative weight of the features that lead to misdetections or misclassifications of the pose. The whole procedure is then repeated iteratively: detection is performed, again, on the same validation dataset, but with the new weights for the model, which gives different negative examples, that are used with the three step rule to update the weights. As shown through our experiments, stable weights are usually reached within the order of 4–5 iterations (Section 5.2, Fig. 9).

Note that, if no validation test set is available, the weights can still be computed as described above by reusing, as validation test set, the training images themselves. When performing detection on the training images, the difficulty is then essentially to recognize the object in one viewpoint versus the other viewpoints (and not versus clutter). As a result, the weights then learned from negative results will help to differentiate each training viewpoint: higher weights are given to the image features that are very informative to a specific viewpoint (Fig. 2). This effect is similar to the one obtained in earlier work [1].

Finally, let us remark that the weighting scheme proposed here could be compared to the classical “term frequency – inverse document frequency” approach used in text mining, where high weights are assigned to words (image features, in our application) specific to a class of documents to retrieve (a specific viewpoint, here), relative to their likelihood of occurrence in general (in background clutter, in our case) [41].

3. Continuous pose estimation

The appearance model presented so far treats the different viewpoints provided in the training data independently, and performs a *coarse* pose estimation, or pose classification, by recognizing one of those discrete viewpoints. Our objective is now to provide a more accurate estimate of the

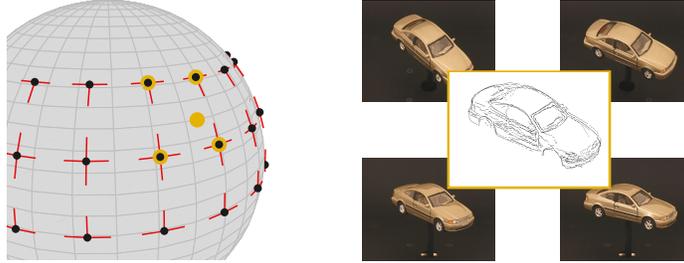


Figure 3: Generative model of appearance for novel viewpoints. The training viewpoints are typically regularly distributed on the viewing sphere (left, black dots). Deformations between adjacent viewpoints (red segments) are detected with an optical flow algorithm, and allow interpolation of object appearance at a novel viewpoint (orange dot and center picture), by combining and deforming the image features of nearby viewpoints (orange circles and outer pictures).

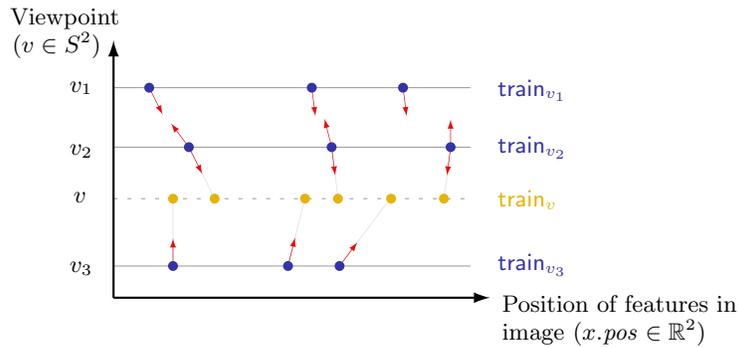


Figure 4: Schematic representation of the training data, and of the generative model for novel viewpoints. Image features (blue points) are available for some discrete training viewpoints v_t , and the deformations (translations in the image plane, red arrows) are detected between adjacent viewpoints during a preprocessing step. These translations are linearly interpolated to infer the appearance of the object at a novel viewpoint v (the set of image features train_v). The novel view includes the image features from several nearby training views, v_2 and v_3 here.

pose, beyond the resolution of the training viewpoints. We first present a generative model capable of synthesizing the appearance of the learned object (or object category) at an arbitrary viewpoint, interpolating between the known views, then we show how to use it for a local optimization of initial (coarse) results.

3.1. Generative model of appearance for arbitrary viewpoints

The goal of our generative model is basically to fill in the gaps between the discrete training viewpoints. Although it is sometimes possible to establish explicit correspondences between image features of nearby training views, this approach could not be relied upon in general, as it does not generalize to dense or non-discriminative image features. Therefore, we chose instead to identify *dense* deformations between pairs of adjacent training views, using an optical flow algorithm. Those deformations are then combined and linearly interpolated to deform the image features of the training images into any arbitrary viewpoint (Fig. 3 and 4).

More precisely, we first define a function $\text{dist}(v, v')$ that measures the angular distance between two viewpoints on the viewing sphere. We define the set of all pairs of neighbouring training viewpoints $\mathcal{V} = \{(t, t') : \text{dist}(v_t, v_{t'}) < th\}$ (with a threshold of $th = 20^\circ$ typically). During an off-line training phase, an optical flow algorithm [42] is applied on all pairs of views $(t, t') \in \mathcal{V}$ ¹. Each pair produces a dense flow map $\text{UV}_{t \rightarrow t'}(x)$ that corresponds, in our case, to the local deformation (translation in the image plane) undergone at an image location x when moving from viewpoint v_t to $v_{t'}$. We can now define our generative model noted train_v , which corresponds to the set of image features defining the appearance of the object at a novel viewpoint v , as the union of image features of nearby training views, translated appropriately using the precomputed deformations. Formally,

$$\text{train}_v = \bigcup_{v_t: \text{dist}(v_t, v) < th} \text{deform}_{v_t \rightarrow v}(\text{train}_{v_t}). \quad (8)$$

The function $\text{deform}_{v_t \rightarrow v}$ adjusts the position of the image features of a training view v_t into the novel viewpoint v . It uses a linear combination of two² precomputed deformations, in order to translate each image feature adequately. We denote these two deformations by the indices of the two viewpoints between which we computed them, and call them (t, t') and (t, t'') . They are chosen from \mathcal{V} so that the novel viewpoint can be reached (on the viewing sphere) by a positive linear combination of them. Therefore, $\exists \alpha, \beta \in \mathbb{R}^+ : v = v_t + \alpha(v_{t'} - v_t) + \beta(v_{t''} - v_t)$. Practically, this means that the viewpoints $v_t, v_{t'}$ and $v_{t''}$ cannot be collinear on the viewing sphere. In the simple case where training viewpoints spaced on a grid (as in the experiments of Section 5), we simply choose $v_{t'}$ and $v_{t''}$ respectively along the changes in azimuth and elevation. It is now straightforward to define the function that combines the two deformations:

$$\begin{aligned} \text{deform}_{v_t \rightarrow v}(\text{train}_{v_t}) = & \left\{ x'_i : \begin{aligned} x'_i.\text{pos} &= x_i.\text{pos} + \alpha \text{UV}_{t \rightarrow t'}(x_i.\text{pos}) \\ &+ \beta \text{UV}_{t \rightarrow t''}(x_i.\text{pos}) \end{aligned} \right. \\ & \left. \text{and } x'_i.\text{app} = x_i.\text{app}, \quad \forall x_i \in \text{train}_{v_t} \right\}. \end{aligned} \quad (9)$$

The appearance of the image features is thus left unchanged, but their position in the image is modified using a linear combination of the deformations detected with optical flow. Using a parameterization of the viewpoint with euler angles as we do in our implementation (Section 4.4),

¹When building a model of an object *category*, the deformations are detected using pairs of views of a single object instance at a time, since the detection of optical flow requires fairly similar images to succeed.

²The use of two precomputed deformations accounts for the two dimensions of the viewing sphere.

this linear interpolation of image location with respect to angles is a simplistic approximation of the underlying transformations (3D rotation and projection onto the image plane). This linear approximation however proved appropriate, since the deformations are detected between fairly close viewpoints (due to the limitations of the optical flow algorithm), and more complex interpolation schemes did not prove more effective in practice.

3.2. Local optimization for the pose of initial detections

We use the algorithm of Section 2.3 to obtain initial detections and recognitions of training poses. Those are then used as starting points to run a local optimization, using the generative model described above, in order to refine and obtain a precise pose estimate. The objective function to maximize during this optimization is still the same as described in Section 2.3 (Eq. 5). The only difference now is that the similarity is measured between the test view and a *generated* view, at an arbitrary viewpoint. Since the appearance of a generated view varies smoothly across viewpoints, the value of the similarity measure (our objective function) is also guaranteed to be smooth in the neighbourhood of the optimum we are seeking. However, no assumption can be made about its convexity, and its complex definition (parameterized on the 6 dimensions of the viewpoint and in-plane transformations) makes the evaluation of its gradient expensive. Fortunately, the initial estimates used as starting points can be assumed to be close approximations of the global optimum. All those conditions motivated the use of a simple hill-climbing algorithm. We iteratively optimize pairs of dimensions at a time, namely the 2 viewpoint angles, the image location, then the scale and in-plane rotation. We empirically observed that a close approximation of the global optimum can be reached in this way after only a few iterations [2].

4. Implementation

This section presents details that are not specific to the method, but rather choices of implementation. Those specific choices discussed below refer to the implementation used throughout the evaluation of Section 5 and available on the author’s website [43].

4.1. Application to different types of image features

We demonstrate the applicability of the method to two different types of image features: edges and intensity gradients extracted at a coarse scale.

Edge points

Image edges are widely used in the context of object recognition as they are effective and efficient representatives of shape (being rather sparse, compared to dense gradients). Using edges alone will also allow a fair comparison of our results with existing methods. We use the classical intensity-based Canny detector to extract edges from input images. The image features considered are then the pixels belonging to the resulting binary edge map. We attach, to each of these *edge point* features, an appearance attribute corresponding to the local (tangent) orientation of the edge, defined on $S_1^+ = [0, \pi[$. The kernel associated with that attribute naturally uses a von Mises distribution (similar to a wrapped Normal distribution) on the half-circle (Table 1).

Note that our distance measure between edges could be compared to the directional chamfer distance [42, 44]. The approximation proposed in earlier work (discretization of orientations, approximation of edges by straight segments [42], etc.) can thus be seen as approximations of our more general formulation. Consequently and unsurprisingly, the directional chamfer distance was

Edge points		
$\mathcal{A}^{\text{edges}}$	$= \mathbb{R}^2 \times S_1^+$	position, orientation
$\mathcal{K}^{\text{edges}}(x_1.\text{app}, x_2.\text{app})$	$= \text{VM}^+(x_1.\text{app}; x_2.\text{app}, \kappa)$	
	$= C_1 \cdot e^{\kappa \cos(x_1.\text{app} - x_2.\text{app})}$	
Gradient points		
$\mathcal{A}^{\text{gradients}}$	$= \mathbb{R}^2 \times S_1$	position, orientation
$\mathcal{K}^{\text{gradients}}(x_1.\text{app}, x_2.\text{app})$	$\stackrel{1}{=} C_2 \cdot \text{VM}^+(x_1.\text{app}; x_2.\text{app}, \kappa)$	undirected
	$\stackrel{2}{=} C_3 \cdot \text{VM}(x_1.\text{app}; x_2.\text{app}, \kappa)$	directed

Table 1: Formal definition each type of image features used in our implementation. The notations VM and VM^+ denote von Mises distributions respectively on $S_1 = [0, 2\pi[$ and $S_1^+ = [0, \pi[$. C denotes a normalization constant.

reported to perform similarly as our base method on the ETHZ shape dataset with hand-drawn examples [42]. This comparison is anecdotal since their exact performance numbers were not available. Moreover, our method includes numerous other improvements like the weights on the features or the learning of models from examples.

Gradient points

The goal of our *gradient* features is to represent regions in the image of slowly varying intensity, due to e.g. shading on smooth surfaces. It is easy to see how this information is complementary to the edges, which rather capture sharp transitions. We extract gradients by first convolving the image with derivative-of-Gaussian filters in horizontal and vertical orientations. Each pixel of the image with significant gradient magnitude (set by a fixed low threshold) is an image feature, which gets, as its appearance attributes, the orientation of the gradient (an angle in $[0, 2\pi[$). The extraction of gradients is performed at several coarse scales (typically, $\sigma = 2 \dots 5$ px), and the gradient of largest magnitude is retained. We propose two versions of a kernel suited to the gradient points (Table 1), using the orientation in either an undirected or directed manner. In the *undirected* manner, the orientation of the gradients is compared only on the half-circle. Two horizontal gradients, from black to white and from white to black would thus be considered identical. In the *directed* manner, their orientation in that case is considered opposite. We compare both versions in our experiments.

4.2. Voting algorithm for object detection

As presented in Section 2.3, performing object detection amounts to identifying maxima of the similarity between the test view and one of the training view. We perform the initial detection using one single type of image features at a time. In practice indeed, in the problem of localization in an image, the meaningful optima of the full similarity function (using several types of image features, Eq. 5) will also correspond to local optima for each type features alone. For efficiency, we typically run this procedure using the (more sparse) *edge points*, and then compute the exact similarity scores with (possibly) additional features (Eq. 5), at those discrete values of (v, w) proposed by the voting algorithm. It is however also possible to use dense features alone (gradients for example) with this voting procedure, e.g. if the object does exhibit any meaningful edges, as demonstrated in Section 5.6.

From the definition of our similarity measure we show below that a procedure akin to a traditional Hough voting can approximate this value, which leads to Algorithm 1. On the one hand, considering

a single type of features f , Eq. 4 and 5 specify how to measure the similarity between the test view and a training view v under the in-plane transformations w :

$$\text{similarity}_{\text{test,train}_v}(w) \approx \frac{1}{LL'} \sum_i^L \sum_j^{L'} \text{wt}(x_j) \mathcal{N}(x_i.\text{pos}; x'_j.\text{pos}, \sigma_{\text{pos}}) \mathcal{K}^f(x_i.\text{app}; x_j.\text{app}). \quad (10)$$

with samples x_i drawn from $\phi_{\text{test}^f}^f$, x_j from $\phi_{\text{train}_v^f}^f$, and $x'_j = \text{transform}_w(x_j)$. Let us consider a common 2D voting space \mathcal{H} corresponding to image locations, containing discrete votes at locations $v_j.\text{pos}$ of respective weights $v_j.\text{weight}$. After convolving this voting space with an isotropic Gaussian kernel of bandwidth σ_{pos} , the value at a location l is given by:

$$\mathcal{H}(l) = \sum_j v_j.\text{weight} \cdot \mathcal{N}(l; v_j.\text{pos}, \sigma_{\text{pos}}). \quad (11)$$

One can now readily see that Eq. 10 and 11 can be made equivalent with votes in the Hough space such that $v_j.\text{pos} = (x'_j.\text{pos} - x_i.\text{pos})$ and $v_j.\text{weight} = \text{wt}(x_j) \cdot \mathcal{K}^f(x_i.\text{app}, x_j.\text{app})$. Thus, by casting votes of such locations and weights, the values in the voting space after blurring will approximate our similarity measure for all the discrete image locations represented by the voting space, from which we can then trivially identify the local maxima. The complete algorithm is given in Algorithm 1. It iterates over discrete viewpoints, scales and in-planes rotations³, then uses, at each iteration, the voting procedure to identify the best image location.

4.3. Building and sampling category models

In order to build a model of an object *category* from several instances, we first identify the discrete viewpoints provided in the training data, and at which the category model will be defined. For each viewpoint, we combine all instances defined at that viewpoint, by aligning the views and simply merging their sets of features (see Fig. 8 for example). To align the views, we trivially translate and/or scale each example as it is added to the model, so as to maximize its similarity (Eq. 5) with the current model (Fig. 6, top row).

Using our distributions of features requires drawing samples from those. Sampling from distributions defined through KDE involves selecting a particle at random, then drawing a sample from its associated kernel. The set of particles that define category models is representative of the distribution of image features among the training examples, which is highly multimodal. If those examples are only roughly segmented and contain significant clutter, as in the ‘‘ETHZ Shape’’ dataset (see Fig. 6, top row), a large fraction of the particles will account for noise. They correspond to non-meaningful variations of appearance among the training examples that we wish *not* to capture. To address this specific concern, we propose a variant of the sampling procedure that focuses on the main modes of the distribution. This variant differs in the selection of a particle. Instead of choosing it uniformly at random, we select particles with a probability proportional to their likelihood under the distribution defined by the whole set of features. Formally, given the set of features $\text{train}^f = \{x_i\}_{i=1}^{M_k}$, which define the distribution $\phi_{\text{train}^f}^f$, we will select a particle x_i with a probability proportional to $\phi_{\text{train}^f}^f(x_i)$. Similar procedures for drawing samples from the main modes of a distribution have been previously proposed in the literature, e.g. in [45] under the name

³The discrete steps for the search scales and in-plane rotations are parameters of the algorithm.

of “2-level importance sampling”. As a side note, formulated using importance sampling, the technique proposed above corresponds to using ϕ as the proposal distribution, in order to sample from a distribution ϕ' in which the probability densities would have been squared. Visual comparisons of sampling methods are provided in Fig. 6. Moreover, we empirically observed that, after selecting particles, drawing random samples from their associated kernels proved unnecessary or sometimes detrimental, unless using very large numbers of samples. We thus only use the subset of particles themselves as samples. For efficiency, we preselect this subset off-line as a preprocessing step. Those precomputed samples are thus readily available at test time, and this also allows precomputing their associated weights (Section 2.4). A complete overview of the different steps involved in the learning of a category model, then in its use for detection and pose estimation, is provided in Algorithm 2.

Algorithm 2 Full algorithm for learning model of object category, and for detection followed by continuous pose estimation in a test image.

Training (off-line)

For each viewpoint

- Extract edge and gradient features from training images of the current viewpoint
- Align features of training images, as to maximize their similarity (Eq. 5)
- Merge aligned features of all those training images
- Pre-draw samples from resulting distribution, assign uniform weights

Extract edge and gradient features from validation images

Pre-draw samples from resulting distributions, assign uniform weights

For each iteration for learning weights

- Perform detection on validation images (Algorithm 1)
- Update weights using incorrect detections as negative examples (Eq. 7)

If training viewpoints are close enough for continuous pose estimation

- Detect deformations between neighbouring viewpoints with optical flow
 - Store deformation of each pre-drawn sample from the training images
-

Testing (on-line)

Extract edge and gradient features from test image

Draw samples from resulting distribution, assign uniform weights

Perform detection, using edges only (Algorithm 1)

Compute full similarity scores of resulting detections, using edges and gradients (Eq. 5)

Return detections with highest scores

If training viewpoints are close enough for continuous pose estimation

Consider the detection with the highest score

For each iteration for optimizing the viewpoint

- Generate appearance of the model at a slightly perturbed viewpoint (Eq. 8)
- Compute similarity score between test image of generated viewpoint (Eq. 5)

If similarity score improved **then** keep perturbed viewpoint

Return the detection with the optimized viewpoint

4.4. Software implementation

The manipulation of our low-level image features typically involves large numbers of very simple operations. These are excellent candidates for massively-parallel execution on a graphical processing

unit (GPU). The provided software is implemented in Matlab and allows execution on either a CPU or a GPU. As a ballpark figure, on a typical consumer-level desktop computer, execution on a GPU is typically 20 times faster than execution on a CPU, with test times in the order of seconds (e.g. on the “ETHZ shape” dataset) to minutes (on the cars of the “3D Object” dataset). Although some specific effort was spent adapting the algorithm for execution on a GPU, performance has not been our primary concern, and further improvements in performance are certainly possible. Existing work on the implementation of the Hough transform on GPUs [46–48] may be of interest in this context. Also note that the test times of the algorithm scale with the number of image features used. Our fine-grained, undistinctive image features (edge points and gradient points) thus present the worst case in this regards. Using sparser image features with richer descriptors within the proposed method would hugely decrease its computational requirements.

5. Experimental evaluation

All the contributions of this paper form together a single coherent framework. One of our goals is to demonstrate the versatility of the resulting method, which we therefore evaluate on a variety of tasks and datasets. We present them by order of relative complexity, starting with object detection, first learned from a clean shape template, then learned from images. We then consider the task of *coarse*, discrete pose estimation (or pose *classification*), i.e. the recognition of specific trained viewpoints. We finally consider *continuous* pose estimation. The task of pose estimation is viewed as the most complex task, as it does also involve the detection and recognition of the object within clutter the image. To the extent possible, we reuse existing datasets, such as the “ETHZ shape” [49] and “3D Object” [27], considered as benchmark datasets. This allows direct comparison with recent and state-of-the-art methods on several of the tasks considered. Additionally, we present some of the unique capabilities of our method with a custom dataset of smooth and non-textured objects that can only be recognized from shading and homogeneous image regions, which we make possible through the use of coarse-scale image gradients as image features. All scripts for replicating the experiments of this paper are available, together with the code of the method, on the author’s website [43]. Very few parameters need to be set within the method. A suitable bandwidth for the kernels (Eq. 1) is set as a fraction of the size of the object in the training images (for example, in the order of $\sigma_{pos} = 10\text{px}$ for the ETHZ shape dataset), and the bandwidth on the orientation of edges and gradients is set with $\kappa = 128$ (in a von Mises distribution, which would correspond to a standard deviation of $\sim 20^\circ$ in a wrapped normal distribution). The effect of the other parameters is discussed below, notably the number of samples drawn from the distributions. We identify overlapping detections from the Hough voting algorithm as per a standard procedure, i.e. when their bounding box overlap exceeds 20%, then keep only the one of higher score. One practical effect is that, if two trained viewpoints are matched on a similar location in the test image, only the one with the highest similarity score is retained.

5.1. ETHZ Shape dataset: benchmark for shape detection, trained from a single or multiple examples

The *ETHZ Shape* dataset is a standard benchmark for object detection, which features five diverse classes (bottles, swans, mugs, giraffes and apple logos) in a total of 255 images collected from the web by Ferrari *et al.* [49]. It is considered very challenging because of intraclass shape variations, large scale variability and severe clutter. The goal of evaluating this dataset is to demonstrate that the proposed method achieves adequate performance of shape-based detection. Although we do achieve performance on this task on par with or superior to previously-proposed methods, our method was *not* specifically aimed at this task, and its many other capabilities will be demonstrated

					
Full proposed method (learned weights)	84.1/84.1	96.4/96.4	74.7/73.0	69.7/62.1	90.9/81.8
No weights	81.8/79.5	96.4/90.9	52.7/44.0	54.5/45.5	78.8/66.7
Contour networks, Ferrari, ECCV 2006 [49]	72.7/56.8	90.9/89.1	68.1/62.6	81.8/68.2	93.9/75.8
TPS-RPM, Ferrari, CVPR 2007 [50]	86.4/84.1	92.7/90.9	70.3/65.9	83.4/80.3	93.9/90.9
Ravishankar, ECCV 2008 <i>et al.</i> [51]	97.7/95.5	92.7/90.9	93.4/91.2	95.3/93.7	96.9/93.9

Table 2: ETHZ Shape dataset: detection with hand-drawn models. Weights on image features are represented on the first line; darker colors correspond to heavier weights. We report detection rates (in %) at 0.4/0.3 FPPI. We obtain performance in the order of state-of-the-art methods specifically designed for contour matching. We perform relatively poorly with giraffes and mugs though, which present more variety in aspect ratio in the test images.

					
Full proposed method (proposed sampling, learned weights)	90.0/85.0	96.4/96.4	63.8/55.3	61.3/61.3	52.9/47.1
Proposed sampling, no weights	80.0/70.0	96.4/96.4	38.3/36.2	58.1/41.9	35.3/35.3
Random sampling, learned weights	25.0/25.0	53.6/53.6	12.8/14.9	6.5/ 9.7	23.5/23.5
Random sampling, no weights	20.0/20.0	75.0/71.4	17.0/12.8	29.0/22.6	23.5/23.5
HOG, Dalal, CVPR 2005 [7, 52]	85.0/ -	14.3/ -	34.0/ -	77.4/ -	67.7/ -
TPS-RPM*, Ferrari, CVPR 2007 [50]	83.2/77.7	81.6/79.9	44.5/40.0	80.0/75.1	70.5/63.2
kAS, Ferrari, PAMI 2008 [52]	60.0/50.0	92.9/92.9	51.1/49.0	77.4/67.8	52.4/47.1
M ² HT, Maji, CVPR 2009 [12]	95.0/95.0	96.4/92.9	89.6/89.6	96.7/93.6	88.2/88.2

Table 3: ETHZ Shape dataset: detection with models learned from images. The first line shows the training data, as all the training examples aligned and superimposed onto each other. We report detection rates (in %) at 0.4/0.3 FPPI. We obtain excellent performance on apple logos and bottles, but perform relatively poorly on the giraffes and the swans, for which the example images include lots of clutter. We do not reach the state-of-the-art performance of M²HT, which includes an additional SVM-based classifier to validate candidate detections. **The results of TPS-RPM are not directly comparable as they use a 5-fold cross validation.*

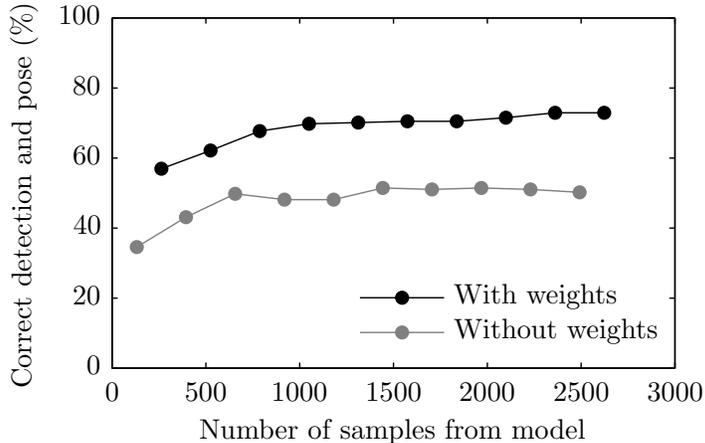


Figure 5: Influence of the number of samples used from the object model. We report the ratio of correct results (correct bounding box *and* correct estimated pose) on the 6th car of the “3D Object” dataset. Performance degrades smoothly with smaller numbers of samples, which can be desirable for efficiency.

on other experiments presented below. The object classes of the ETHZ dataset are intrinsically defined by their shape, and we therefore focus on the use of image edges, as most competing methods do. We did not obtain significant differences in the results with other image features such as our coarse-scale gradients. We consider each object class separately, with a model (with a single viewpoint) trained for each of them independently. The common evaluation measure for this dataset is to plot detection rates (DR) versus the incidence of false positives (false positives per image, FPPI), while varying the detection threshold. Detection rates at a fixed FPPI of 0.3 are used for direct comparisons. Detections are counted as correct with a bounding box overlap of at least 20% with hand-drawn models, and 50% with models learned from images (again, as in existing work such as [52]). All parameters were kept identical for all object classes, except σ_{pos} , set from the size of the training template, as stated above.

The first setting we consider is the use of a single, hand-drawn model of each shape for training, as in [49]. The hand-drawn model is treated directly as an edge map, from which we pre-draw samples by selecting points along these edges, and of which we then learn weights. To allow a valid comparison with [18, 49], we use all 255 images as test set, and learn weights using incorrect detections (negative examples) in 20 random images collected from the web. We obtain the weights represented in Table 2. One can observe that long, uncharacteristic and easily matchable parts of the contours receive low weights, while high weights are assigned to salient parts with higher curvature, naturally less frequent among the random negative examples used to learn these weights. As expected, the detection results show that those weights significantly improve the results by decreasing the number of false positives (Table 2). While not surpassing the state of the art, we obtain remarkable performance, especially considering the fact that competing methods were specifically designed for the particular task of shape matching of contours, whereas our approach is a much more general one.

The second setting in which we evaluate this dataset involves learning the models from example images. We use the training and test splits of [52], i.e. the first half of the images of each class as

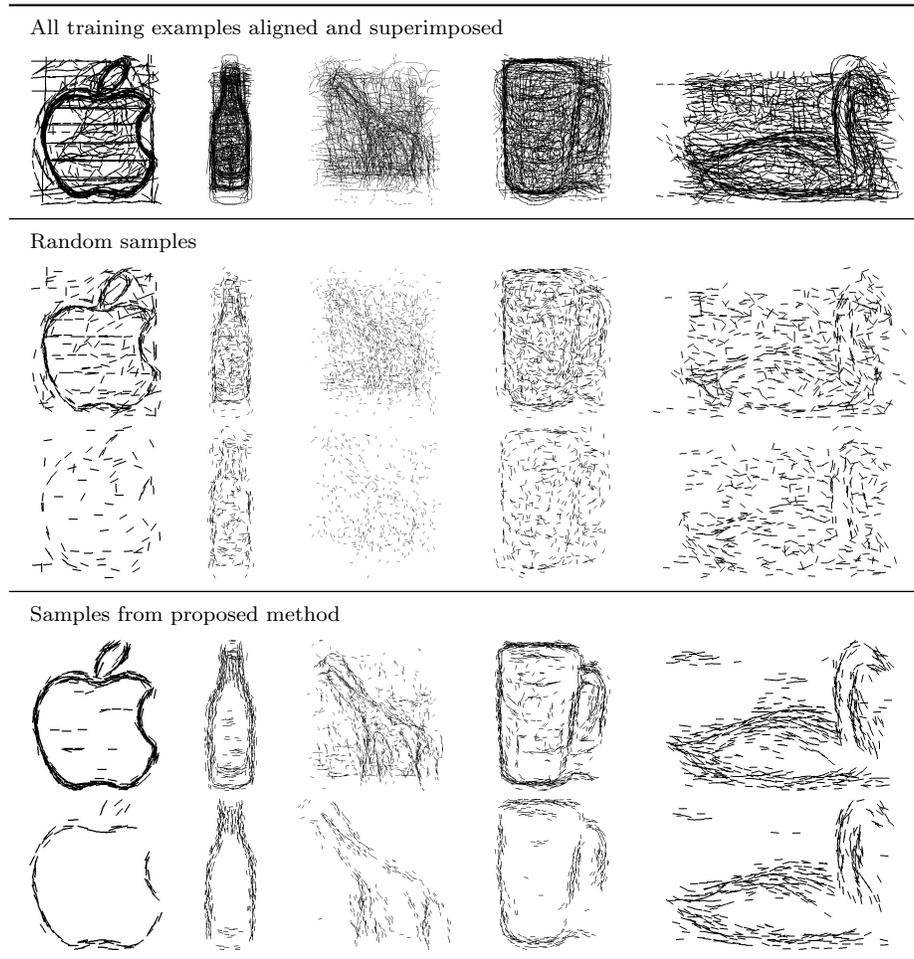


Figure 6: ETHZ Shape dataset: models learned from images, visualized as samples drawn from the distributions of features. We visualize two different amounts of samples for each sampling method (with equal amounts for the two methods). The proposed sampling scheme is able to recover very simple representations of the shapes with small number of samples, whereas a basic, random sampling includes unwanted samples corresponding to clutter in the training images. The model of the giraffe is noticeably worse than the other shapes, because of the large fraction of clutter in most of the training examples.

	Correct detections (AP)	Correct poses (MPPE)
Full proposed method (edges and directed gradients, learned weights)	92.5%	91.0%
Edges only, no weights	54.1%	85.2%
Edges only, learned weights	90.4%	92.6%
Arie, ICCV 2009 [53]	–	48.5%
Su, ICCV 2009 [54]	55.3%	67.0%
Liebelt, CVPR 2010 [55]	76.7%	70.0%
Payet, ICCV 2011 [56]	–	85.4%
Xiang, CVPR 2012 [57]	98.4%	93.4%

Table 4: 3D Object car dataset: detection and discrete pose estimation. The use of weights substantially improves the detection among clutter. We outperform most existing methods, although we do not reach the near-perfect performance of the “Aspect Layout Model” of Xiang *et al.* [57], which explicitly considers the 3D structure of the objects.

the training set. We also use the rough presegmentation of these images provided as ground truth bounding boxes. Those images are aligned and set at a same scale (Section 4.3). We pre-draw samples from the model, of which we learn weights, using, as negative training images, images from the four other classes (as in [52]). The testing is performed on all other images of all classes. The models learned for each class are visualized in Fig. 6. The effect of the proposed sampling method (Section 4.3) versus a random sampling is quite dramatic. The proposed procedure concentrates on the main modes of the distributions, and provides reliable representations of the shape, even with limited numbers of samples. These “cleaner” models hide some undesirable variation from the training data, such as the water waves around the swans, or the inner texture within the apple logos.

We outperform a number of existing methods (Table 3). We do not reach the near-perfect results of M²HT [12], which uses a discriminative classifier on top of their detections. Interestingly however, their detection algorithm alone achieved a rather low detection rate of only 60.9% at 1.0 FPPI, whereas our detector achieves 72.9% at 0.4 FPPI (averaged over the five classes). They also reported a notable improvement by performing detection at different aspect ratios, which we do not.

5.2. 3D Object dataset: multiview model, detection in clutter and coarse pose estimation

We now consider the “3D Object” dataset introduced by Savarese *et al.* [27]. We focus on the “car” object, as it is the most widely used, and gives us the most points of comparison with existing methods. The dataset features 10 different cars, each viewed under 24 viewpoints (8 azimuths and 3 elevations) and 3 scales. The task is both to detect the car among background clutter and to identify its azimuth angle (one of the 8 discrete values, i.e. whether it is view from the front, the left side, the 3/4 front/right side, etc). Pose estimation is limited to this coarse classification into the trained viewpoints, as these are too distant from each other to use our procedure for continuous pose estimation; finding dense correspondences between views of such complex objects would require viewpoints much closer than 45° apart.

We use similar conditions and evaluation criteria as [27]: the first 5 cars for training and the last 5 for testing. The training images are used both to build the model (with the provided ground truth segmentation), and then to learn weights by using the incorrect detections on them as negative examples (Section 2.4). Results are measured in terms of the rate of correct detections (average

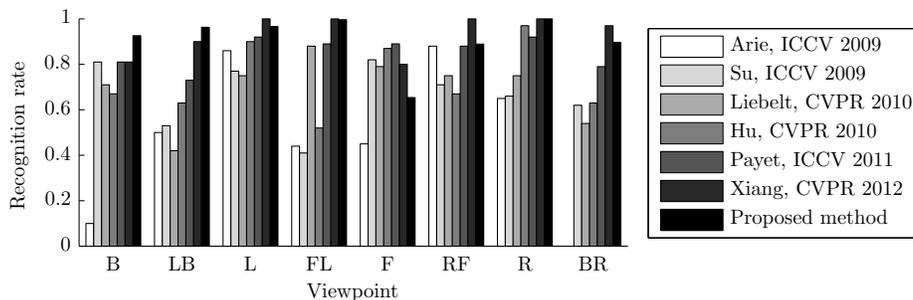


Figure 7: 3D Object Car dataset: classification rate of the different viewpoints. We clearly outperform most existing methods.



Figure 8: Visualization of the edge model for three views of the “3D Object Car” dataset. Darker colors correspond to heavier weights. Low weights are assigned to parts that can easily be matched to common background clutter (and lead to false positive detections), such as horizontal lines. More characteristic parts, such as the wheels in the side view, receive, on the opposite, high weights.

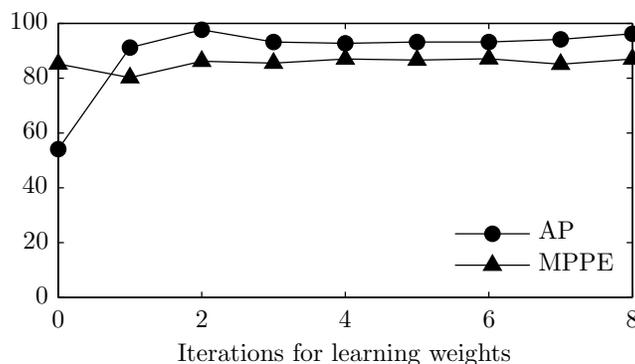


Figure 9: 3D Object Car dataset: evolution of performance for detection (AP) and pose estimation (MPPE) as a function of the number of iterations for learning the weights of the samples drawn from the distributions of features. At each iteration, weights are updated based on negative examples provided as incorrect detections in the training images themselves, then used in the manner of a validation dataset. Stable weights are reached with a small number of iterations.

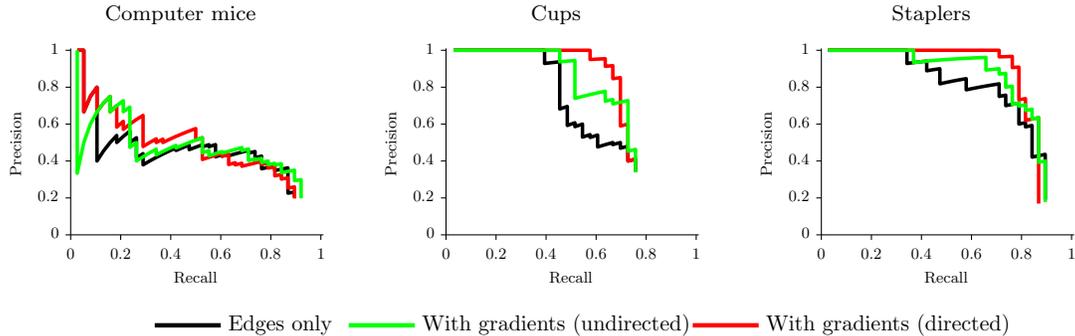


Figure 10: Results of detection on the tabletop dataset as precision/recall curves for each of the three object categories. The detection rate is significantly improved by using coarse-scale gradients in addition to edges, especially for the mugs, which present characteristic shading patterns captured by those additional features.

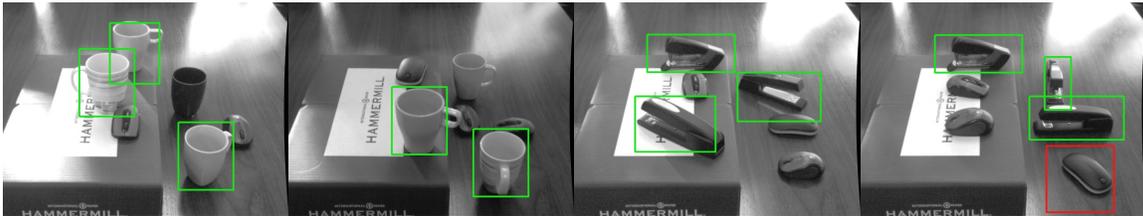


Figure 11: Sample detections of cups (left, center-left) and staplers (center-right, right) on the tabletop dataset (correct detections in green, incorrect ones in red).

precision, or AP), defined by a bounding box overlap of 50%, and the ratio, among correct detections, of correct estimates of the azimuth angle (mean precision in pose estimation, or MPPE). As reported in Fig. 7 and Table 4, we outperform most existing methods evaluated on this dataset. The visualization of the weights learned for the image features (Fig. 8) provides some insight on their significant impact on performance. In the side view for example, the long horizontal lines, which are also frequent in background clutter, receive low weights. The wheels, on the opposite, are more characteristic and much better indicators of a car seen from the side, and thus receive higher weights. Interestingly, this distribution of weights is visually very similar to those obtained by Maji and Malik [12] with their own procedure, on side views of cars of the “UIUC car” dataset. We also observe that using our coarse-scale gradients as features, in addition to edges, brings a slight improvement. The difference is however marginal, as the appearance of the cars is already well defined by their shape and edges alone.

5.3. Tabletop dataset: multiview model, detection in clutter

We further evaluate our performance for object detection in clutter using the “tabletop” dataset of Sun *et al.* [58]. It features a total of 30 objects from 3 categories: computer mice, mugs and staplers. These object categories present more basic shapes than the cars in the “3D Object” dataset, which is a different challenge and provides complementary evaluation points. We use,

Number of training views	15	30	40
Full proposed method (optimized viewpoint)	8.15°	1.16°	0.80°
Nearest neighbour detection only	8.63°	3.89°	3.00°
Torki and Elgammal [30]	5.47°	1.93°	1.84°
Teney and Piater, CRV 2013 [1]	4.42°	1.62°	1.49°

Table 5: Rotating car dataset: continuous pose estimation on a single instance (the first car). We report the mean error on the estimated azimuth angle, in degrees. We outperform existing methods on the two largest sizes of the training set; with smallest training set however, the viewpoints are often too distant to each other to reliably interpolate the appearance at intermediate viewpoints, and the optimization of the viewpoint is thus not as effective.

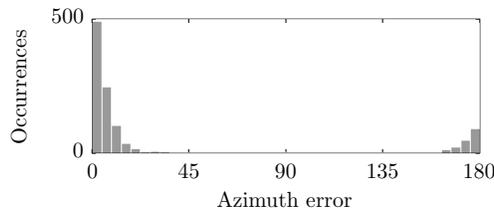


Figure 12: Rotating car dataset: distribution of error on estimated azimuth (in degrees) during experiments on multiple cars; a number of images yield an error of about 180°, due to ambiguous appearance of side views and front/rear views.

as the training set, the part of the dataset with objects appearing on a turntable under known viewpoints (“Table-Top-Pose”; see Fig. 1). A model is learned for each object category. Testing is performed on scenes (“Table-Top-Local”) containing one or several instances of the objects in a cluttered office environment; note that those experimental conditions are more challenging than existing evaluations (e.g. in [56]) since those two parts of the dataset feature different imaging and lighting conditions. We perform detection in the test images of each object category separately, and we measure the detection rates with the standard criterion of 50% bounding box overlap. We report results in Fig. 10 as precision/recall curves. The use of coarse-scale gradients brings here a significant improvement, in particular on cups, the shape of which produces very characteristic shading patterns. The improvement is marginal for the computer mice: the different instances are very diverse in shape, and observed under fixed lighting conditions in the training images that produce specular highlights, which do not appear in the testing images. The simple gradients are obviously not robust to such variations by themselves, but we believe that they would show a better advantage if the training images presented more varied lighting conditions, although this could unfortunately not be tested with this dataset.

5.4. EPFL Rotating cars dataset: continuous pose estimation

We now evaluate the unique capability to perform *continuous* pose estimation within our appearance-based method. Few other methods have tackled this problem, especially at the level of object *categories*, which explains the limited choice of suitable datasets. The most appropriate, in our view, is the “Multiview car” dataset introduced by Ozuysal *et al.* [59]. It includes about 2000 images of 20 very different cars filmed on rotating stands at a motor show. The dataset is very challenging



	Median	Mean 90%ile	Mean	Error < 22.5°	Error < 45°
Full proposed method (optimized viewpoint, learned weights)	5.2°	18.7°	34.7°	80.3%	82.1%
Nearest neighbour detection only, no weights	7.6°	24.7°	39.8°	71.6%	76.3%
Nearest neighbour detection only, learned weights	5.7°	19.1°	35.0°	80.2%	82.1%
Ozuysal <i>et al.</i> [59]	–	–	46.5°	41.7%	71.2%
Glasner <i>et al.</i> [60]	24.83°	–	–	–	–
Torki and Elgammal [30]	11.3°	19.4°	34.0°	70.3%	80.7%
Teney and Piater, CRV 2013 [1]	5.8°	23.7°	39.0°	78.1%	79.7%

Table 6: Rotating car dataset: continuous pose estimation at the category level. Instances 1–10 are used for training (first row of pictures) and 11–20 for testing (second row of pictures). We outperform existing methods. Note however that the precision of the best methods reaches the accuracy and level of imprecision (estimated around 3 – 4°) in the ground truth annotations, which explains why no further improvements can be made, especially by our optimization of the viewpoint.

	Detection rate (AP)	Azimuth error < 10°	Mean azimuth error
Full proposed method (optimized viewpoint)	83.3% (40/48)	70.0%	3.8°
Nearest neighbour detection only	83.3% (40/48)	70.0%	4.0°
Zia <i>et al.</i> [61] (with hand-made CAD models)	93.8%(45/48)	73.3%	3.8°

Table 7: Detection and continuous pose estimation, using the model learned from rotating cars (instances 1–10, as in Table 6), tested on images from the “3D Object” dataset (instance 6). The model is able to detect and estimate the orientation of the car accurately, despite challenging differences in imaging conditions, in scale and in object appearance between the two datasets. We use the same metrics as [61]: the rate of correct azimuths is measured on correct detections, and the mean error is measured on those correct azimuths. Incorrect azimuths often are off by about 180°. The results of Zia *et al.* [61] are included for reference only: they rely on hand-built CAD models, whereas our method is purely appearance-based and trained on example images.

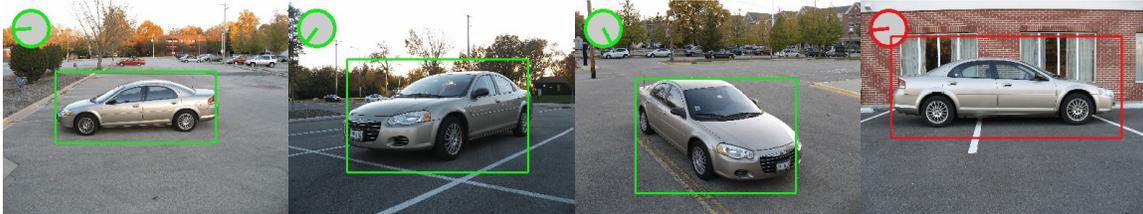


Figure 13: Samples results of continuous pose estimation on the “3D Object” dataset using the model learned from rotating cars. Boxes indicate the localization of the car as identified by our system, and the roses in the upper-left corners indicate the orientation of the front of the car as seen from the top (as in [59]).

due to changing lighting conditions, high intraclass variability in shape, appearance and texture, and highly similar views (symmetrical side views, similar front and rear views) which are sometimes hard to differentiate even for a human. The dataset was used in [59] for pose classification in 16 discrete bins, and in [30] for continuous pose estimation. We first evaluate our method, as in [30], on the first car of the dataset, training a model on this single specific car. We select 15, 30 or 40 equally-spaced images of the sequence as training images, and use all other images (spaced about 4° apart) for testing. We obtain superior results to [30] (Table 5). We then perform experiments at the *category* level, in conditions similar to those used in [30]. The first 10 cars of the dataset are used for training, and the other 10 for testing. Again, we obtain performance superior to all published results to our knowledge (Table 6). As highlighted in Fig. 12, the remaining errors in pose estimation correspond to an error of about 180° . This is caused by the symmetric aspects of some cars in the side views, and confusion between front- and rear-facing views.

We further evaluated the generalization capabilities of the model learned from this dataset. We thus use this model, trained from the 10 first rotating cars, for testing on the “3D Object” dataset (see Section 5.2 above). This is a challenging task, as those two datasets present very different conditions in terms of imaging conditions, scale, background clutter, etc. We do the testing specifically on the sixth car of the dataset, the exact pose of which was annotated by Zia *et al.* [61] by manually fitting 3D models to the images. These annotations are used as ground truth to measure the accuracy of the azimuth angle estimated by our method for continuous pose estimation. We obtain excellent results (Fig. 13), close to the accuracy obtained by the complex method of Zia *et al.* [61], which basically aligns 3D CAD models of cars with the images, compared to our more general appearance-based procedure.

5.5. Volvo car: continuous 3D pose estimation and synthesis of novel views

We further evaluate our method for continuous pose estimation, this time with a model spanning both dimensions of the viewing sphere around the object, as opposed to the single degree of freedom (azimuth angle) of the rotating cars presented above. The choice of datasets for this task that allow comparison with existing methods is limited, here again. We use the “3D pose Volvo car” of Viksten *et al.* [62, 63] (Fig. 14). This allows a comparison with a classical method [62] that uses discriminative image descriptors with a voting and averaging scheme, which is the classical approach for robust 3D pose estimation (with the disadvantage of being limited to specific object instances). The dataset features a toy car viewed under regular increments of azimuth and elevation angles. We consider two training/test splits: a small and a large training set, with views spaced respectively 20° and 10° apart (on both azimuth and elevation angles), and exactly one test view

Spacing between training views		20°	10°
Full proposed method (optimized viewpoint)	Azimuth	27.22°(1.67°)	0.84° (1.11°)
	Elevation	2.65° (1.11°)	0.86° (0.56°)
Nearest neighbour detection only	Azimuth	35.56°(10.00°)	5.00°(5.00°)
	Elevation	10.00°(10.00°)	5.14°(5.00°)
Johansson <i>et al.</i> [62]	Azimuth	4.21°	1.25°
	Elevation	2.66°	1.06°

Table 8: Continuous pose estimation on the Volvo car. We report the mean (median) error of azimuth/elevation angles, in degrees. The large mean error in azimuth comes from a single misclassified test image, as attested by the small median error. We clearly outperform the classical method of Johansson *et al.* based on discriminative feature descriptors and an averaging scheme in pose space.



Figure 14: Training images of the Volvo car with views spaced 20° apart.

between each pair of training view, i.e. as a grid on the viewing sphere (as in [62]). In both cases, we obtain results significantly superior to [62] in terms of accuracy (Table 8). The smaller training set is more challenging for detecting deformations between views, and seemed to reach the limits of the optical flow algorithm we use to detect deformations between neighbouring views. The dataset also allows a good visualization of the capabilities of our generative model, by varying continuously the viewpoint around the object. The effect, unfortunately hard to convey in static images (Fig. 15), is a vivid impression of manipulating a 3D model of the object – although there is no underlying explicit representation of the 3D shape. Videos and an interactive viewing tool are available on the author’s website [43].

5.6. Non-textured objects

We finally demonstrate the interest of using coarse-scale gradients with a new dataset featuring non-textured objects. These toy objects, made of plastic, feature basic shapes with few internal edges (Fig. 16). This lack of distinctive visual characteristics actually makes them difficult to identify among clutter, and the absence of texture renders the estimation of their pose problematic. For example, considering the knife, one cannot differentiate the (round) handle from the (flat) blade, observing edges and silhouette alone. We made this new dataset available on the author’s website [43]. It comprises examples images of each object with segmentation and pose annotations (used for training), plus a series of test images of cluttered scenes feature these objects, also with ground truth segmentations and annotations (used for evaluation). Results of detection are counted as correct when the overlap of bounding boxes with the ground truth exceeds 50% *and* the estimated pose is

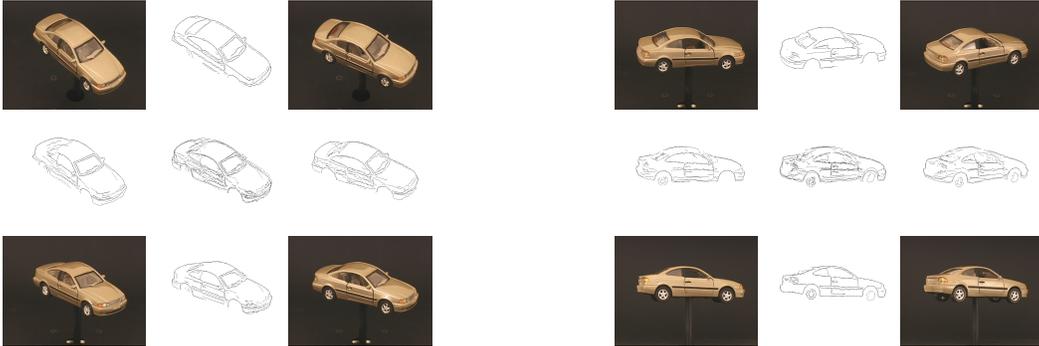


Figure 15: Demonstration of our generative model, with training views spaced 20° apart in azimuth and elevation angles, and the edge-based appearance of the object generated at intermediate, unseen viewpoints.

correct (error on viewpoint angles smaller than 20°). Unsurprisingly, most objects are detected very poorly using edges alone (with our method, though any other contour-based recognition method is expected to work as poorly). Our full method however, using coarse-scale gradients in the measure of similarity between the detections and the training examples, is able to differentiate between similar-looking poses, and achieves far superior detection rates (Fig. 16). Most remaining incorrect detections are due to clutter and confusion from the similar appearance of these simple objects. We also tested the detection of those objects using gradient features alones, without edges. This did not prove effective in practice, since their appearance, defined by these gradients, is very simple and easily confused with the background or other objects. The knife for example, just corresponds to a region without gradients (the flat blade) and a part with gradients oriented orthogonally to the knife’s length (the round handle). Such a description is complementary to the silhouette represented by edges, but is not informative enough by itself to localize such the object among clutter.

6. Discussion and conclusions

We introduced a representation of 2D appearance as distributions of low-level, fine-grained image features. We used this representation to build multiview models of object categories. Those models encode the appearance of objects at a number of discrete viewpoints, and, in addition, how these viewpoints deform into one another as the viewpoint continuously varies. Those deformations between neighbouring viewpoints are detected with an optical flow algorithm, and encoded as translations of individual image features with respect to viewpoint changes. We provide a way to measure the similarity between an arbitrary test image and an object model at a specific viewpoint. We use this measure of similarity to perform a number of tasks: detection and localization in cluttered images (identifying the local maxima of the similarity measure with respect to locations in the test image), discrete pose estimation (identifying the learned viewpoint with the highest similarity measure with the test image) and continuous pose estimation (identifying the maxima of the similarity measure as the viewpoint *continuously* varies). In contrast with common practice, we address and evaluate a number of related tasks with a single approach. This is reflected in our experimental evaluation, which includes extensive testing on a number of very different benchmark datasets, which are seldom considered together. We demonstrate performance on the “ETHZ Shape” dataset for

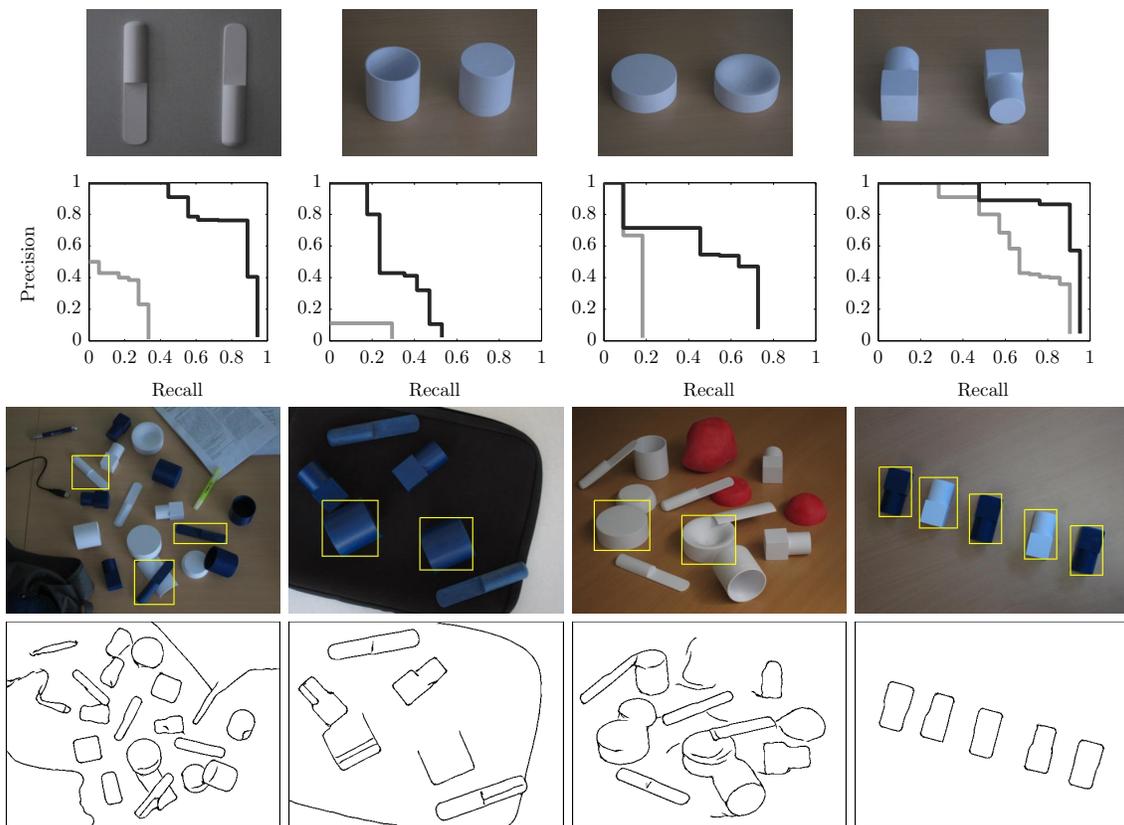


Figure 16: New dataset featuring non-textured objects with few distinctive characteristics. Sample training images (first row) of the objects, the “knife” (round handle, flat blade), the “cup”, the “ashtray” (both hollow on one side only) and the “peg” (round on one side, square on the other). Results of detection and pose estimation on a total of 28 scenes are reported as interpolated precision/recall curves, using edges only (gray) and in conjunction with coarse-scale gradients (black). Sample scenes (bottom rows, with bounding boxes of highest-scored detections) show that edges provide only ambiguous information to determine the pose of the objects.

shape matching and detection in clutter of categories well above baseline methods, on par with a number of more task-specific methods. We also obtain remarkable performance on the recognition of more complex objects, notably the cars of the “3D Object” dataset, with detection rates of 92.5% and an accuracy in pose estimation of 91%. For the task of continuous pose estimation, we obtain results superior to the state-of-the-art on the “rotating cars” dataset.

The limitations of our appearance model lie mostly in the representation of object categories. The distribution of image features are representative of the occurrence of features among the training examples, but they do not encode the co-occurrence of these features. The resulting model can thus represent all combinations of variations present in the examples. A model learned from images of cars and giraffes would not only represent those two types of objects, but also anything looking partially like a car and partially like a giraffe (i.e. combining visual features from different training examples). This may be seen as a strength, as few examples can suffice to represent wide variations of overall appearance. However, this also means that the overall procedure will practically be most effective with training examples sharing strong visual characteristics, and not with categories defined semantically or including instances looking vastly different. This representation of appearance thus also assumes fairly rigid objects (although we still obtained good performance on shape matching of the ETHZ classes). Complex deformable objects would probably be better handled by part-based models (e.g. [17, 64]). We believe that this limitation was probably masked by the relative simplicity of the objects in the available datasets. Let us note however that the proposed representation as distributions of features could serve as a building block of part-based models.

The importance of shape and structure in the model leads to another limitation, in the context of object recognition in complex scenes. As opposed to, e.g. the classical “bag of visual words” approach, our model does not encode contextual clues of the scene. For example, blue color and clouds in the background of an image may be indicative of the presence of an airplane. Such information is however not encoded within our model, aimed at individual object recognition. This information could be taken into account at another, higher level, dealing for overall scene understanding.

All limitations discussed above lead to potential avenues for further developments. In addition, on the task of continuous pose estimation, one could explore alternative optimization algorithms to use with our generative model. Improvements in efficiency at this level could render the model suitable for continuous pose *tracking*, thereby widening its range of applicability even further. The detections of the deformations between the trained viewpoints, which currently uses a standard algorithm to detect optical flow, could also be improved, be made applicable to more distant viewpoints and to other types of training data, e.g. videos of the object. Finally, one could evaluate other types of image features within the proposed approach. We demonstrated its particular applicability to low-level features, although more traditional, higher-level features could also be used, such as histogram-based descriptors [7, 16] or region features [65].

Funding

The research leading to these results has received funding from the European Community’s Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 270273, Xperience. Damien Teney is supported by a research fellowship of the Belgian National Fund for Scientific Research (FNRS).

Authors

Damien Teney



Damien Teney obtained a Ph.D. in computer science from the University of Liège in 2013, being then supported by a Research Fellowship of the Belgian Fund for Scientific Research (FNRS). His research interest cover the visual recognition of objects, and global scene understanding from 2D images, with the ultimate objective of allowing machines to interact with their environment, e.g. through robotic applications. He is currently a Research Associate at the University of Bath, United Kingdom.

Justus Piater



After graduating with highest honors from the University of Magdeburg, Germany, in 1994, Justus H. Piater was awarded a Fulbright scholarship and earned his M.Sc. and Ph.D. in computer science at the University of Massachusetts Amherst, USA, in 1998 and 2001, respectively. A recipient of a European Marie-Curie Individual Fellowship, he was a postdoctoral researcher at INRIA Rhne-Alpes, France, from 2000 to 2002. He subsequently became a professor of computer science at the University of Lige, Belgium, where he founded and directed the Computer Vision research group. He spent the academic year 2008–09 as a visiting researcher with the group of Prof. Schlkopf at the Max Planck Institute for Biological Cybernetics in Tübingen, Germany. He moved to the University of Innsbruck, Austria, in 2010, where he has formed a research group in Intelligent and Interactive Systems. His research interests include computer vision and machine learning, with a focus on visual learning and closed-loop sensorimotor interaction motivated by robotics, and video analysis. He has published about 120 papers in international journals and conferences.

References

- [1] D. Teney, J. Piater, Continuous Pose Estimation in 2D Images at Instance and Category Levels, in: *Computer and Robot Vision*, 2013.
- [2] D. Teney, J. Piater, Modeling Pose/Appearance Relations for Improved Object Localization and Pose Estimation in 2D images, in: *6th Iberian Conference on Pattern Recognition and Image Analysis*, vol. 7887 of *LNCS*, 59–68, 2013.
- [3] F. Rothganger, S. Lazebnik, C. Schmid, J. Ponce, 3D Object Modeling and Recognition Using Local Affine-Invariant Image Descriptors and Multi-View Spatial Constraints, *International Journal of Computer Vision* 66 (3) (2006) 231–259.

- [4] D. Hoiem, C. Rother, J. M. Winn, 3D LayoutCRF for Multi-View Object Class Recognition and Segmentation, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2007.
- [5] P. Yan, S. M. Khan, M. Shah, 3D Model based Object Class Detection in An Arbitrary View, in: IEEE International Conference on Computer Vision, 2007.
- [6] D. Glasner, M. Galun, S. Alpert, R. Basri, G. Shakhnarovich, Viewpoint-Aware Object Detection and Continuous Pose Estimation, *Image and Vision Computing* 30 (12) (2012) 923–933, ISSN 0262-8856.
- [7] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE International Conference on Computer Vision and Pattern Recognition, ISSN 1063-6919, 886–893, 2005.
- [8] C. H. Lampert, M. B. Blaschko, T. Hofmann, Beyond sliding windows: Object localization by efficient subwindow search, in: In Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, 1–8, 2008.
- [9] D.H., Ballard, Generalizing the Hough transform to detect arbitrary shapes, *Pattern Recognition* 13 (2) (1981) 111 – 122, ISSN 0031-3203.
- [10] B. Leibe, A. Leonardis, B. Schiele, An Implicit Shape Model for Combined Object Categorization and Segmentation, in: Towards Category-Level Object Recognition, 496–510, 2006.
- [11] A. Lehmann, B. Leibe, , L. V. Gool, PRISM: PRincipled Implicit Shape Model, in: British Machine Vision Conference, 2009.
- [12] S. Maji, J. Malik, Object detection using a max-margin Hough transform, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2009.
- [13] S. Ekvall, F. Hoffmann, D. Kragic, Object Recognition and Pose Estimation for Robotic Manipulation using Color Cooccurrence Histograms, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003.
- [14] P. Mittrapiyanuruk, G. N. DeSouza, A. C. Kak, Calculating the 3D-pose of Rigid-objects using Active Appearance Models, in: IEEE International Conference on Robotics and Automation, 2004.
- [15] A. R. Pope, D. G. Lowe, Probabilistic Models of Appearance for 3D Object Recognition, *International Journal of Computer Vision* 40 (2) (2000) 149–167.
- [16] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision* 60 (2) (2004) 91–110.
- [17] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object Detection with Discriminatively Trained Part-Based Models, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32 (9) (2010) 1627–1645, ISSN 0162-8828.
- [18] V. Ferrari, F. Jurie, C. Schmid, From Images to Shape Models for Object Detection, *International Journal of Computer Vision* 87 (3) (2010) 284–303.

- [19] A. Opelt, A. Pinz, A. Zisserman, Learning an Alphabet of Shape and Appearance for Multi-class Object Detection, *International Journal of Computer Vision* 80 (1) (2008) 16–44.
- [20] M. Martinez Torres, A. Collet Romea, S. Srinivasa, MOPED: A Scalable and Low Latency Object Recognition and Pose Estimation System, in: *IEEE International Conference on Robotics and Automation*, 2010.
- [21] F. Viksten, R. Soderberg, K. Nordberg, C. Perwass, Increasing pose estimation performance using multi-cue integration, in: *IEEE International Conference on Robotics and Automation*, 2006.
- [22] H. Schneiderman, T. Kanade, A statistical method for 3D object detection applied to faces and cars, in: *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, ISSN 1063-6919, 746–751, 2000.
- [23] A. Torralba, K. Murphy, W. Freeman, Sharing features: efficient boosting procedures for multiclass object detection, in: *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, ISSN 1063-6919, II-762–II-769, 2004.
- [24] M. Weber, W. Einhauser, M. Welling, P. Perona, Viewpoint-invariant learning and detection of human heads, in: *IEEE International Conference on Automatic Face and Gesture Recognition*, 20–27, 2000.
- [25] D. Teney, J. Piater, Generalized Exemplar-Based Full Pose Estimation from 2D Images without Correspondences, in: *Digital Image Computing: Techniques and Applications*, 2012.
- [26] A. Kushal, C. Schmid, J. Ponce, Flexible Object Models for Category-Level 3D Object Recognition, in: *IEEE International Conference on Computer Vision and Pattern Recognition*, ISSN 1063-6919, 2007.
- [27] S. Savarese, L. Fei-Fei, 3D generic object categorization, localization and pose estimation, in: *IEEE International Conference on Computer Vision*, ISSN 1550-5499, 2007.
- [28] A. Thomas, V. Ferrar, B. Leibe, T. Tuytelaars, B. Schiel, L. Van Gool, Towards Multi-View Object Class Detection, in: *IEEE International Conference on Computer Vision and Pattern Recognition*, 2006.
- [29] D. Baltieri, R. Vezzani, R. Cucchiara, People orientation recognition by mixtures of wrapped distributions on random trees, in: *IEEE European Conference on Computer Vision*, ISBN 978-3-642-33714-7, 270–283, 2012.
- [30] M. Torki, A. M. Elgammal, Regression from local features for viewpoint and pose estimation, in: *IEEE International Conference on Computer Vision*, 2011.
- [31] L. Mei, J. Liu, A. Hero, S. Savarese, Robust object pose estimation via statistical manifold modeling, in: *Computer Vision (ICCV), 2011 IEEE International Conference on*, ISSN 1550-5499, 967–974, 2011.
- [32] S. Avidan, A. Shashua, Novel view synthesis in tensor space, in: *IEEE International Conference on Computer Vision and Pattern Recognition*, 1997.
- [33] S. E. Chen, L. Williams, View interpolation for image synthesis, in: *SIGGRAPH*, 1993.

- [34] S. M. Seitz, C. R. Dyer, Toward Image-Based Scene Representation Using View Morphing, in: International Conference on Pattern Recognition, 84–89, 1996.
- [35] S. Savarese, L. Fei-Fei, View synthesis for recognizing unseen poses of object classes, in: IEEE European Conference on Computer Vision, 2008.
- [36] M. Sun, H. Su, S. Savarese, L. Fei-Fei, A multi-view probabilistic model for 3D object classes, in: IEEE International Conference on Computer Vision and Pattern Recognition, ISSN 1063-6919, 2009.
- [37] R. E. Caflisch, Monte Carlo and quasi-Monte Carlo methods, *Acta Numerica* 7 (1998) 1–49, ISSN 1474-0508.
- [38] A. Frome, Y. Singer, F. Sha, J. Malik, Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification, in: IEEE International Conference on Computer Vision, 2007.
- [39] C. Gu, J. J. Lim, P. Arbelaez, J. Malik, Recognition using regions, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2009.
- [40] P. Yarlagadda, B. Ommer, From Meaningful Contours to Discriminative Object Shape, in: IEEE European Conference on Computer Vision, 2012.
- [41] C. D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, New York, NY, USA, ISBN 0521865719, 9780521865715, 2008.
- [42] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, R. Chellappa, Fast directional chamfer matching, in: IEEE International Conference on Computer Vision and Pattern Recognition, 1696–1703, 2010.
- [43] D. Teney, Personal website, <http://montefiore.ulg.ac.be/~dtaney/cviu.htm>, 2013.
- [44] J. Shotton, A. Blake, R. Cipolla, Multiscale Categorical Object Recognition Using Contour Fragments, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (7) (2008) 1270–1281.
- [45] R. Detry, N. Pugeault, J. Piater, A Probabilistic Framework for 3D Visual Object Representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31 (10) (2009) 1790–1803.
- [46] J. Gómez-Luna, J. M. González-Linares, J. I. Benavides, E. L. Zapata, N. G. Mata, Load Balancing versus Occupancy Maximization on Graphics Processing Units: The Generalized Hough Transform as a Case Study, *IJHPCA* 25 (2) (2011) 205–222.
- [47] N. Jotwani, S. Sah, A Fast and Accurate GHT Implementation on CUDA, in: International Conference on Meta Computing, 2011.
- [48] G.-J. van den Braak, C. Nugteren, B. Mesman, H. Corporaal, Fast Hough Transform on GPUs: Exploration of Algorithm Trade-Offs, in: *ACIVS*, 611–622, 2011.
- [49] V. Ferrari, T. Tuytelaars, L. Van Gool, Object detection by contour segment networks, in: IEEE European Conference on Computer Vision, ISBN 3-540-33836-5, 978-3-540-33836-9, 14–28, 2006.

- [50] V. Ferrari, F. Jurie, C. Schmid, Accurate Object Detection with Deformable Shape Models Learnt from Images, in: IEEE International Conference on Computer Vision and Pattern Recognition, ISSN 1063-6919, 1–8, 2007.
- [51] S. Ravishankar, A. Jain, A. Mittal, Multi-stage contour based detection of deformable objects, in: IEEE European Conference on Computer Vision, 483–496, 2008.
- [52] V. Ferrari, L. Fevrier, F. Jurie, C. Schmid, Groups of Adjacent Contour Segments for Object Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (1) (2008) 36–51.
- [53] M. Arie-Nachimson, R. Basri, Constructing Implicit 3D Shape Models for Pose Estimation, in: IEEE International Conference on Computer Vision, 2009.
- [54] H. Su, M. Sun, L. Fei-Fei, S. Savarese, Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories., in: IEEE International Conference on Computer Vision, 2009.
- [55] J. Liebelt, C. Schmid, Multi-view object class detection with a 3D geometric model, in: IEEE International Conference on Computer Vision and Pattern Recognition, ISSN 1063-6919, 1688–1695, 2010.
- [56] N. Payet, S. Todorovic, From contours to 3D object detection and pose estimation (2011) 983–990.
- [57] Y. Xiang, S. Savarese, Estimating the Aspect Layout of Object Categories, in: IEEE International Conference on Computer Vision and Pattern Recognition, ISBN 978-1-4673-1226-4, 3410–3417, 2012.
- [58] M. Sun, G. Bradski, B.-X. Xu, S. Savarese, Depth-encoded Hough voting for joint object detection and shape recovery, in: IEEE European Conference on Computer Vision, 658–671, 2010.
- [59] M. Ozuysal, V. Lepetit, P. Fua, Pose estimation for category specific multiview object localization, in: IEEE International Conference on Computer Vision and Pattern Recognition, ISSN 1063-6919, 2009.
- [60] D. Glasner, M. Galun, S. Alpert, R. Basri, G. Shakhnarovich, Viewpoint-aware object detection and pose estimation, in: IEEE International Conference on Computer Vision, ISSN 1550-5499, 1275–1282, 2011.
- [61] Z. Zia, M. Stark, K. Schindler, B. Schiele, Revisiting 3D Geometric Models for Accurate Object Shape and Pose, in: IEEE International Workshop on 3D Representation and Recognition, 2011.
- [62] B. Johansson, A. Moe, Patch-duplets for object recognition and pose estimation, in: Computer and Robot Vision, 2005.
- [63] F. Vikstén, P.-E. Forssén, B. Johansson, A. Moe, Comparison of local image descriptors for full 6 degree-of-freedom pose estimation, in: IEEE International Conference on Robotics and Automation, 2779–2786, 2009.

- [64] B. Leibe, A. Leonardis, B. Schiele, Robust Object Detection with Interleaved Categorization and Segmentation, *International Journal of Computer Vision* 77 (1-3) (2008) 259–289, ISSN 0920-5691.
- [65] J. Matas, O. Chum, U. Martin, T. Pajdla, Robust wide baseline stereo from maximally stable extremal regions, in: *British Machine Vision Conference*, vol. 1, 384–393, 2002.