Object Proposals Estimation in Depth Images Using Compact 3D Shape Manifolds

Shuai Zheng[1], Victor Adrian Prisacariu[1], Melinos Averkiou[2], Ming-Ming Cheng[1,5], Niloy J. Mitra[2], Jamie Shotton[3], Philip H. S. Torr[1], Carsten Rother[4]

[1]University of Oxford[†], [2]University College London[‡], [3]Microsoft Research, [4]TU Dresden, [5]Naikai University [§]

Abstract. Man-made objects, such as chairs, often have very large shape variations, making it challenging to detect them. In this work we investigate the task of finding particular object shapes from a single depth image. We tackle this task by exploiting the inherently low dimensionality in the object shape variations, which we discover and encode as a compact shape space. Starting from any collection of 3D models, we first train a low dimensional Gaussian Process Latent Variable Shape Space. We then sample this space, effectively producing infinite amounts of shape variations, which are used for training. Additionally, to support fast and accurate inference, we improve the standard 3D object category proposal generation pipeline by applying a shallow convolutional neural network-based filtering stage. This combination leads to considerable improvements for proposal generation, in both speed and accuracy. We compare our full system to previous state-of-the-art approaches, on four different shape classes, and show a clear improvement.

1 Introduction

Object detection has recently undergone significant advances, thanks to progress in GPU design [23], deep convolutional neural networks (ConvNets) [27, 38, 14], and big image recognition dataset [10] collected by e.g. Amazon Mechanical Turk. However, man-made objects, such as chairs, often have very large shape variations, making them still challenging to detect. On the other hand, there is a large number of CAD models available in 3D Warehouse. In this work we want to thoroughly analyse how to leverage this significantly large CAD model collections for the task of finding particular object shapes in a single depth image.

Most object detection approaches have focused on the 2D domain, with 3D being considered only recently, in works such as [16, 34, 3]. Gupta *et al.*[16] is an example of using a standard ConvNet pipeline. The authors use manually annotated RGB-D data from the NYU dataset to train a deep convolutional neural network for feature extraction and classification. At inference time, they classify only the proposed object locations

[†] This work has been supported by UK EPSRC EP/I001107/2 and EP/J014990 (VAP).

[‡] This work has been supported by Starting Grant SmartGeometry (StG-2013-335373) and Melinos Averkiou is grateful for a scholarship from the Rabin Ezra Scholarship Trust.

[§] This work has been partially supported by Youth Leader Program of Nakai University.

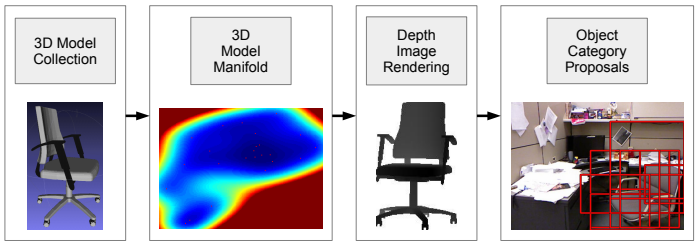

Fig. 1. System overview. Given a set of 3D shapes, we learn a low dimensional latent shape space using GP-LVM (3D Model Manifold). We then generate shapes from this space and render them from a number of random 3D poses. We use these to train a three layer proposal pipeline, based on SVM and ConvNets.

returned from the accurate (but slow) proposal generator of [2]. An alternative approach is presented in [34, 3], where the authors take a collection of 3D CAD models which they use to generate synthetic depth maps. Gupta *et al.* [15] used a convolutional neural network to predict the coarse pose of the object and then align the CAD models to the objects through a model fitting. In that work, however, object proposals are found using an exhaustive standard sliding window proposal generator.

A hallmark of current 3D object detection work is the focus on the classification/feature discovery phase. Finding object location proposals still uses standard 2D strategies, such as selective search or sliding window. In contrast, in this paper we explicitly tackle the problem of proposal generation, and exploit the inherently lower appearance variance of the 3D domain to provide a method that is both faster and more accurate than the current state of the art. Inspired by the work of Karpathy *et al.* [19], we show how to use a compact 3D shape space in detecting those objects with high shape variations.

In the 2D (RGB) domain, objects have often a large appearance variance, due to colour, texture and varying lighting conditions. These, however, do not manifest themselves in the 3D depth domain, which has enabled works like [31] to use primarily synthetic data for training. Inspired by such methods, we do not assume the existence of a large quantity of manually labelled training data, but instead interpolate between manually constructed 3D models, using a variance-preserving approach. We start from a collection of 3D models, obtained from the Trimble Google 3D warehouse. We use these to train a low dimensional latent space using the Gaussian Process Latent Variable Models (GL-LVM, [26]) method. Such spaces capture the intrinsic variance of the training data and have been used previously as shape priors for 3D tracking and reconstruction in [30] and semantic SLAM in [9]. Next we generate 3D shapes back from these spaces and finally render them into multiple 2.5D depth-only projections.

A second requirement for a proposal generator is fast and accurate inference. With this in mind, we train a cascaded object proposal method, comprising of *two* layers. The first is a traditional "objectness" proposal generator such as BING [7] or edgeBox [40], which are the fastest ones. We use this to generate a large number (over 1000) of low accuracy proposals, very quickly, at over 1000 fps. The last layer then is designed to filter out the noise and retain only a small number (about 100) of very accurate proposals. This is constructed using a shallow ConvNet and a linear SVM classifier.

As shown in Figure 1, the output of this cascade is a set of proposals that can be classified by any downstream classifier, e.g. ConvNet [27, 16]. This work therefore proposes a novel method for finding object location proposals, specific to the 3D depth domain. Using our test data, as outlined in the results section, the standard 2D selective search method result in an accuracy of 56.3%, using 100 proposals, while ours has an accuracy of 82.9%. Furthermore, whereas selective search required over 2.6 seconds per frame, our approach needs 0.88 seconds, giving a relative speed up of almost $3\times$. The improvement in accuracy and speed comes as a result of our two **main contributions**:

- We leverage the generative abilities of GP-LVM shape spaces, coupled with a random pose rendering stage, to generate effectively *infinite* amounts of shape variance-maintaining training data.
- We improve the standard 3D object category proposal pipeline, by integrating a proposal generator with a shallow ConvNet-based filtering stage. This leads to considerable improvements in both speed and accuracy of the proposal generation.

2 Related Work

We review related approaches for proposal generation, along with methods that use synthetic data for depth-based inference.

Object proposal methods have been developed to find a small number (e.g. 1,000) of category-independent bounding box candidates that are expected to cover all objects in an image [1, 12]. Such pruning methods are extremely effective in object detection, as demonstrated in recent state-of-the-art approaches [14]. One category of object proposal methods [11, 6] uses rough segmentations to generate the object candidates. While such methods successfully reduce the search space for category-based classifiers, they are computationally very expensive, requiring 2-7 minutes to process a single image. Alexe *et al.* [1] developed an efficient method that integrates several objectness cues to predict the object candidates. Zhang *et al.* [39] proposed a cascaded ranking SVM approach with orientated gradient features to generate the object proposals. More recently, Uijlings *et al.* [38] proposed a selective search method that achieves higher recall prediction. The method, when integrated with an SVM classifier, has been demonstrated to achieve state-of-the-art performance in object detection. Recently, Cheng *et al.* [7], proposed a very fast cascaded SVM method that generates object proposals at over 300 fps. Zitnick *et al.* [40] use edge detection to generate reliable and relative fast proposals. Arbeláez *et al.* [2] develop a multiscale combinatorial grouping method which can provide very accurate segmentation proposals. Krähenbühl *et al.* [22] use a method to identify critical level sets in geodesic distance transforms computed for seeds placed in the image, based on which they generate a lot of reliable segmentation proposals.

Synthetic data has been used for object detection in two primary ways. One is to learn multi-view priors for object detectors from 3D models [21, 29]. The other is to use transfer learning [8] to train a detector using the 3D model data in the 3D domain and use it in 2D images. Generating realistic RGB data from 3D models, however, is very difficult, as it requires realistic 3D shapes, textures, poses, and lighting. Related approaches have been used, for example, in model-based hand 3D tracking by [35, 24, 37]. Fortunately, in the context of depth images, rendering realistic synthetic depth is

comparatively much easier, as it only requires realistic 3D models and pose. Such an approach was used successfully in detection based human pose estimation by [31].

Song *et al.* [34] and Aubry *et al.* [3] developed exemplar-based 3D object detectors trained on 3D CAD datasets. Our approach differs from theirs, as they explore 3D object detection with a sliding window whereas we propose a data-driven object category proposal generator. Our approach is complementary to Gupta *et al.* [16], who use the region-based convolutional neural network [14] framework to learn rich features for 3D object detection, and have achieved very high accuracy in 3D object detection. We leverage publicly available 3D CAD models to improve both the speed and quality of the object category proposal generators, which is the bottleneck of their system.

Another defining feature of our approach is the use of dimensionality reduction for variance-maintaining shape interpolation. This has been used before for e.g. 3D tracking and reconstruction, in e.g. [30, 9], but, to our knowledge, has not yet employed in object proposal generation. Dimensionality reduction in detection has so far primarily targeted training data preconditioning, by removing unnecessary variance from local descriptors in e.g. [20, 5, 33], thereby leading to improved final results.

In this work we follow [31] and use synthetic depth generated from a collection of 3D models to train a detector. We learn low dimensional GP-LVM shape manifolds. We then sample the explicit shape manifolds to generate low variance 3D shapes, which we use to synthetically generate several depth images from multiple views. These are next used to train a fast SVM object category proposal generator method, similar to [7].

3 Algorithm

We propose an algorithm for generating category proposals for single view depth images, that is specialised in handling a particular shape family such as e.g. chairs, monitors, toilets, or sofas. The algorithm runs in three main stages: starting from a set of object models, we construct a corresponding shape manifold to model the in-category variations (§3.1); we then sample the extracted manifold to create representative shapes that are then used to synthetically produce depth images (§3.2); and finally we use the synthetic depth images to train a cascaded proposal generator (§3.3).

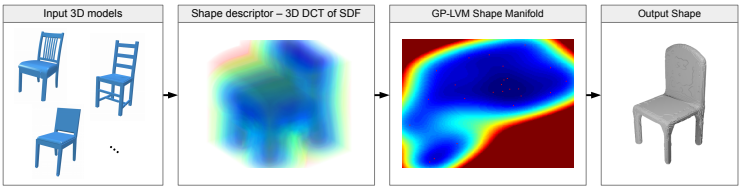

Fig. 2. 3D Parametrised Manifold: given an unorganised 3D chair model collection we build shape descriptors and learn low dimensional embeddings which we use to remove unnecessary shape and training dataset variance.

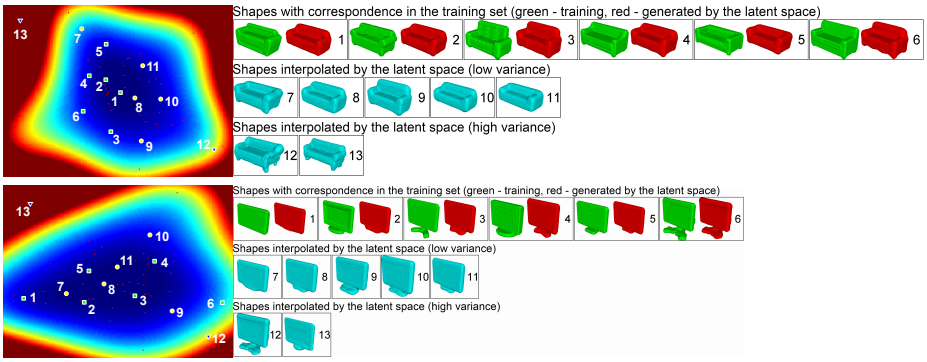

Fig. 3. Example Latent Shape Space: Each row shows a two dimensional latent space of 3D shapes (left) and samples from it (right). Warmer colours indicate higher variance, colder colours lower variance and the red dots point out the latent points corresponding to the training data. Red shapes are generated by the latent space and have the green shapes as ground truth. Blue shapes are interpolated by the latent space, with no correspondence in the training data.

3.1 Constructing a 3D Shape Manifold

We learn Gaussian Process Latent Variable Models (GP-LVM, [25]) shape spaces [9, 30], using the pipeline outlined in Figure 2. In §4, we show how the access to parametrised shape manifolds improves the object category proposal generation, leading to a performance that is superior to several state-of-the-art alternatives.

We assume a given set of training 3D models from the Google Warehouse. These are then aligned (using ICP), voxelised to a volumetric representation, embedded inside 3D signed distance functions, and compressed using the 3D discrete cosine transform.

We next apply GP-LVM on the DCT-SDF descriptor to find a low dimensional shape embedding space. GP-LVM is a nonlinear and probabilistic dimensionality reduction technique. It is used to represent a set of N high dimensional observations $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$ with a set of corresponding low dimensional points $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, where the dimensionality of $\mathbf{X}$ is (much) smaller than that of $\mathbf{Y}$. In our case the observation variables are the DCT compressed SDF volumes, so we can write:

$$\mathbf{y}_i = \text{DCT}_{3\text{D}}(\text{SDF}(M_i)) \qquad M_i = H_{\text{e}}(\text{IDCT}_{3\text{D}}(\mathbf{y}_i)) \tag{1}$$

where M_i is the volumetric representation of the i-th 3D shape, H_{e} is the smooth Heaviside function, SDF computes a signed distance function, and DCT/IDCT are the forward and reverse discrete cosine transforms. Figure 3 shows an example 2D latent space embedding 3D shapes of chairs. We use $256 \times 256 \times 256$ 3D volumes and $40 \times 40 \times 40$ 3D DCT harmonics, for a 64000D final shape descriptor.

Finding a GP-LVM embedding is done by maximising the probability of the observation data $\mathbf{Y}$ jointly given the latent variables $\mathbf{X}$ and the hyperparameters of a Gaussian Process (GP) [25] mapping $\mathbf{Y}$ into $\mathbf{X}$. This probability is formally written as:

$$P(\mathbf{Y}|\mathbf{X}) = \prod_{i=1}^{N} \mathcal{N}(\mathbf{y}_i|0, \mathbf{K}) \tag{2}$$

where $\mathbf{K}$ is the covariance matrix of the GP with the following nonlinear kernel:

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \kappa(\mathbf{x}_i, \mathbf{x}_j) = \theta_1 e^{-\frac{\theta_2}{2}||\mathbf{x}_i - \mathbf{x}_j||^2} + \theta_3 + \theta_4 \delta_{ij} \tag{3}$$

with θ_{1-4} being the GP hyperparameters, δ_{ij} Kronecker's delta function and $\kappa(\cdot, \cdot)$ the GP covariance function. This model generates 3D shapes $\mathbf{y}_i$ from latent variables $\mathbf{x}_i$ as Gaussian distributions:

$$\mathbf{y}_i|\mathbf{X} \sim \mathcal{N}(\mu_i, \sigma_i^2) \tag{4}$$

$$\mu_i = \kappa(\mathbf{x}_i, \mathbf{X})\mathbf{K}^{-1}\mathbf{Y} \tag{5}$$

$$\sigma_i^2 = \kappa(\mathbf{x}_i, \mathbf{x}_i) - \kappa(\mathbf{x}_i, \mathbf{X})\mathbf{K}^{-1}\kappa(\mathbf{x}_i, \mathbf{X})^T. \tag{6}$$

Identifying unusual shapes when using a GP-LVM shape space simply amounts to generating all training shapes back from the latent space and sorting them by variance, with the lowest variance corresponding to the most typical 3D models.

3.2 Depth Rendering and Data Synthesis

Real world objects have different shapes and can be placed in different poses, with different camera viewpoints. This leads to a very large possible appearance space, whose variability we need to deal with. Following Shotton *et al.* [31], we build a randomised depth image rendering pipeline based on the extracted 3D model manifold. Thus we generate a large number of depth images, from different viewpoints and with the object in different poses and displaying intrinsic shape variations. When rendering the depth images (the shapes M_i described above are converted to meshes), we randomly sample the set of 3D appearance parameters using a heuristic approximation of the variability we expected to observe in the real world. Also, in order to make our data more realistic, we use the intrinsic parameters used in NYU V2 data.

3.3 Cascaded Object Category Proposal Generator

Our depth-based object category proposal generator draws inspiration from recent object proposal generators, such as BING [7], EdgeBox [40], and ConvNet-based object detection approaches, such as [14]. We suggest a two-layer structure. The first layer follows the object proposal generator. At inference time, these produce a large number of detections very quickly (at over 1000 fps). Precision however can often be quite low. The second layer is then designed to remove some false positives and so reduce the number of proposals needed for an accurate detection from e.g. 1000 to e.g. 100, with little to no loss of recall. This layer is implemented using a shallow ConvNet.

Unlike in RGB images, the object contour information is very salient in depth images. One way to detect such contours is to use gradient convolution filters. This led us to adapt the 64D normalised gradients feature used for *2D RGB* object category proposal estimation in [7], to our depth-only scenario. In our proposed framework, we can also use a more accurate proposal generator approach, such as EdgeBox [40].

Fig. 4. Shallow ConvNet model architecture. This figure presents our inference process during inference time. We adapted successful network configuration described in [28] for this task.

Refining the Object Category Pool The first stage of our proposal generating cascade is very fast, but often leads to low quality proposals. In order to refine the proposal pool, we use a shallow 4-layer ConvNet and a following linear SVM, as shown in figure 4.

We trained the ConvNet first on the ImageNet dataset and next fine-tuned it on the NYU V2 depth training data and the synthetic data generated from the GP-LVM low dimensional latent space. Using both real and artificial examples prevents the network from overfitting. Compared to the standard deep networks [18] used in the ImageNet object detection task, our network is shallower than the deep networks [18, 36], while having lower accuracy, is faster at run-time, making it better suited for the task at hand.

4 Experiments

We evaluate our method on the NYU V2 [32] dataset using four categories of objects (chairs, sofas, toilets and TV). The remainder of this section is split into four parts: (i) §4.1 describes our experimental setup; (ii) §4.2 shows that using our variance-preserving synthetic data and random view rendering improves accuracy; (iii) §4.3 shows that the extra ConvNet filtering further improves accuracy.

4.1 Experimental Setup

Dataset. The NYU V2 dataset [32] contains 1449 RGB and depth images with pixel-level segmentation annotations. We split data set according to the standard NYU V2 train/test split to obtain 495 training and validation images, and 404 testing images.

To train our latent space and classifiers we also use 3D models downloaded from the Google 3D Warehouse. We select 374 chairs, 42 TV, 36 sofas, and 24 toilets 3D CAD models. After passing through the latent space filtering, we use them to render the depth images, 37400 for the chair class, 25200 for TV, 21600 for sofa, and 14400 for toilet. Here we considered 600 different random pose and viewpoint configurations, which excluded the top and bottom viewpoints, as these are rarely seen in indoor scenes.

Evaluation Criteria. We use standard DR-#WIN accuracy measure [1], which quantifies detection rate (DR) given #WIN proposals. A proposal covers an object if the strict VOC [13] criterion is satisfied, i.e. if INT-UNION> 0.5.

Implementation. Our approach is implemented based on the Caffe [17] library. We fine-tune the network-in-network model [28] on the NYU V2 dataset and the synthetic data. The learning step size is 5000, the momentum is 0.9, and the weight decay is

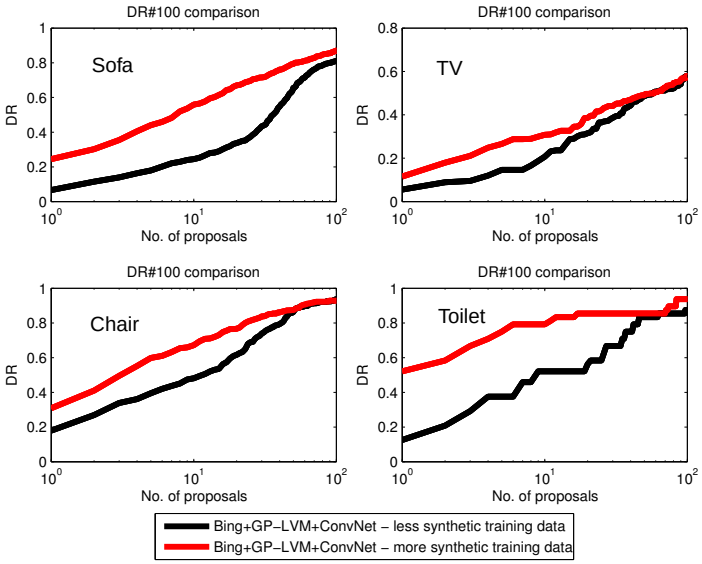

Fig. 5. Logarithmic plot measuring the DR#100 (50% Intersection-over-union) accuracy when we double the amount of synthetic data (i.e. sofa: 43200 (red), 21600 (black); TV: 50400 (red), 25200 (black); chair: 74800 (red), 37400 (black); tiolet: 28800 (red), 14400 (black).). The extra data leads to much higher accuracy when using fewer proposals. As the number of proposals increases the extra training points do not help much, as the BING discrimination ability saturates.

0.0005. Using a NVIDIA Titan Black GPU, the per frame inference processing is 0.88 seconds per frame, with a pool of 1000 category proposal candidates from BING [1].

4.2 Synthetic Data

We first investigate the effect of the *number of rendered depth images* on the classification result. As shown in the Figure 5, we observe the clear trend that adding more synthetic renderings into the training set helps boost the performance of the approach.

We next investigate the effect of the *data preconditioning* (i.e. the number of GP-LVM dimensions and variance of the 3D shapes) on the classification result. Not all object 3D models are realistic and not all 3D shape details are important in the classification. The GP-LVM shape manifold allows us to remove both unusual shapes and unnecessary intra-shape variance from the training data. To showcase this feature we used 374 3D chair shapes to train 3, 4, 5, 6 and 7 dimensional GP-LVM shape spaces. The results are shown in Table 1 (left and right) and Table 2. In Table 1 (left) we show the results obtained from the original training set (i.e. not compressed with GP-LVM) and when learning a 5D GP-LVM latent space and training with (i) the top 100, 150 and 200 shapes with the lowest variance, and (ii) the full training set. Initially, as the

[1] https://github.com/bittnt/Objectness

number of training shapes increases (between 100 and 150 models) accuracy improves. At some point between 150 and 200 though unusual shapes start being added to the training set, which decreases the accuracy. In Table 1 (right) we vary the number of dimensions used for the trained GP-LVM spaces from 3 to 7. The same trend as in Table 1 (left) can be observed. Initially (when using between 3 and 5 dimensions) we add useful shape variance to the training set which improves the final accuracy. When more than 5 dimensions are used (and that includes the full uncompressed training set) we add unnecessary variance to the training set thus decreasing accuracy. Finally, in Table 2 [2], we use the chair class and 1000 BING proposals to evaluate our method of shape generation (BING+GPLVM) against (i) BING + the alternative shape generation method of ShapeSynth [4] and (ii) other methods for proposal generations that do not use synthetic data: BING [7], OBN [1], CSVM [39], SEL [38], and random guessing. OBN, CSVM and BING are trained on the NYU V2 training set whereas SEL does not require any data. BING+ShapeSynth and BING+GPLVM are trained on the NYU V2 training data and the sampled ShapeSynth or GPLVM synthetic shapes. Our method outperforms all the other proposal generators. Of particular note is that BING+GP-LVM outperforms BING+ShapeSynth, in spite of ShapeSynth usually generating much more realistic looking shapes. This result complements the experiment from Table 1 and shows that sampling only low variance shapes from the manifold is beneficial.

BING-GPLVM		BING-GPLVM	
#DIM(#SAM)	DR-#1000W	#DIM(#SAM)	DR-#1000W
5-100	88.7	3-150	88.0
5-150	**89.7**	4-150	89.6
5-200	89.2	**5-150**	**89.7**
5-374	88.4	6-150	89.3
original-374	87.8	7-150	89.2

Table 1. Effect of data preconditioning. Left - accuracy results obtained when training on the chair category with the original dataset not compressed with GP-LVM and when using a 5D low dimensional GP-LVM space and training with (i) the top 100, 150 and 200 shapes with the lowest variance (ii) the full training set. Right - accuracy obtained when generating data from 3-7D low dimensional spaces and selecting the top 150 shapes with the lowest variance. #DIM indicates the number of latent space dimensions and #SAM the number of samples from each latent space.

4.3 ConvNet Filtering Layer

In Figure 6 we compare selective search, BING+GP-LVM and BING+GP-LVM+ConvNet, when using *only 100* proposals, and all four object classes. The best results are obtained when using our full approach (BING+GP-LVM+ConvNet), with BING+GP-LVM following with 59.2% respectively, and selective search being the last with 56.3%. Of course both selective search would reach higher accuracy with more proposals, as

[2] Experiments are carried out on a machine with a Intel Xeon E5-2687w(32 Cores).

Method	Random Guess	OBN [1]	CSVM [39]	SEL [38]	BING [7]	BING-ShapeSynth [4]	BING-GPLVM our approach
DR-#1000W	42.0	83.0	84.5	85.9	85.6	88.5	89.7
Time (seconds)	N/A	2.10	1.20	2.6	**0.0009**	**0.0009**	**0.0009**

Table 2. Quantitative results on different proposal estimation approaches. We compared different approaches on the chair category using the NYU V2 depth image dataset. We follow the standard evaluation criteria, which is the detection rate over 1000 object proposals [1]. The best result are obtained when using BING+GP-LVM.

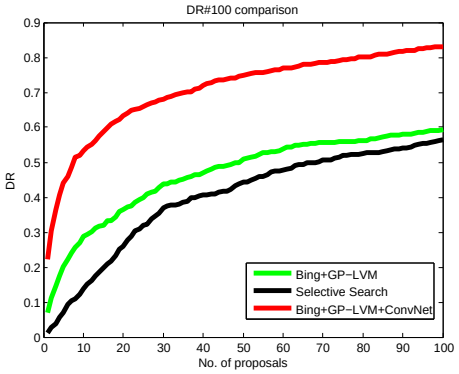

Fig. 6. DR#100 Comparison. We compare three methods, selective search (which we consider to be the state of the art), BING+GP-LVM, and BING+GP-LVM+ConvNet, when using *only 100* proposals, and all four object classes. Our full approach is the most accurate, with an accuracy of 82.9% whereas selective search produces 56.3%.

shown before. However, no method other than BING+GP-LVM+ConvNet is able to reach this level of accuracy with *just 100 proposals*. We also note that per-frame processing with our full approach was 0.88s, whereas for selective search it was 2.6s.

5 Conclusions

We presented an algorithm for generating depth-based proposals for high-variation specific object categories. Our main message is that (i) the use of synthetic data, sampled from variance-maintaining compact shape manifolds, boosts the accuracy of object category proposal estimation, as it enables the classifier to focus on the intrinsic 'classness' variance and ignore unusual shape details; and (ii) a final shallow ConvNet layer further dramatically improve the overall accuracy. As future work, we intend to investigate the use of this proposal generator in various applications, such as depth fusion or 3D reconstruction.

References

1. Alexe, B., Deselaers, T., Ferrari, V.: Measuring objectness of image windows. IEEE Trans. Pattern Anal. Mach. Intell. 34(11), 2189–2202 (2012)
2. Arbeláez, P., Pont-Tuset, J., Barron, J., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: CVPR. pp. 328–335 (2014)
3. Aubry, M., Maturana, D., Efros, A.A., Russel, B., Sivic, J.: Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of CAD models. In: CVPR. pp. 3762–3769 (2014)
4. Averkiou, M., Kim, V., Zheng, Y., Mitra, N.J.: Shapesynth: Parameterizing model collections for coupled shape exploration and synthesis. Comput. Graph. Forum 33(2), 125–134 (2014)
5. Brown, M., Hua, G., Winder, S.: Discriminative learning of local image descriptors. IEEE Trans. Pattern Anal. Mach. Intell. 33(1), 43–57 (2011)
6. Carreira, J., Sminchisescu, C.: CPMC: Automatic object segmentation using constrained parametric min-cuts. IEEE Trans. Pattern Anal. Mach. Intell. 34(7), 1312–1328 (2012)
7. Cheng, M.M., Zhang, Z., Lin, W.Y., Torr, P.: BING: Binarized normed gradients for objectness estimation at 300fps. In: CVPR. pp. 3286–3293 (2014)
8. Chiu, H.P., Kaelbling, L.P., Lozano-Perez, T.: Virtual training for multi-view object class recognition. In: CVPR. pp. 1–8 (2007)
9. Dame, A., Prisacariu, V.A., Ren, C.Y., Reid, I.: Dense reconstruction using 3d object shape priors. In: CVPR. pp. 1288–1295 (2013)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: CVPR. pp. 248–255 (2009)
11. Endres, I., Hoeim, D.: Category independent object proposals. In: ECCV. pp. 575–588 (2010)
12. Endres, I., Hoiem, D.: Category-independent object proposals with diverse ranking. IEEE Trans. PAMI pp. 222–234 (2014)
13. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. International Journal of Computer Vision 88(2), 303–338 (Jun 2010)
14. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR. pp. 580–587 (2014)
15. Gupta, S., Arbeláez, P.A., Girshick, R.B., Malik, J.: Aligning 3D models to RGB-D images of cluttered scenes. In: CVPR. pp. 4731–4740 (2015)
16. Gupta, S., Girshick, R., Arbelaez, P., Malik, J.: Learning rich features from rgb-d images for object detection and segmentation. In: ECCV. pp. 345–360 (2014)
17. Jia, Y.: Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/ (2013)
18. Karen Simonyan, A.Z.: Very deep convolutional networks for large-scale image recognition. In: arXiv:1409.1556v2 (2014)
19. Karpathy, A., Miller, S., Li, F.F.: Object discovery in 3d scenes via shape analysis. In: ICRA. pp. 2088–2095 (2013)
20. Ke, Y., Sukthankar, R.: PCA-SIFT: A more distinctive representation for local image descriptors. In: CVPR. pp. 506–513 (2004)
21. Kim, Y.M., Mitra, N.J., Huang, Q., Guibas, L.: Guided real-time scanning of indoor objects. Computer Graphics Forum (Proc. Pacific Graphics) 32, 177–186 (2013)
22. Krähenbühl, P., Koltun, V.: Geodesic object proposals. In: ECCV. pp. 725–739 (2014)
23. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NIPS. pp. 1106–1114 (2012)
24. de La Gorce, M., Paragios, N., Fleet, D.: Model-based hand tracking with texture, shading and self-occlusions. In: CVPR. pp. 1–8 (2008)

25. Lawrence, N.: Probabilistic Non-linear Principal Component Analysis with Gaussian Process Latent Variable Models. JMLR 2005 6, 1783–1816
26. Lawrence, N.D.: Gaussian process latent variable models for visualisation of high dimensional data. In: NIPS. pp. 329–336 (2003)
27. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE. pp. 2278–2324 (1998)
28. Lin, M., Chen, Q., Yan, S.: Network in network. In: ICLR (2013)
29. Pepik, B., Stark, M., Gehler, P., Schiele, B.: Multi-view priors for learning detectors from sparse viewpoint data. arXiv:1312.6095 (2014)
30. Prisacariu, V.A., Segal, A., Reid, I.: Simultaneous monocular 2D segmentation, 3D pose recovery and 3D reconstruction. In: ACCV. pp. 593–606 (2013)
31. Shotton, J., Fitzgibbon, A.W., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: CVPR. pp. 1297–1304 (2011)
32. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: ECCV. pp. 746–760 (2012)
33. Simonyan, K., Vedaldi, A., Zisserman, A.: Learning local feature descriptors using convex optimisation. IEEE Trans. Pattern Anal. Mach. Intell. pp. 1573–1585 (2014)
34. Song, S., Xiao, J.: Sliding shapes for 3D object detection in depth images. In: ECCV. pp. 634–651 (2014)
35. Stenger, B., Thayananthan, A., Torr, P.H.S., Cipolla, R.: Model-based hand tracking using a hierarchical bayesian filter. IEEE Trans. Pattern Anal. Mach. Intell. 28(9), 1372–1384 (2006)
36. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: arXiv:1409.4842 (2014)
37. Tang, D., Yu, T.H., Kim, T.K.: Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In: ICCV. pp. 3224–3231 (2013)
38. Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M.: Selective search for object recognition. IJCV 104(2), 154–171 (2013)
39. Zhang, Z., Warrell, J., Torr, P.H.: Proposal generation for object detection using cascaded ranking svms. In: CVPR. pp. 1497–1504 (2011)
40. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: ECCV. pp. 391–405 (2014)