Pose from Shape: Deep Pose Estimation for Arbitrary 3D Objects

Yang Xiao
https://youngxiao13.github.io
Xuchong Qiu
http://imagine.enpc.fr/~qiux/
Pierre-Alain Langlois
http://imagine.enpc.fr/~langloip/
Mathieu Aubry
http://imagine.enpc.fr/~aubrym/
Renaud Marlet
http://imagine.enpc.fr/~marletr/

LIGM (UMR 8049)
Ecole des Ponts ParisTech, UPE
Champs-sur-Marne, France

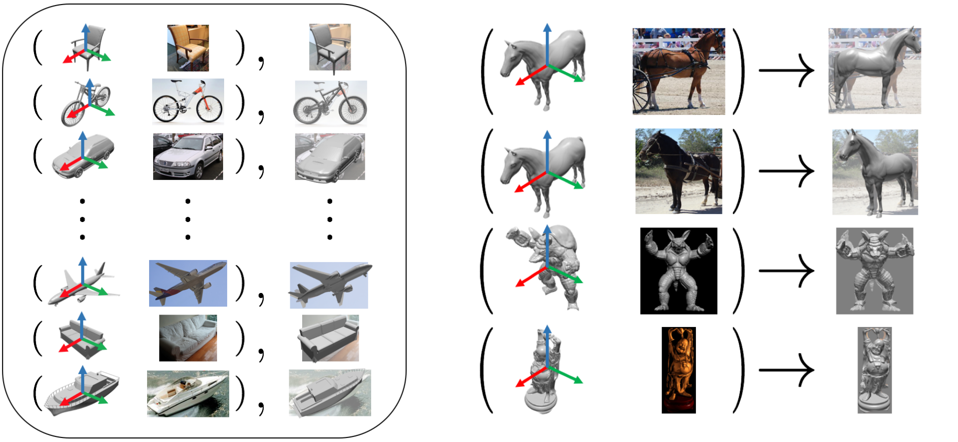
Abstract

Most deep pose estimation methods need to be trained for specific object instances or categories. In this work we propose a completely generic deep pose estimation approach, which does not require the network to have been trained on relevant categories, nor objects in a category to have a canonical pose. We believe this is a crucial step to design robotic systems that can interact with new objects "in the wild" not belonging to a predefined category. Our main insight is to dynamically condition pose estimation with a representation of the 3D shape of the target object. More precisely, we train a Convolutional Neural Network that takes as input both a test image and a 3D model, and outputs the relative 3D pose of the object in the input image with respect to the 3D model. We demonstrate that our method boosts performances for supervised category pose estimation on standard benchmarks, namely Pascal3D+, ObjectNet3D and Pix3D, on which we provide results superior to the state of the art. More importantly, we show that our network trained on everyday man-made objects from ShapeNet generalizes without any additional training to completely new types of 3D objects by providing results on the LINEMOD dataset as well as on natural entities such as animals from ImageNet. Our code and model is avalaible at http://imagine.enpc.fr/~xiaoy/PoseFromShape/.

1 Introduction

Imagine a robot that needs to interact with a new type of object not belonging to any predefined category, such as a newly manufactured object in a workshop. Using existing single-view pose estimation approaches for this new object would require stopping the robot and training a specific network for this object before taking any further action. Here we propose an approach that can directly take as input a 3D model of the new object and estimate the pose of the object in images relatively to this model, without any additional training procedure. We argue that such a capability is necessary for applications such as robotics "in the wild",

© 2019. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.



(a) Training with shape and pose (b) Testing on unseen objects

Figure 1: Illustration of our approach. (a) Training data: 3D model, input image and pose annotation for everyday man-made object; (b) At testing time, pose estimation of any arbitrary object, even an unknown category, given a RGB image and the corresponding 3D shape.

where new objects of unfamiliar categories can occur routinely at any time and have to be manipulated or taken into account for action. It also applies to virtual reality with similar circumstances.

To overcome the fact that deep pose estimation methods were category-specific, i.e., predicted different orientations according to object category, recent works [10, 54] have proposed to perform category-agnostic pose estimation on rigid objects, producing a single prediction. However, [10] only evaluated on object categories that were included in the training data, while [54] required the testing categories to be similar to the training data. On the contrary, we want to stress that our method works on novel objects that can be widely different from those seen at training time. For example, we can train only on man-made objects, but still be able to estimate the pose of animals such as horses, whereas not a single animal has been seen in the training data (cf. Fig. 1 and 3). Our method is similar to category-agnostic approaches in that it only produces one pose prediction and does not require additional training to produce predictions on novel categories. However, it is also instance-specific, because it takes as input a 3D model of the object of interest.

Indeed, our key idea is that viewpoint is better defined for a single object instance given its 3D shape than for whole object categories. Our work can be viewed as leveraging the recent advances in deep 3D model representations [37, 38, 40] for the problem of pose estimation. We show that using 3D model information also boosts performances on known categories, even when the information is only approximate, as in the Pascal3D+ [48] dataset.

When an exact 3D model of the object is known, as in the LINEMOD [15] dataset, state-of-the-art results are typically obtained by first performing a coarse viewpoint estimation and then applying a pose-refinement approach, typically matching rendered images of the 3D model to the target image. Our method is designed to perform the coarse alignment. Pose-refinement can be performed after applying our method using a classical approach based on ICP or the recent DeepIM [25] method. Note that while DeepIM only performs refinement, it is similar to our work in the sense that it is category agnostic and leverages some knowledge of the 3D model, using a view rendered in the estimated pose, to predict its pose update.

Our core contributions are as follows:

- To the best of our knowledge, we present the first deep learning approach to category-free viewpoint estimation, which can estimate the pose of any object conditioned only on its 3D model, whether or not it is similar to objects seen at training time.
- We can learn with and use "shapes in the wild", whose reference frame do not have to be consistent with a canonical orientation, simplifying pose supervision.
- We demonstrate on a large variety of datasets [15, 42, 48, 49] that adding 3D knowledge to pose estimation networks provides performance boosts when applied to objects of known categories, and meaningful performances on previously unseen objects.

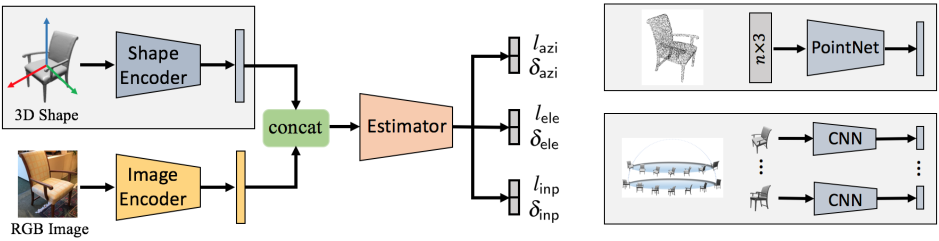
2 Related Work

In this section, we discuss pose estimation of a rigid object from a single RGB image first in the case where the 3D model of the object is known, then when the 3D model is unknown.

Pose estimation explicitly using object shape. Traditional methods to estimate the pose of a given 3D shape in an image can be roughly divided into feature-matching methods and template-matching methods. Feature-matching methods try to extract local features from the image, match them to the given object 3D model and then use a variant of PnP algorithm to recover the 6D pose based on estimated 2D-to-3D correspondences. Increasingly robust local feature descriptors [27, 34, 45, 46] and more effective variants of PnP algorithms [6, 21, 24, 53] have been used in this type of pipeline. Pixel-level prediction, rather than detected features, has also been proposed [1]. Although performing well on textured objects, these methods usually struggle with poorly-textured objects. To deal with this type of objects, template-matching methods try to match the observed object to a stored template [14, 15, 23, 26]. However, they perform badly in the case of partial occlusion or truncation.

More recently, deep models have been trained for pose estimation from an image of a known or estimated 3D model. Most methods estimate the 2D position in the test image of the projections of the object 3D bounding box [10, 32, 39, 43] or object semantic keypoints [9, 34] to find 2D-to-3D correspondences and then apply a variant of the PnP algorithm, as feature-matching methods. Once a coarse pose has been estimated, deep refinement approaches in the spirit of template-based methods have also been proposed [25, 29].

Pose estimation not explicitly using object shape. In recent years, with the release of large-scale datasets [8, 15, 42, 48, 49], data-driven learning methods (on real and/or synthetic data) have been introduced which do not rely on an explicit knowledge of the 3D models. These can roughly be separated into methods that estimate the pose of any object of a training category and methods that focus on a single object or scene. For category-wise pose estimation, a canonical view is required for each category with respect to which the viewpoint is estimated. The prediction can be cast as a regression problem [30, 33, 35], a classification problem [4, 41, 46] or a combination of both [12, 22, 28, 31]. Besides, Zhou *et al*. directly regress category-agnostic 3D keypoints and estimate a similarity between image and world coordinate systems [54]. Following the same strategy, it is also possible to estimate the pose of a camera with respect to a single 3D model but without actually using the 3D model information. Many recent works have applied this strategy to recover the full 6-DoF pose for object [17, 22, 31, 44, 50] and camera re-localization in the scene [18, 19].



(a) Our pose estimation approach (b) Two possible shape encoders

Figure 2: Overview of our method. (a) Given an RGB image of an object and its 3D shape, we use two encoders to extract features from each input, then estimate the orientation of the pictured object w.r.t. the shape using a classification-and-regression approach, predicting probabilities of angle bins l and bin offsets δ, for azimuth, elevation and in-plane rotation. (b) For shape encoding, we encode a point cloud sampled on the object with PointNet (top), or we rendered images around the object and use a CNN to extract the features (bottom).

In this work, we propose to merge the two lines of work described above. We cast pose estimation as a prediction problem, similar to deep learning methods that do not explicitly leverage viewpoint information. However, we condition our network on the 3D model of a single instance, represented either by a set of views or a point cloud, allowing our network to rely on the exact 3D model, similarly to the feature and template matching methods. To the best of our knowledge, we are the first to combine image and shape information as input to a network to estimate the relative orientation of the depicted object with respect to the shape.

3 Network Architecture and Training

Our approach consists in extracting deep features from both the image and the shape, and using them jointly to estimate a relative orientation. An overview is shown in Fig. 2. In this section, we present in more details our architecture, our loss function and our training strategy, as well as a data augmentation scheme specifically designed for our approach.

Feature extraction. The first part of the network consists of two independent modules: (i) image feature extraction and (ii) 3D shape feature extraction. For image features, we use a standard CNN, namely ResNet-18 [13]. For 3D shape features, we experimented with two approaches depicted in Fig. 2(b) which are state-of-the-art 3D shape description networks.

First, we used the point set embedding network PointNet [37], which has been successfully used as a point cloud encoder for many tasks [5, 11, 36, 47, 52].

Second, we tried to represent the shape using rendered views, similar to [40]. Virtual cameras are placed around the 3D shape, pointing towards the centroid of the model; the associated rendered images are taken as input by CNNs, sharing weights for all viewpoints, which extract image descriptors; a global feature vector is obtained by concatenation. We considered variants of this architecture using extra input channels for depth and/or surface normal orientation but this did not improve our results significantly. Ideally, we would consider viewpoints on the whole sphere around the object with any orientation. In practice however, many objects have a strong bias regarding verticality and are generally seen only from the side/top. In our experiments, we thus only considered viewpoints on the top hemisphere and sampled evenly a fixed number of azimuths and elevations.

Orientation estimation. The object orientation is estimated from both the image and 3D shape features by a multi-layer perceptron (MLP) with three hidden layers of size 800-400-200. Each fully connected layer is followed by a batch normalization, and a ReLU activation.

As output, we estimate the three Euler angles of the camera, azimuth (azi), elevation (ele) and in-plane rotation (inp), with respect to the shape reference frame. Each of these angles $\theta \in \mathcal{E} = \{\mathsf{azi}, \mathsf{ele}, \mathsf{inp}\}$ is estimated using a mixed classification-and-regression approach, which computes both angular bin classification scores and offset information within each bin. Concretely, we split each angle $\theta \in \mathcal{E}$ uniformly in L_θ bins. For each θ-bin $l \in \{0, L_\theta - 1\}$, the network outputs a probability $\hat{p}_{\theta,l} \in [0,1]$ using a softmax non-linearity on the θ-bin classification scores, and an offset $\hat{\delta}_{\theta,l} \in [-1,1]$ relatively to the center of θ-bin l, obtained by a hyperbolic tangent non-linearity. The network thus has $2 \times (L_{\mathsf{azi}} + L_{\mathsf{ele}} + L_{\mathsf{inp}})$ outputs.

Loss function. As we combine classification and regression, our network has two types of outputs (probabilities and offsets), that are combined into a single loss $\mathcal{L}$ that is the sum of a cross-entropy loss for classification $\mathcal{L}_{\mathsf{cla}}$ and Huber loss [16] for regression $\mathcal{L}_{\mathsf{reg}}$.

More formally, we assume we are given training data $(x_i, s_i, y_i)_{i=1}^N$ consisting of input images x_i, associated object shapes s_i and corresponding orientations $y_i = (y_{i,\theta})_{\theta \in \mathcal{E}}$. We convert the value of the Euler angles $y_{i,\theta}$ into a bin label $l_{i,\theta}$ encoded as a one-hot vector and relative offsets $\delta_{i,\theta}$ within the bins. The network parameters are learned by minimizing:

$$\mathcal{L} = \sum_{i=1}^{N} \sum_{\theta \in \mathcal{E}} \mathcal{L}_{\mathsf{cla}}\Big(l_{i,\theta}, \hat{p}_\theta(x_i, s_i)\Big) + \mathcal{L}_{\mathsf{reg}}\Big(\delta_{i,\theta}, \hat{\delta}_{\theta, l_{i,\theta}}(x_i, s_i)\Big), \tag{1}$$

where $\hat{p}_\theta(x_i, s_i)$ are the probabilities predicted by the network for angle $\theta \in \mathcal{E}$, input image x_i and input shape s_i, and $\hat{\delta}_{\theta, l_{i,\theta}}(x_i, s_i)$ the predicted offset within the ground truth bin.

Data augmentation. We perform standard data augmentation on the input images: horizontal flip, 2D bounding box jittering, color jittering.

In addition, we introduce a new data augmentation, specific to our approach, designed to avoid the network to overfit the 3D model orientation, which is usually consistent in training data since most models are aligned. On the contrary, we want our network to be category-agnostic and to always predict the pose of the object with respect to the reference 3D model. We thus add random rotations to the input shapes, and modify the orientation labels accordingly. In our experiments, we restrict our rotations to azimuth changes, again because of the strong verticality bias in the benchmarks, but could theoretically apply it to all angles. Because of objects with symmetries, typically at 90° or 180°, we also restrict azimuthal randomization to a uniform sampling in $[-45°, 45°]$, which allows to keep the 0° bias of the annotations. See supplementary material for details and parameter study.

Implementation details. For all our experiments, we set the batch size as 16 and trained our network using the Adam optimizer [20] with a learning rate of 10^{-4} for 100 epochs then 10^{-5} for an additional 100 epochs. Compared to a shape-less baseline method, the training of our method with the shape encoded from 12 rendered views is about 8 times slower, on a TITAN X GPU.

4 Experiments

Given an RGB image of an object and a 3D model of that object, our method estimates its 3D orientation in the image. In this section, we first give an overview of the datasets we used, and explain our baseline methods. We then evaluate our method in two test scenarios: object belonging to a category known at training time, or unknown.

Datasets. We experimented with four main datasets. Pascal3D+ [48], ObjectNet3D [49] and Pix3D [42] feature various objects in various environments, allowing benchmarks for object pose estimation in the wild. On the contrary, LINEMOD [15] focuses on few objects with little environment variations, targeting robotic manipulation. Pascal3D+ and ObjectNet3D only provide approximate models and rough alignments while Pix3D and LINEMOD offer exact models and pixelwise alignments. We also used ShapeNetCore [2] for training on synthetic data, with SUN397 backgrounds [51], and tested on Pix3D and LINEMOD.

Unless otherwise stated, ground-truth bounding boxes are used in all experiments. We compute the most common metrics used with each dataset: $Acc_{\frac{\pi}{6}}$ is the percentage of estimations with rotation error less than 30°; *MedErr* is the median angular error (°); ADD-0.1d is the percentage of estimations for which the mean distance of the estimated 3D model points to the ground truth is smaller than 10% of the object diameter; ADD-S-0.1d is a variant of ADD-0.1d used for symmetric objects where the average is computed on the closest point distance. More details on the datasets and metrics are given in the supplementary material.

Baselines. A natural baseline is to use the same architecture, data and training strategy as for our approach, but without using the 3D shape of the object. This is reported as 'Baseline' in our tables, and corresponds to the network of Fig. 2 without the shape encoder shown in light blue. We also report a second baseline, aiming at evaluating the importance of the precision of the 3D model for our approach to work. We used exactly our approach, but at testing time we replaced the 3D shape of the object in the test image by a random 3D shape of the same category. This is reported as 'Ours (RS)' in the tables.

4.1 Pose estimation on supervised categories

We first evaluate our method in case the categories of tested objects are covered by training data. We show that leveraging the 3D model of the object clearly improves pose estimation.

We evaluate our method on ObjectNet3D, which has the largest variety of object categories, 3D models and images. We report the results in Table 1 (top). First, an important result is that using the 3D model information, whether via a point cloud or rendered views, provides a very clear boost of the performance, which validates our approach. Second, results using rendered multiple views (MV) to represent the 3D model outperform the point-cloud-based (PC) representation [37]. We thus only evaluated Ours(MV) in the rest of this section. Third, testing the network with a random shape (RS) in the category instead of the ground truth shape, implicitly providing class information without providing fine-grained 3D information, leads to results better than the baseline but worst than using the ground truth model, demonstrating our method ability to exploit fine-grained 3D information. Finally, we found that even our baseline model already outperformed StarMap [54], mainly because of five categories (iron, knife, pen, rifle, slipper) on which StarMap completely fails, likely because a keypoint-based method is not adapted for small and narrow objects.

ObjectNet3D	bed	bcase	calc	cphone	comp	door	cabi	guit	iron	knife	micro	pen	pot	rifle	shoe	slipper	stove	toilet	tub	wchair	mean
category-specific networks/branches — test on supervised categories ($Acc_{\frac{\pi}{6}}$)																					
Xiang [49]*	61	85	93	60	78	90	76	75	17	23	87	33	77	33	57	22	88	81	63	50	62
category-agnostic network — test on supervised categories ($Acc_{\frac{\pi}{6}}$)																					
Zhou [54]	73	78	91	57	82	–	84	73	3	18	94	13	56	4	–	12	87	71	51	60	56
Baseline	70	89	90	55	87	91	88	62	29	20	93	43	76	26	58	30	91	68	51	55	64
Ours(PC)	**83**	**92**	95	58	82	87	**91**	67	43	**36**	94	53	81	39	45	35	91	80	65	56	69
Ours(MV,RS)	74	89	91	62	81	90	88	71	41	28	94	50	70	37	57	38	89	81	60	60	68
Ours(MV)	82	90	**96**	**65**	**93**	**97**	89	**75**	**52**	32	**95**	**54**	**82**	**45**	**67**	**46**	**95**	**82**	**67**	**66**	**73**
category-agnostic network — test on novel categories ($Acc_{\frac{\pi}{6}}$)																					
Zhou [54]	37	69	19	52	73	–	78	61	2	9	88	12	51	0	–	11	82	41	49	14	42
Baseline	56	79	26	53	77	86	83	51	4	16	90	42	65	2	34	22	86	43	50	35	50
Ours(PC)	63	85	84	51	**85**	83	83	61	**9**	**35**	92	44	**80**	8	39	20	87	56	**71**	**39**	59
Ours(MV,RS)	60	88	84	60	76	91	82	61	2	26	90	46	73	13	45	28	79	59	61	36	58
Ours(MV)	**65**	**90**	**88**	**65**	84	**93**	**84**	**67**	2	29	**94**	**47**	79	**15**	**54**	**32**	**89**	**61**	68	**39**	**62**

[images: 90,127, in the wild | objects: 201,888 | categories: 100 | 3D models: 791, approx. | align.: rough]

Table 1: Pose estimation on ObjectNet3D [49]. Train and test are on the same data as [54]; for experiments on novel categories, training is on 80 categories and test is on the other 20. * Trained jointly for detection and pose estimation, tested using estimated bounding boxes.

Pascal3D+ [48]	aero	bike	boat	bottle	bus	car	chair	dtable	mbike	sofa	train	tv	mean	aero	bike	boat	bottle	bus	car	chair	dtable	mbike	sofa	train	tv	mean
Categ.-specific branches, supervised categ.	$Acc_{\frac{\pi}{6}}$ (%)													*MedErr* (degrees)												
Tulsiani [46]*	81	77	59	93	**98**	89	80	62	**88**	82	80	80	80.75	13.8	17.7	21.3	12.9	5.8	9.1	14.8	15.2	14.7	13.7	8.7	15.4	13.6
Su [41]†	74	**83**	52	91	91	88	**86**	**73**	78	**90**	**86**	**92**	82.00	15.4	14.8	25.6	9.3	3.6	6.0	**9.7**	**10.8**	16.7	**9.5**	**6.1**	12.6	11.7
Mousavian [31]	78	**83**	57	93	94	90	80	68	86	82	82	85	81.03	13.6	**12.5**	22.8	**8.3**	3.1	5.8	11.9	12.5	12.3	12.8	6.3	11.9	11.1
Pavlakos [34]*	81	78	44	79	96	90	80	–	–	74	79	66	–	**8.0**	13.4	40.7	11.7	**2.0**	5.5	10.4	–	–	9.6	8.3	32.9	–
Grabner [10]	**83**	82	**64**	**95**	97	**94**	80	71	**88**	87	80	86	**83.92**	10.0	15.6	**19.1**	8.6	3.3	**5.1**	13.7	11.8	**12.2**	13.5	6.7	**11.0**	**10.9**
Categ.-agnostic network, supervised categ.	$Acc_{\frac{\pi}{6}}$ (%)													*MedErr* (degrees)												
Grabner [10]	80	82	57	90	**97**	**94**	72	**67**	**90**	80	**82**	**85**	81.33	10.9	**12.2**	23.4	9.3	3.4	5.2	15.9	16.2	12.2	11.6	6.3	11.2	11.5
Zhou [54]*	**82**	**86**	50	92	**97**	92	**79**	62	88	**92**	77	83	81.67	**10.1**	14.5	30.0	9.1	3.1	6.5	11.0	23.7	14.1	11.1	7.4	13.0	12.8
Baseline	77	74	54	91	**97**	89	74	52	85	80	79	77	77.42	13.0	18.2	27.3	11.5	6.8	8.1	15.4	20.1	14.7	13.2	10.2	14.7	14.4
Ours(MV,RS)	79	81	49	91	96	89	78	53	**90**	88	80	77	79.25	11.6	15.5	30.9	8.2	3.6	6.0	13.8	22.8	13.1	11.1	6.0	15.0	13.1
Ours(MV)	81	83	**60**	**93**	**97**	91	**79**	**67**	**90**	90	81	79	**82.66**	10.5	13.7	**21.0**	**7.7**	**3.0**	**5.0**	**10.9**	**11.9**	**11.8**	**9.1**	**5.4**	**10.3**	**10.0**

[images: 30,889, in the wild | objects: 36,292 | categories: 12 | 3D models: 79, approx. | align.: rough]

Table 2: Pose estimation on Pascal3D+ [48]. * Trained using keypoints. † Not trained on ImageNet data but trained on ShapeNet renderings.

Pix3D [42]	tool	misc	bcase	wdrobe	desk	bed	table	sofa	chair	mean
category-specific networks — tested on supervised categories ($Acc_{\frac{\pi}{6}}$)										
Georgakis [9]	-	-	-	-	**34.9**	**50.8**	-	-	**31.2**	-
category-agnostic network — tested on supervised categories ($Acc_{\frac{\pi}{6}}$)										
Baseline	2.2	9.8	10.8	0.6	30.0	36.8	17.3	63.8	43.6	23.9
Ours(MV,RS)	4.1	3.6	22.8	9.5	52.8	50.1	30.8	66.3	44.5	31.6
Ours(MV)	**6.5**	**19.7**	**34.6**	**10.2**	**56.6**	**59.8**	**40.8**	**70.0**	**52.4**	**38.9**
category-agnostic network — tested on novel categories ($Acc_{\frac{\pi}{6}}$)										
Baseline	2.2	**13.1**	5.4	0.6	30.3	19.6	14.9	11.9	28.0	14.0
Ours(MV,RS)	3.0	5.9	4.5	5.2	44.7	31.5	24.1	48.5	33.9	22.4
Ours(MV)	**10.9**	**13.1**	**22.3**	**6.6**	**52.0**	**55.3**	**35.6**	**64.6**	**35.8**	**32.9**

[images: 10,069, in the wild | objects: 10,069 | categ.: 9 | 3D models: 395, exact | align.: pixel]

Pix3D [42]	chair	
categ.-specific, supervised		
# bins (% correct)	24 azim.	12 elev.
Su [41]	40	37
Sun [42]	49	61
Baseline	51	64
Ours(MV)	**54**	**65**

Table 3: Pose estimation on Pix3D [42]. Right table compares to [41, 42], that only test bin success on 2 angles (24 azimuth bins and 12 elevation bins).

We then evaluate our approach on the standard Pascal3D+ dataset [48]. Results are shown in Table 2 (top). Interestingly, while our baseline is far below state-of-the-art results, adding our shape analysis network provides again a very clear improvement, with results on par with the best category-specific approaches, and outperforming category agnostic methods. This is especially impressive considering the fact that the 3D models provided in Pascal3D+ are only extremely coarse approximations of the real 3D models. Again, as can be expected, using a random model from the same category provides intermediary results between the model-less baseline and using the actual 3D model.

Finally, we report results on Pix3D in Table 3 (top). Similar to the other methods, our model was purely trained on synthetic data and tested on real data, without any fine-tuning. Again, we can observe that adding 3D shape information brings a large performance boost, from 23.9% to 36% $Acc_{\frac{\pi}{6}}$. Note that our method clearly improves even over category-specific baselines. We believe it is due to the much higher quality of the 3D models provided on Pix3D compared to ObjectNet3D and Pascal3D+. This hypothesis is supported by the fact that our results are much worse when a random model of the same category is provided.

These state-of-the-art results on the three standard datasets are thus consistent and validate (i) that using the 3D models provides a clear improvement (comparison to 'Baseline'), and (ii) that our approach is able to leverage the fine-grained 3D information from the 3D model (comparison to estimating with a random shape 'RS' in the category).

4.2 Pose estimation on novel categories

We now focus on the generalization to unseen categories, which is the main focus of our method. We first discuss results on ObjectNet3D and Pix3D. We then show qualitative results on ImageNet horses images and quantitative results on the very different LINEMOD dataset.

Our results when testing on new categories from ObjectNet3D are shown in Table 1 (bottom). We use the same split between 80 training and 20 testing categories as [54]. As expected, the accuracy decreases for all methods when supervision is not provided on these latter categories. The fact that the Baseline performances are still much better than chance is accounted by the presence of similar categories is the training set. The advantage of our method is however even more pronounced than in the supervised case, and our multi-view approach (MV) still outperforms the point cloud (PC) approach by a small margin. Similarly, we removed from our ShapeNet [2] synthetic training set the categories present in Pix3D, and reported in Table 3 (bottom) the results on Pix3D. Again, the accuracy drops for all methods, but the benefit from using the ground-truth 3D model increases.

In both ObjectNet and Pix3D experiments, the test categories were novel but still similar to the training ones. We now focus on evaluating our network, trained using synthetic images generated from man-made shapes from ShapeNetCore [2], on completely different objects.

We first obtain qualitative results by using a fixed 3D model of horse from an online model repository [7] to estimate the pose of horses in ImageNet images. Indeed, compared to other animals, horses have more limited deformations. While this of course does not work for all images, the images for which the network provides the highest confidence are impressively good. On Figure 3, we show the most confident images for different poses, and we provide more results in the supplementary material. Note the very strong appearance gap between the rendered 3D models and the test images.

Finally, to further validate our network generalization ability, we evaluate it on the texture-less objects of LINEMOD [15], as reported in Table 4. This dataset focuses on

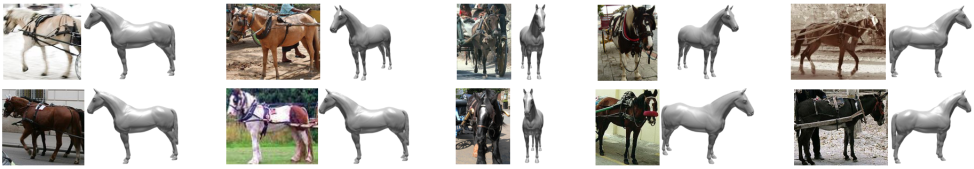


Figure 3: Visual results of pose estimation on horse images from ImageNet [3] using models from Free3D [7]. We rank the prediction for each orientation bin by the network prediction and show the first (best) results for various poses.

LINEMOD [15]		ape	bvise	cam	can	cat	drill	duck	ebox*	glue*	holep	iron	lamp	phone	mean
instance-specific networks/branches — tested on supervised models (ADD-0.1d)*															
w/o Ref.	Brachmann [1]	-	-	-	-	-	-	-	-	-	-	-	-	-	32.3
	SSD-6D [17]‡	0	0.2	0.4	1.4	0.5	2.6	0	8.9	0	0.3	8.9	8.2	0.2	2.4
	BB8 [39]	**27.9**	62.0	40.1	48.1	45.2	58.6	32.8	40.0	27.0	42.4	67.0	39.9	35.2	43.6
	Tekin [43]	21.6	**81.8**	36.6	68.8	41.8	63.5	27.2	69.6	80.0	42.6	**75.0**	**71.1**	47.7	56.0
	PoseCNN [50]†	27.8	68.9	**47.5**	**71.4**	**56.7**	**65.4**	**42.8**	**98.3**	**95.2**	**50.9**	65.6	70.3	**54.6**	**62.7**
w/ Ref.	Brachmann [1]	33.2	64.8	38.4	62.9	42.7	61.9	30.2	49.9	31.2	52.8	80.0	67.0	38.1	50.2
	BB8 [39]	40.4	91.8	55.7	64.1	62.6	74.4	44.3	57.8	41.2	**67.2**	84.7	76.5	54.0	62.7
	SSD-6D [17]‡	65	80	78	86	70	73	66	**100**	**100**	49	78	73	79	79.0
	PoseCNN [50]† + [25]†	**76.9**	**97.4**	**93.5**	**96.6**	**82.1**	**95.0**	**77.7**	97.0	99.4	52.7	**98.3**	**97.5**	**87.8**	**88.6**
instance/category-agnostic network — tested on novel models (ADD-0.1d)*															
w/o Ref.	Ours‡	**7.5**	**25.1**	**12.1**	**11.3**	**15.4**	**18.6**	**8.2**	**100**	**81.2**	**18.5**	**13.8**	**6.5**	**13.4**	**25.5**
w/ Ref.	Ours‡ + DeepIM [25]†	**59.1**	**63.8**	**40.0**	**50.8**	**54.1**	**75.3**	**48.6**	**100**	**98.7**	**49.8**	**49.5**	**55.3**	**50.4**	**61.2**

[scenes: 13, artificially arranged | images: 13407 | objects: 13 | categ.: 13 | 3D models: 13, exact | align.: pixel]

Table 4: Pose estimation on LINEMOD [15]. † Training also on synthetic data. ‡ Training only on synthetic data. * ADD-S-0.1d used for symmetric objects eggbox and glue.

very accurate alignment, and most approaches propose to first estimate a coarse alignment and then to refine it with a specific method. Our method provides a coarse alignment, and we complement it using the recent DeepIM [25] refinement approach. Our method yields results below the state of the art, but they are nevertheless very impressive. Indeed, our network has never seen objects any similar the LINEMOD 3D models during training, while all the other baselines have been trained specifically for each object instance on real training images, except SSD-6D [17] which uses the exact 3D model but no real image and for which coarse alignment performances are very low. Our method is thus very different from all the baselines in that it does not assume the test object to be available at training time, which we think is a much more realistic scenario for robotics applications. We actually believe that the fact our method provides a reasonable accuracy on this benchmark is a very strong result.

5 Conclusion

We have presented a new paradigm for deep pose estimation, taking the 3D object model as an input to the network. We demonstrated the benefits of this approach in terms of accuracy, and improved the state of the art on several standard pose estimation datasets. More importantly, we have shown that our approach holds the promise of a completely generic deep learning method for pose estimation, independent of the object category and training data, by showing encouraging results on the LINEMOD dataset without any specific training, and despite the domain gap between synthetic training data and real images for testing.

References

[1] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University – Princeton University – Toyota Technological Institute at Chicago, 2015.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[4] Mohamed Elhoseiny, Tarek El-Gaaly, Amr Bakry, and Ahmed M. Elgammal. A comparative analysis and study of multiview CNN models for joint object categorization and pose estimation. In *International Conference on Machine Learning (ICML)*, 2016.

[5] Francis Engelmann, Theodora Kontogianni, Alexander Hermans, and Bastian Leibe. Exploring spatial context for 3D semantic segmentation of point clouds. In *International Conference on Computer Vision (ICCV)*, 2017.

[6] Luis Ferraz, Xavier Binefa, and Francesc Moreno-Noguer. Very fast solution to the PnP problem with algebraic outlier rejection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[7] Free3D. Free3d. https://free3d.com.

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[9] Georgios Georgakis, Srikrishna Karanam, Ziyan Wu, and Jana Kosecka. Matching RGB images to CAD models for object pose estimation. *CoRR*, abs/1811.07249, 2018.

[10] Alexander Grabner, Peter M. Roth, and Vincent Lepetit. 3D pose estimation and 3D model retrieval for objects in the wild. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[11] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A papier-mâché approach to learning 3D surface generation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[12] Rıza Alp Güler, George Trigeorgis, Epameinondas Antonakos, Patrick Snape, Stefanos Zafeiriou, and Iasonas Kokkinos. Densereg: Fully convolutional dense shape regression in-the-wild. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] Stefan Hinterstoisser, Cedric Cagniart, Slobodan Ilic, Peter Sturm, Nassir Navab, Pascal Fua, and Vincent Lepetit. Gradient response maps for real-time detection of textureless objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2012.

[15] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Asian Conference on Computer Vision (ACCV)*, 2012.

[16] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*. Springer New York, 1992.

[17] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again. In *International Conference on Computer Vision (ICCV)*, 2017.

[18] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[19] Alex Kendall, Matthew Koichi Grimes, and Roberto Cipolla. PoseNet: A convolutional network for real-time 6-DOF camera relocalization. In *International Conference on Computer Vision (ICCV)*, 2015.

[20] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2014.

[21] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. EPnP: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision (IJCV)*, 2009.

[22] Chi Li, Jin Bai, and Gregory D. Hager. A unified framework for multi-view multi-class object pose estimation. In *European Conference on Computer Vision (ECCV)*, 2018.

[23] Dengwang Li, Hongjun Wang, Yong Yin, and Xiuying Wang. Deformable registration using edge-preserving scale space for adaptive image-guided radiation therapy. *Journal of Applied Clinical Medical Physics (JACMP)*, 2011.

[24] Shiqi Li, Chi Xu, and Ming Xie. A robust O(n) solution to the perspective-n-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2012.

[25] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep iterative matching for 6D pose estimation. In *European Conference on Computer Vision (ECCV)*, 2018.

[26] David G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 1991.

[27] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 2004.

[28] Siddharth Mahendran, Haider Ali, and René Vidal. A mixed classification-regression framework for 3D pose estimation from 2D images. In *British Machine Vision Conference (BMVC)*, 2018.

[29] Fabian Manhardt, Wadim Kehl, Nassir Navab, and Federico Tombari. Deep model-based 6D pose refinement in RGB. In *European Conference on Computer Vision (ECCV)*, 2018.

[30] Francisco Massa, Renaud Marlet, and Mathieu Aubry. Crafting a multi-task CNN for viewpoint estimation. In *British Machine Vision Conference (BMVC)*, 2016.

[31] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3D bounding box estimation using deep learning and geometry. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[32] Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Making deep heatmaps robust to partial occlusions for 3D object pose estimation. In *European Conference on Computer Vision (ECCV)*, 2018.

[33] Margarita Osadchy, Yann Le Cun, and Matthew L. Miller. Synergistic face detection and pose estimation with energy-based models. *Journal of Machine Learning Research (JMLR)*, 2007.

[34] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 6-dof object pose from semantic keypoints. In *International Conference on Robotics and Automation (ICRA)*, 2017.

[35] Hugo Penedones, Ronan Collobert, François Fleuret, and David Grangier. Improving object classification using pose information. Technical report, Idiap Research Institute, 2012.

[36] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3D object detection from RGB-D data. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[37] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[38] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Conference on Neural Information Processing Systems (NIPS)*, 2017.

[39] Mahdi Rad and Vincent Lepetit. BB8: a scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth. In *International Conference on Computer Vision (ICCV)*, 2017.

[40] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *International Conference on Computer Vision (ICCV)*, 2015.

[41] Hao Su, Charles R. Qi, Yangyan Li, and Leonidas J. Guibas. Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views. In *International Conference on Computer Vision (ICCV)*, 2015.

[42] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3D: Dataset and methods for single-image 3d shape modeling. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[43] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6D object pose prediction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[44] Henning Tjaden, Ulrich Schwanecke, and Elmar Schömer. Real-time monocular pose estimation of 3D objects using temporally consistent local color histograms. In *International Conference on Computer Vision (ICCV)*, 2017.

[45] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2010.

[46] Shubham Tulsiani and Jitendra Malik. Viewpoints and keypoints. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[47] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph CNN for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018.

[48] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond PASCAL: A benchmark for 3D object detection in the wild. In *Winter Conference on Applications of Computer Vision (WACV)*, 2014.

[49] Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Choy, Hao Su, Roozbeh Mottaghi, Leonidas Guibas, and Silvio Savarese. ObjectNet3D: A large scale database for 3D object recognition. In *European Conference Computer Vision (ECCV)*, 2016.

[50] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In *Robotics: Science and Systems (RSS)*, 2018.

[51] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[52] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3D bounding box estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[53] Yinqiang Zheng, Yubin Kuang, Shigeki Sugimoto, Kalle Astrom, and Masatoshi Okutomi. Revisiting the PnP problem: A fast, general and optimal solution. In *International Conference on Computer Vision (ICCV)*, 2013.

[54] Xingyi Zhou, Arjun Karpur, Linjie Luo, and Qixing Huang. Starmap for category-agnostic keypoint and viewpoint estimation. In *European Conference on Computer Vision (ECCV)*, 2018.