Appearing in the proceedings of the 10th Conference on Computer and Robot Vision (CRV), 2013.

Continuous Pose Estimation in 2D Images at Instance and Category Levels

Damien Teney
University of Liège, Belgium
Damien.Teney@ULg.ac.be

Justus Piater
University of Innsbruck, Austria
Justus.Piater@UIBK.ac.at

***Abstract*—We present a general method for tackling the related problems of pose estimation of known object instances and object categories. By representing the training images as a probability distribution over the joint appearance/pose space, the method is naturally suitable for modeling the appearance of a single instance of an object, or of diverse instances of the same category. The training data is weighted and forms a generative model, the weights being based on the informative power of each image feature for specific poses. Pose inference is performed through probabilistic voting in pose space, which is intrinsically robust to clutter and occlusions, and which we render tractable by treating separately the least interdependent dimensions. The scalability of category-level models is ensured during training by clustering the available image features in the joint appearance/pose space. Finally, we show how to first efficiently use a category-model, then possibly recognize a particular trained instance to refine the pose estimate using the corresponding instance-specific model. Our implementation uses edge points as image features, and was tested on several existing datasets. We obtain results on par with or superior to state-of-the-art methods, on both instance- and category-level problems, including for generalization to unseen instances.**

I. Introduction and related work

The problem we focus on is the localization and the estimation of the precise 3D pose of objects in a new scene, given a single image of that scene, and multiple images of the objects as training examples. This is a central problem in computer vision, and there exists a wealth of literature on the topic, especially when dealing with specific object *instances*, e.g. a particular car or a particular coffee mug. The classical methods rely on the use discriminative image features and descriptors (such as SIFT or Geometric Blur), matched between the test view and the training examples. Such features are sometimes stored together with a rigid explicit 3D model of the object [1], [2], which brings viewpoint-invariance to the model. Other techniques have been proposed to encode viewpoint-invariant models, especially in the context of object recognition, e.g. by linking the observed features across different viewpoints [3], [4], [5], or modeling the object as a collection of planar parts [4]. Those methods however were used mainly with the goal or *localizing* and *recognizing* those objects in the images, but without recovering their 3D pose explicitly. One exception is the work of Savarese *et al.* [4], but the recovered pose is only a rough identification, such as "frontal view" or "side view". This limitation is present in many other methods [6], [7], [4], [8] which use discretized pose values, treated as separate classes, with different classifiers tuned to each of them. There exist however methods, often presented in the robotics community (with applications such as object grasping in mind), which can provide accurate pose estimates [9], [10], but they are mostly limited to specific object instances.

One particular aspect we are interested in is to provide the capability for pose estimation at the *category* level. There is an increased interest for this more challenging task; the goal is for example to train the system with a set of different mugs, then to recognize the pose of a new, unseen mug. The categories in such a scenario are defined implicitly by the training instances used as examples.

Previous work on object recognition does acknowledge the close link between handling the variability of object appearance as a function of pose and due to the diversity of objects within a category. Gu and Ren [11] showed how to solve for instance and *discrete* (coarse) pose recognition at the same time. Lai *et al.* [12] did so as well, using a tree of classifiers tuned for the different tasks. However, they use presegmented views of the objects, without any clutter or occlusions, and provide modest results on the accuracy of the retrieved pose. The methods mentioned in the previous paragraphs, while modeling the change of appearance due to different viewpoints, generally cannot directly handle the variability within *categories* of objects [3], [5]. One way this capability has been provided is by encoding — in addition to a rough 3D model — the possible variations in appearance [13], [14]; one limitation however is that no shape variability is possible. Our model, on the contrary, is purely appearance-based, and naturally accommodates variability in shape as well as in appearance. The traditional models of rigid geometrical constraints and highly discriminative features [2] are not adequate for encoding within-category variations. One exception to most methods here is again the model of Savarese *et al.* [4], which is specifically designed to provide viewpoint-invariance while handling within-category differences — but still provides only coarse pose estimates.

Recently, some methods have been introduced that can handle category variability and perform localization together with *precise* pose estimation. Glasner *et al.* [15] 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 us. They perform pose infer-

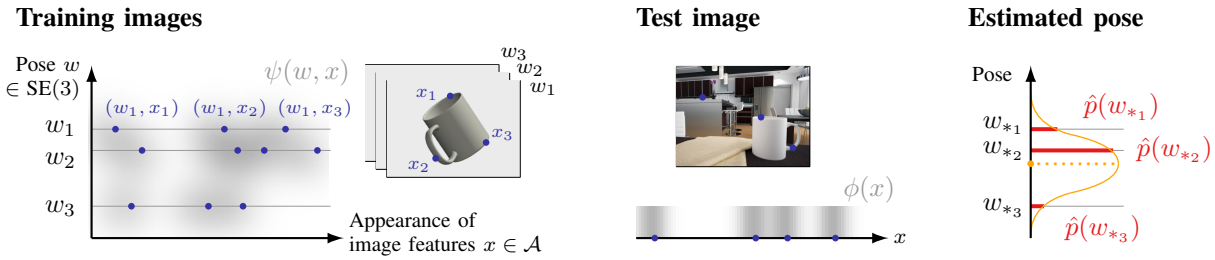

Fig. 1: Proposed method for representing training/test data and for pose estimation. Images features (blue points) are extracted from training and images; their appearance descriptor (in the case of our edge points, a position and orientation in the image) is defined on the generic space $\mathcal{A}$. Training/test observations define, using KDE, continuous probability distributions, respectively ψ and ϕ (gray shaded areas). Our pose inference algorithm (Fig. 2) returns approximations (red bars) of the pose likelihood function $p(w)$ at some discrete poses w_{*_i}. Finally, we locally fit, on those approximations, a simple distribution in the pose space (orange curve), and keep its mean as our final, precise pose estimate (orange dot).

ence through probabilistic voting in the 6D pose space, again in a similar way as we do, thereby solving for localization and viewpoint identification together. However, the reconstruction of such dense 3D models relies on the initial availability of a large number of views. By contrast, the appearance-based model used in this paper can use an arbitrary number of views and can be incrementally updated as more views become available. In a very different approach, Torki and Elgammal [16] learn a regression from local image features to the pose of the object. They recover 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. On the contrary, our method is framed from the start in the context of the full 3D pose.

Our method can accommodate different types of image features, but we chose to use very basic points along edges (combined with their tangent orientation) as opposed to more elaborate features such as SIFT descriptors. Recognition by matching such descriptors, while easier with specific instances, does not easily extend well to object categories. We differ from most edge-based shape recognition methods (e.g. [17], among many others) by avoiding intermediate representations such as contour fragments, and leveraging the simplicity of low-level features — in our implementation, simple points along edges. These simple features provide invariance to viewpoint and to within-category changes of appearance. Using such non-discriminative features for recognition however raises an additional challenge, since no matching is possible. This motivated the use of the framework proposed by Teney and Piater [18] for pose estimation in 2D images, which does not rely on correspondences between the test and the training data. Like [4], this model is generative and does not include any discriminative aspects, but has however been shown to be useful for localization and recognition in the presence of heavy clutter and occlusions [18]. Compared to that work, (1) we use a more efficient method for pose inference that does not need to consider the whole 6D pose space at once, (2) we introduce a weighting scheme of the training features which, as we will show, enhances significantly the performance of the system, and (3) we extend the methodology from instance-specific to category-level models.

The capabilities of the approach proposed in this paper differ from existing work by (1) handling, within the same framework, *instance*-specific models and *category*-specific models of objects, in the latter case allowing variations in shape and appearance, (2) performing *continuous* (precise) 3D pose estimation using those models, as opposed to viewpoint classification and coarse pose estimates, and (3) using such models to solve pose estimation *and* image localization together, as opposed to competing methods that do not handle clutter or occlusions. In addition, we present how to use category- and instance-level models successively, for optimal accuracy and efficiency: the category-model is used first to recover an initial pose estimate, which then allows one to possibly recognize a particular trained instance, so that the corresponding instance-specific model can be used to refine the pose estimate. Finally, in Section IV, the performance of our approach is compared to the most closely related methods [7], [4], [16]; we obtained promising results, on par with or superior to published data.

II. Pose estimation of specific object instances

A. Probabilistic representation of input data

The method we use is based on a probabilistic representation of both the training and the test data. This approach can be seen as a smoothing over the available data, providing continuous distributions of features and interpolating, to some extent, between the available data points (see Fig. 1, left and middle). Practically, the training examples are a set of K images of the object to learn, each annotated with the 3D pose of the object, $w_k \in \mathrm{SE}(3)$ with $k = 1, \dots, K$. We extract, from each training image, features x_i, which are edge points (see Section IV) with their tangent orientation, and which are thus defined on $\mathbb{R}^2 \times S_1^+$ (accounting for the position in the image, plus an orientation without direction). In the general case, we will call this space the *appearance* space, $\mathcal{A}$. We then

pair all features x_i of a view k with the pose w_k, so that we obtain a set of *pose/appearance pairs* $(x_i, w_k)_i$. Considering the whole training set, the pairs from all example images are concatenated to form our full training set $\mathcal{T} = \{(w_i, x_i)\}_{i=1}^{M}$, with $x_i \in \mathcal{A}$, and $w_i \in \mathrm{SE}(3)$.

The elements of our training set are then simply used to define a continuous probability distribution ψ on the pose/appearance space, in a non-parametric manner, with kernel density estimation:

$$\psi(w, x) \;=\; \frac{1}{M} \sum_{(w_i, x_i) \in \mathcal{T}} \mathsf{K}_1(w, w_i)\, \mathsf{K}_2(x, x_i)\;, \qquad (1)$$

where $w \in \mathrm{SE}(3)$ and $x \in \mathcal{A}$. The kernel functions $\mathsf{K}_1(\cdot,\cdot)$ and $\mathsf{K}_2(\cdot,\cdot)$ handle respectively the pose and the appearance spaces. Details on suitable kernels can be found, e.g. in [18], [19]; the first is an isotropic kernel allowing small deviations in both position and orientation, and the second, similarly, allows small variations in the location in the image and tangent orientation of the image feature.

The test data, which is a single 2D image of a new scene, is handled in a similar fashion as the training data. We extract the same type of image features, which we store as a set of *observations* $\mathcal{O} = \{x_i\}_{i=1}^{N}$, where $x_i \in \mathcal{A}$. This set is then used to define the continuous probability density ϕ on $\mathcal{A}$:

$$\phi(x) \;=\; \frac{1}{N} \sum_{x_i \in \mathcal{O}} \mathsf{K}_2(x, x_i)\;. \qquad (2)$$

As noted in [18], the transformations in the pose/appearance space corresponding to in-plane rotations/translations/scale changes are known from the camera calibration; those trivial transformations (e.g. a change in depth corresponds to a change of scale) are thus hard-coded. This allows us, when using ψ as a generative model, to extend its definition to parts of the pose space not explicitly covered by the training data.

B. Pose inference

The pose of the object of interest in the test scene is modeled as random variable $W \in \mathrm{SE}(3)$, the distribution of which is given by the likelihood function

$$p(w) \;=\; \int_{\mathcal{A}} \psi(w, x)\, \phi(x)\, \mathrm{d}x\;, \qquad (3)$$

This expression simply measures the compatibility of the training data at a pose w, with the distribution of features observed in the test image. The objective is to identify the main modes and peaks of the distribution of W, which was accomplished in [18] by a probabilistic voting scheme on the 6D pose space. This procedure is extremely costly in memory and processing [15], [18] due to the high dimensionality of the pose space. We now propose an approximation of that method that handles different dimensions of the pose space in different ways. Formally, a pose $w \in \mathrm{SE}(3)$ can be decomposed as a concatenation of 3 simpler entities, such that $w = w^3 \circ w^2 \circ w^1$. The first, w^1, corresponds to the "viewpoint", i.e. which side of the object is facing the camera; w^2 is a combination of an in-plane rotation and scale change, and w^3 corresponds to a pure translation parallel to the image plane. The main supporting observation for our proposed method is that a significant peak in the distribution of W will most likely appear as a peak in the distribution corresponding to the dimensions of w^3 alone. Indeed, an object of the test scene in any specific pose w will appear at a *precisely defined* image location (dimensions of w^3). This leads to the algorithm presented in Fig. 2, which iterates over discretized values for the dimensions of w^1 and w^2, and uses probabilistic voting only on the dimensions of w^3 (the 2D localization in the image). The peaks in those last two dimensions are thus identified by the algorithm for discrete viewpoints, scale and in-plane rotation values. This formulation is reminiscent of the classical Hough voting scheme used extensively for object localization [20]. The main advantage over [15], [18] is to avoid considering the entire pose space at once.

Input: training pairs $\mathcal{T} = \{(w_i, x_i)\}_i$ defining ψ
test observations $\mathcal{O} = \{x_i\}_i$ defining ϕ
Output: set $\mathcal{R}$ of approximations of the pose likelihood function
$\mathcal{R} = \{(w_{*_i}, \hat{p}(w_{*_i}))\}_i$

Procedure:
$\mathcal{R} \leftarrow \emptyset$
For each discrete w^1 in $\mathcal{T}$ (viewpoint)
  For each discrete step of w^2 (in-plane rotation and scale)
    Considering pose $w' = w^2 \circ w^1$,
    find best w^3 (image translation) between $\psi(w', x)$ and $\phi(x)$:
      Get samples: $(w_i^\psi, x_i^\psi) \sim \psi(w', x)$
                   $x_j^\phi \sim \phi(x)$
      Each possible pairing (x_i^ψ, x_j^ϕ) cast a vote in space of w^3 of weight $\mathrm{wt}(w_i^\psi, x_i^\psi)$
    Keep highest density peak in vote space: w_*^3 of vote score s
    $\mathcal{R} \leftarrow \mathcal{R} \cup (w_*, s)$ with $w_* = w_*^3 \circ w^2 \circ w^1$

Fig. 2: Pose inference algorithm

We also propose an additional step for refining the pose estimate, beyond the precision of the discretized pose values. As illustrated in Fig. 1 (right), we use the peaks identified by the algorithm in the pose space, together with their score value, as approximations of the likelihood function $p(w)$ (Eq. 3) at some discrete "probing" points. We simplistically assume that the main modes in the underlying distribution of W must *locally* approximate a simple isotropic distribution in the pose space. We therefore locally fit such a distribution (isotropic Gaussian and von Mises-Fisher distributions [19]) on the main peaks of $p(w)$, using non-linear least squares. The mean of the fitted distribution is then retained as the peak of that particular mode of the distribution (Fig. 1, right). This provides a much more accurate estimate of the optimal pose(s) compared to the above algorithm (as demonstrated in Section IV-A), at a very small additional computational cost.

C. Weighting of training data

We now present a way of weighting the available training data. The model we use does not include any discriminative

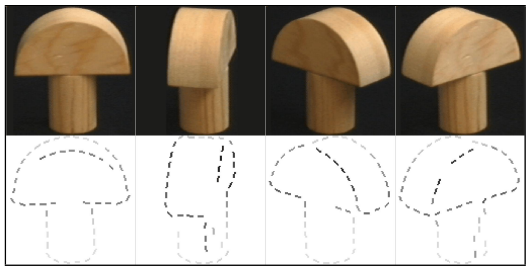

Fig. 3: Visualization of the weights attributed to each image feature (edge fragments) on a toy example; darker colors correspond to heavier weights. The parts looking similar in different views (e.g. the cylindrical base) receive lower weights, while the image features that can unambiguously determine a precise pose (e.g. non-silhouette edges) receive high weights.

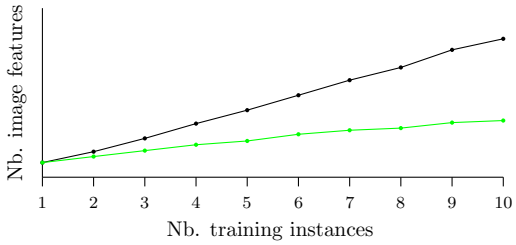

Fig. 4: Size of the category-level model of rotating cars built using different numbers of training instances: without (black) and with (green) the pruning of features by clustering. The proposed approach ensures a sublinear growth of the model.

aspects per se, and this weighting proved to significantly enhance the overall performance of the method (see Section IV). Appropriately weighting training data in the context of object recognition was previously shown to increase performance e.g. in [21], [22], [23], [24]. The formulation proposed here is different, suited to our non-discriminative low-level image features, and does not rely on massive amounts of training examples. The idea is to weight each image feature, depending on how informative it actually is for determining a specific pose. As detailed in the algorithm of Fig. 2, a training feature (w, x) is allowed to cast a vote of weight $\mathrm{wt}(w, x)$, given by

$$\mathrm{wt}(w,x) = 1 - \left[\frac{1}{K} \sum_{w':(w',\cdot)\in\mathcal{T}} \psi'(w',x)\big(1 - \mathsf{K}'_1(w,w')\big)\right] \quad (4)$$

with ψ' and K'_1 being variants of ψ and K_1 with maximum values of 1. This definition yields numerically-convenient weights in the range $[0, 1]$.

In Eq. 4, the expression in square brackets measures, for an image feature x observed in a training pose w, how likely this feature would be in poses very different than w. The weight is then defined using the opposite of that value. This effectively corresponds to the *specificity* of that feature x for the pose w (see also Fig. 3).

III. Learning object *category* models

The model and methods presented above naturally extend to *category*-level models. In that case, the training images include different objects, which together implicitly define the category. This capability of our model is due both to the fact that we can use very simple, non-discriminative image features (points along edges), which often generalize well across different objects of a same category, and by the non-parametric representation of the training data, which can naturally handle variability in the training data, in this case coming from several object instances.

Formally, each object instance $\ell \in [1, L]$ used for training produces a training set $\mathcal{T}_\ell$, as defined in Section II-A. A category-level model is then simply created using all features of all example instances, $\mathcal{T} = \bigcup_\ell \mathcal{T}_\ell$.

A. Pruning of training features by clustering

The above formulation uses a linearly growing number of training points (pose/appearance pairs) as more object instances are used to learn a given category. This correspondingly increases the computational requirements of using the model. Fortunately, object instances within a category often share common appearance traits, and the elements of $\mathcal{T}$ can thus be pruned at a very small cost of the representative capabilities of the model (as shown in Section IV). Practically, the elements of $\mathcal{T}$ are grouped using a simple agglomerative clustering scheme on the joint pose/appearance space, and only the cluster centers are retained. A maximum variance is enforced within the clusters, both in pose and appearance, which determines the amount of discarded training points. Note that the clustering procedure is most efficiently performed after normalizing the training examples from different instances for in-plane translation, rotation and scale, using the hardcoded transformations mentioned in Section II-A.

B. Recognizing a particular trained instance

The clustering of training features limits the size of a category model for efficiency. To compensate for lost accuracy, after identifying an initial pose estimate w_* with this category model, one can determine whether the recognized object corresponds to a specific trained instance. We measure the score of each trained instance ℓ at the pose w_* with

$$p_\ell(w_*) \;=\; \int_{\mathcal{A}} \psi_\ell(w_*, x)\, \phi(x)\, \mathrm{d}x \;, \quad (5)$$

where ψ_ℓ is defined as in Eq. 1, but using only the elements $\mathcal{T}_\ell$ of the instance ℓ. The value is easily approximated [18] with

$$p_\ell(w_*) \;\approx\; \frac{1}{n} \sum_{i}^{n} \psi_\ell(w, x_i) \quad \text{where} \quad x_i \sim \phi(x) \;. \quad (6)$$

If the value of $p_\ell(w_*)$ is significantly higher for a certain ℓ, the corresponding model of that instance ℓ (using all training data available for that instance) is then used to obtain a new, more accurate pose estimate (Section IV-A).

IV. Experimental evaluation

We now evaluate the proposed method under various conditions, using publicly-available datasets. We first analyze the incremental improvements in performance due to the individual ideas proposed in this paper. We then compare our results to existing, competing methods. The image features used are simple points identified along image edges, extracted with the classical Canny detector (see the examples in Fig. 3). Each of those points is characterized by its position in the image, and by the local orientation (smoothed for stability) of the edge at that point (an angle in $[0, \pi[$). As a ballpark figure of efficiency, on a standard laptop, our Matlab implementation of the method takes 20-30 seconds to process an image of the dataset of Section IV-B.

A. COIL Dataset

We first evaluate our method on the classical COIL dataset [25]. This dataset has been used in a variety of contexts, but not in the particular conditions we were interested in. The purpose of this part of our evaluation is to demonstrate the merits of the proposed method, by highlighting the incremental improvements brought by each proposed key point.

We selected a few objects from the original dataset, which correspond to reasonable categories (rectangular boxes, toy cars, flat bottles; see Fig. 5). Most other objects of the dataset were not suitable for estimating their pose (e.g. bell peppers, cylindrical cans) or could not be grouped into categories (e.g. duck toy). The dataset contains 72 images of each object undergoing a full rotation around a single (vertical) axis, with a fixed elevation. The estimated pose is thus similarly limited to this degree of freedom. For training, we use 18 images of each object (thus 20° apart), and the others for testing. We report the error as the median and mean (over all test images) of the absolute error of the estimated orientation. The rectangular boxes and the flat bottles present a 180° rotational symmetry, the error is accordingly evaluated on the half-circle.

1) Seen instances: The first series of tests uses 4 instances of each object (2 for the bottles) for training category models, and those same objects for testing. The basic method (algorithm of Fig. 2 without weighting the training data) already provides accurate results (see Fig. 5), with a median error of 5° which is the best achievable for the nearest-neighbour classification of the algorithm (Fig. 2) iterating on the discrete viewpoint values of the training data. The *mean* error decreases as we use the weights on the training data, as a few ambiguous test images are now better classified, which indicates the superior discriminability between different poses when using those weights. Interestingly, the fitting of a distribution on the pose space over the discrete approximations of the likelihood function (Section II-B) reduces the error significantly, as this allows accuracy beyond the resolution of the nearest-neighbour classification mentioned above. Finally, we refine the pose using the procedure proposed in Section III-B: the pose estimate obtained with the category model is used to efficiently check the resemblance with a particular trained instance. If one trained instance receives a significantly higher likelihood than the others (Eq. 6), its corresponding *instance-specific* model is used to perform a (hopefully) more accurate estimation; this is indeed the case as reported in Fig. 5. This procedure thus makes use of both the category- and instance-models for best efficiency without sacrificing accuracy.

	Toy cars		Boxes		Flat bottles	
Seen instances						
Without weights	5.0	12.1	5.0	13.5	5.0	7.9
With weights on training data	5.0	10.1	5.0	11.6	5.0	8.3
Weights + pruning of train. data	5.0	8.7	5.0	11.0	5.0	6.8
Weights + pruning + fitting of dist.	2.9	7.2	3.4	10.2	5.5	6.1
Refined w/ instance-specific model	2.0	5.8	3.2	9.4	4.2	5.3
Unseen instances						
Without weights	10.0	36.8	10.0	14.4	25.0	28.2
With weights on training data	5.0	39.7	10.0	11.0	30.0	31.2
Weights + pruning of train. data	10.0	44.6	10.0	11.8	15.0	25.5
Weights + pruning + fitting of dist.	2.9	41.8	4.3	8.8	16.8	23.6

Fig. 5: Results of category-level pose estimation with objects from the COIL dataset. Image top row: objects used for training and as *seen* test instances; image bottom row: objects used as *unseen* test instances. We report median (black) and mean (gray) error in degrees; large mean error is caused by (near-)symmetries which often induce errors of 90° and 180°.

2) Unseen instances: The second series of tests uses the same category models, but with a test set of other, *unseen* objects (Fig. 5, second row). The purpose is to verify the generalization capability of the category models. The results, as reported in Fig. 5, show accurate pose estimation results in all of the 3 tested categories, even though the test objects vary in shape, appearance and proportions from the training instances. This is made possible by the combination of different appearance traits of different training instances, which is possible in our non-parametric representation of the model. The flat bottles however yielded slightly worse results, which indicate the difficulty of generalizing the appearance of such objects on the category level. A test view of a novel instance could equally correspond to a wide bottle seen from its side, or to a front-facing thin one.

B. Rotating cars

We evaluated our method using the "Multiview car dataset" used by [7] and [16]. It includes about 2000 images of 20 very different rotating cars filmed at a motor show. The dataset is very challenging due to clutter, changing lighting conditions, high within-class variance in appearance, shape and texture, and highly symmetric side views, or similar front and rear views, which are sometimes hard to discriminate even for a human. The dataset was used in [7] for pose classification in 16 discrete bins, and in [16] for continuous pose estimation.

Number of training examples	15	30	40
Baseline comparison: Torki and Elgammal [16]	5.47	1.93	1.84
Without weights	6.75	3.83	2.94
With proposed weights on training data	6.68	3.81	2.91
Weights + fitting of pose distribution	**4.42**	**1.62**	**1.49**

Fig. 6: Results of pose estimation on a single car; mean error in degrees.

We first evaluated our method, as in [16], on the first car of the dataset, using thus an instance-specific model. We select 15, 30 or 40 equally-spaced images of the sequence as training images, and use all other images (spaced about 3–4° apart) for testing. Using all the key techniques proposed in this paper, we obtain superior results to [16] (see Fig. 6 for details). We then performed an evaluation the "10/10 split", where the first 10 cars of the dataset are used for training, and the other 10 for testing. We obtain again accurate pose estimation results. As highlighted in Fig. 8, most estimated poses are very accurate, while a number have an error of about 180°. This is caused by the symmetric aspects of some cars in the side views, as well as to confusion between front- and rear-facing views. This explains the seemingly large error reported as the mean in Fig. 7, even though the median error is clearly better than the results reported by [16]. In this case, the median as an evaluation metric better reflects the actual precision of the pose estimates, focusing on all the "successful" test cases.

We tested again the generalization capabilities of our model. As proposed in [7], we used the model trained on the cars at the motor show for testing on the database of Savarese *et al.* [4]. The cars appear here in natural environments with more clutter and in very diverse conditions; nevertheless, we obtained interesting results, of which we show some representative examples in Fig. 9. This again demonstrates the good capability of our system to generalize category-level models to conditions very different from those trained for. Note that, unfortunately, no quantitative results for these particular test conditions (proposed in [7]) — that we could compare to — were previously reported.

As a side note, let us mention that we tested our method on this same dataset [4] under the conditions of [8], i.e. training the model with 5 instances of that dataset. We obtained performance on pose estimation of the same order of magnitude as [8], but we missed some information for an exact quantitative comparison (which instances to use for training, and whether or not to include pose estimation results of inaccurate detections). Those experimental conditions were also evaluating coarse pose classification, whereas we focus on continuous pose estimation.

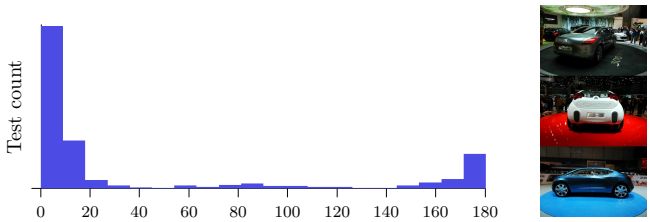

Fig. 8: Histogram showing distribution of error (in degrees) during experiments on multiple cars and sample test images that yielded an error of about 180°, due to ambiguous appearance.

Fig. 9: Detection and pose estimation results on the database of [4], using the model trained with Fig. 7. Boxes indicate the localization of the object 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 [7]). The last column contains failure cases, often due to the symmetrical appearance of the cars, or to too much clutter in the background.

V. Conclusions and Future Work

We presented a framework for representing the appearance of object instances or categories, together with its mechanisms to perform object localization and pose estimation in 2D images. The training examples are represented by a probability distribution, stored in a non-parametric manner, in the joint pose/appearance space. This approach can naturally represent a single object, or a whole object category by including different training exemplars of that category. The localization and identification of the pose of the object in a new scene is accomplished via probabilistic voting in the pose space, intrinsically robust to background clutter and occlusions. The overall approach was shown to be competitive or outperform comparable methods. As future work, it will be interesting to evaluate the method in the context of robotic applications,

	Median	Mean 90%ile	Mean	Error<22.5°	Error<45°	Used training features
Baseline comparison: Ozuysal *et al.* [7]	–	–	46.5	41.7%	71.2%	
Baseline comparison: Torki and Elgammal [16]	11.3	19.4	34.0	70.3%	80.7%	
Without weights on training data	9.3	33.1	47.4	65.1%	70.0%	100%
With weights and fitting of distribution	5.8	23.7	39.0	78.1%	79.7%	100%
Same + moderate pruning of features	6.1	25.8	41.0	77.0%	78.7%	54%
Same + aggressive pruning of features	9.4	32.4	46.8	67.1%	70.0%	30%

Fig. 7: Results of pose estimation on multiple cars; instances 1–10 used for training (top), 11–20 for testing (bottom). Errors of 180° are common (e.g. on instances 16 and 19) and explain the greater mean but smaller median error compared to [16].

with training sets spanning the whole viewing sphere around the objects to learn.

Acknowledgments

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.

References

[1] V. Ferrari, T. Tuytelaars, and L. J. V. Gool, "Simultaneous object recognition and segmentation from single or multiple model views," *Int. J. Comp. Vis. (IJCV)*, vol. 67, no. 2, pp. 159–188, 2006. 1

[2] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *Int. J. Comp. Vis. (IJCV)*, vol. 66, no. 3, pp. 231–259, 2006. 1

[3] A. Kushal, C. Schmid, and J. Ponce, "Flexible object models for category-level 3D object recognition," in *IEEE Int. Conf. on Comp. Vis. and Patt. Rec. (CVPR)*, 2007. 1

[4] S. Savarese and L. Fei-Fei, "3D generic object categorization, localization and pose estimation," in *IEEE Int. Conf. on Comp. Vis. (ICCV)*, 2007. 1, 2, 6

[5] A. Thomas, V. Ferrar, B. Leibe, T. Tuytelaars, B. Schiel, and L. Van Gool, "Towards multi-view object class detection," in *IEEE Int. Conf. on Comp. Vis. and Patt. Rec. (CVPR)*, 2006. 1

[6] J. Liebelt, C. Schmid, and K. Schertler, "Viewpoint-independent object class detection using 3D feature maps," in *IEEE Int. Conf. on Comp. Vis. and Patt. Rec. (CVPR)*, 2008. 1

[7] M. Ozuysal, V. Lepetit, and P. Fua, "Pose estimation for category specific multiview object localization," in *IEEE Int. Conf. on Comp. Vis. and Patt. Rec. (CVPR)*, 2009. 1, 2, 5, 6, 7

[8] M. Sun, H. Su, S. Savarese, and L. Fei-Fei, "A multi-view probabilistic model for 3D object classes," in *IEEE Int. Conf. on Comp. Vis. and Patt. Rec. (CVPR)*, 2009. 1, 6

[9] M. Martinez Torres, A. Collet Romea, and S. Srinivasa, "MOPED: A scalable and low latency object recognition and pose estimation system," in *IEEE Int. Conf. on Rob. and Autom. (ICRA)*, 2010. 1

[10] F. Viksten, R. Soderberg, K. Nordberg, and C. Perwass, "Increasing pose estimation performance using multi-cue integration," in *IEEE Int. Conf. on Rob. and Autom. (ICRA)*, 2006. 1

[11] C. Gu and X. Ren, "Discriminative mixture-of-templates for viewpoint classification," in *IEEE Europ. Conf. on Comp. Vis. (ECCV)*, 2010. 1

[12] K. Lai, L. Bo, X. Ren, and D. Fox, "A scalable tree-based approach for joint object and pose recognition," in *Conf. on Artificial Intelligence (AAAI)*, 2011. 1

[13] D. Hoiem, C. Rother, and J. M. Winn, "3D LayoutCRF for multi-view object class recognition and segmentation," in *IEEE Int. Conf. on Comp. Vis. and Patt. Rec. (CVPR)*, 2007. 1

[14] P. Yan, S. M. Khan, and M. Shah, "3D model based object class detection in an arbitrary view," in *IEEE Int. Conf. on Comp. Vis. (ICCV)*, 2007. 1

[15] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich, "Viewpoint-aware object detection and continuous pose estimation," *Image and Vision Computing*, 2012. 1, 3

[16] M. Torki and A. M. Elgammal, "Regression from local features for viewpoint and pose estimation," in *IEEE Int. Conf. on Comp. Vis. (ICCV)*, 2011. 2, 5, 6, 7

[17] A. Opelt, A. Pinz, and A. Zisserman, "Learning an alphabet of shape and appearance for multi-class object detection," *Int. J. Comp. Vis. (IJCV)*, 2008. 2

[18] D. Teney and J. Piater, "Generalized Exemplar-Based Full Pose Estimation from 2D Images without Correspondences," in *Digital Image Computing: Techniques and Applications (DICTA)*, 2012. 2, 3, 4

[19] R. Detry and J. Piater, "Continuous surface-point distributions for 3D object pose estimation and recognition," in *Asian Conf. on Comp. Vis. (ACCV)*, 2010. 3

[20] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *Int. J. Comp. Vision (IJCV)*, vol. 77, no. 1-3, pp. 259–289, May 2008. 3

[21] A. Frome, Y. Singer, F. Sha, and J. Malik, "Learning globally-consistent local distance functions for shape-based image retrieval and classification," in *IEEE Int. Conf. on Comp. Vis. (ICCV)*, 2007. 4

[22] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik, "Recognition using regions," in *IEEE Int. Conf. on Comp. Vis. and Patt. Rec. (CVPR)*, 2009. 4

[23] S. Maji and J. Malik, "Object detection using a max-margin hough transform," in *IEEE Int. Conf. on Comp. Vis. and Patt. Rec. (CVPR)*, 2009. 4

[24] P. Yarlagadda and B. Ommer, "From meaningful contours to discriminative object shape," in *IEEE Europ. Conf. on Comp. Vis. (ECCV)*, 2012. 4

[25] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library COIL-100," Columbia University, Tech. Rep., 1996. 5