Object Discovery in Depth Images

Tzu-Wei Huang*, Yu-An Wei*, Hwann-Tzong Chen* and JenChi Liu†
* Department of Computer Science
National Tsing Hua University, Taiwan
† Intelligent Vision System Division
Electronic and Optoelectronic System Research Laboratories, ITRI, Taiwan

***Abstract*—We present an unsupervised method for discovering objects from depth information. Our method can identify new common objects appearing in different depth images. We use 2D bounding box proposals to detect candidate locations of objects in each depth image, and then retrieve the corresponding 3D bounding boxes using the depth information. Invalid object proposals can be further removed by analyzing the point cloud distribution inside the 3D bounding box. We measure the similarity between each pair of the object proposals in different images to identify co-occurrences of the same instance. The similarity measure is automatically learned by a Siamese convolutional neural network. Our method is unsupervised in a sense that we do not need human labeled data to train the Siamese network. We use 3D CAD models to synthesize a large set of similar and dissimilar pairs of depth images as the positive and negative data. Our experiments on synthetic data show that the proposed method is able to discover the co-occurrences of the common objects in different depth images.**

I. Introduction

The task of unsupervised object discovery is to recognize instances or categories of objects based on their multiple occurrences across images. Different from object recognition and object detection, which require training examples of the 'to-be-classified' objects, methods for unsupervised object discovery do not assume that the object categories of interest have to be seen or learned before. Instead, the methods themselves should explore and find out where the salient objects are and and how they associate with each other across images.

The pioneering work of Sivic *et al.*[13] on unsupervised object discovery aims to answer the research question that 'Is it possible to learn visual object classes simply from looking at images?' They use probabilistic methods to model the instances of categories as mixtures of topics. Objects are represented using SIFT-based visual words. An earlier survey of unsupervised object discovery literature is available in [15].

Recent methods of object discovery often incorporate additional settings characterizing specific scenarios. For example, unsupervised object discovery can be achieved jointly with object segmentation for Internet images. The method of Rubinstein *et al.*[12] can segment out the common objects from large image collections. Kwak *et al.*[7] combined the task of object discovery with object tracking for video data. They used region matching to associate objects in different videos and used object tracking to single out good candidates with each video. Doersch *et al.*[3] used the context as a supervisory cue for discovering objects. The information of visual saliency [4] or region proposals [2] can also be used for object discovery .

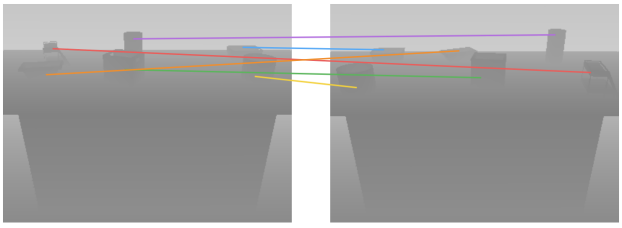

Fig. 1. Can you identify the same objects in the two depth images?

As mentioned in [12], object discovery is closely correlated to the problem of co-segmentation [5], [11], [16]. Many strategies can be shared for solving both tasks. Another related research topic is zero-shot learning [8], [9] of which the tasks are to learn to class data when labeled training data are not available. Zero-shot learning is usually achieved by cultivating semantic features or attributes. Our method also does not rely on directly labeled data for training. We use 3D CAD models [14], [17] to generate automatically a lot of depth images of similar and dissimilar object pairs as positive and negative examples for learning a similarity measure. The test data can contain object instances and categories that are never seen before.

A. Our Approach

This paper aims to address the problem of discovering objects in depth images. Given a set of depth images, we seek to identify and localize the same 3D objects in different images. Fig. 1 shows an example of finding and matching 3D objects in two depth images. We propose an unsupervised method for solving this problem. While prior work of Karpathy *et al.*presents a method for discovering object from 3D colored meshes [6], our method only needs the 2.5D depth maps, which is much easier to acquire than 3D color meshes. Our method can localize and match new common 3D objects in different depth images. The objects may have different poses in the depth images, and the depth images may correspond to different 3D scenes viewed from different angles.

Our approach consists of two parts. The first part is to detect candidate object locations. We use 2D bounding box proposals to detect candidate locations of objects in each depth image, and then retrieve the corresponding 3D bounding boxes according to the depth information. We may further remove

invalid object proposals based on the point cloud distribution within the 3D bounding box. The second part of our approach is to identify co-occurrences of the same instance in different depth images. We use 3D CAD models [17] to synthesize a large number of similar and dissimilar pairs of depth images. Each pair of depth images might contain the same 3D object or two different 3D objects with different poses observed from different viewing angles. We use such kinds of data to train a Siamese convolutional neural network as a similarity measure. Note that although our method employs discriminative deep learning techniques for deriving the similarity measure, our approach can still be considered an unsupervised method in a sense that we do not need human labeled data to 'teach' our method how to recognize 3D objects. Our method can automatically learn to discover common unknown objects in depth scenes.

II. Detecting 2D and 3D Bounding Boxes

Object proposal generator is a critical step in the object discovery pipeline. Instead of using CNN-based methods, e.g., Region Proposal Networks [10], to generate a bounding box, we use Edge Boxes [18] to detect 2D bounding boxes first, and then extend each bounding box into a 3D bounding boxes according to the depth information. We replace the original edge detector included in the Edge Boxes method with the Sobel operator, and enhance the edge responses along the x and the y axis. After we obtain the candidate 2D bounding boxes, we crop out the point cloud derived from the depth image to set up a rough 3D bounding box aligned with the camera coordinate system. At this point, the generated 3D object proposals are likely to contain many false positives. We can further check the validity of each proposal by calculating its point-cloud density within the bounding box, similar to the strategy used in [1]. We remove those unreasonable proposals to suppress false positives and reduce the required computational cost for the subsequent step of object matching. In our experiments, we only retain ten percent of the top 500 bounding boxes in each scene.

A. Adjusting the Bounding Boxes

The 3D bounding boxes derived from the aforementioned step are aligned with the camera coordinates. We can use the depth information to estimate the planar areas in the scene, such as the table top or the wall, and obtain the object coordinates. The object coordinate system can be modeled by applying Principal Component Analysis to the surface normal at every point. The surface normals of points on the table should be perpendicular to the table top, so that the normal direction of the table top can be approximated by the main eigen vector of all normals in the scene. After we find the surface normal of the table top, we can rotate each bounding box so that it is aligned with the object coordinate system. Fig. 2 shows some detected bounding boxes after adjustment. We can further perform non-maximum suppression to merge similar bounding boxes, as shown in Fig. 3.

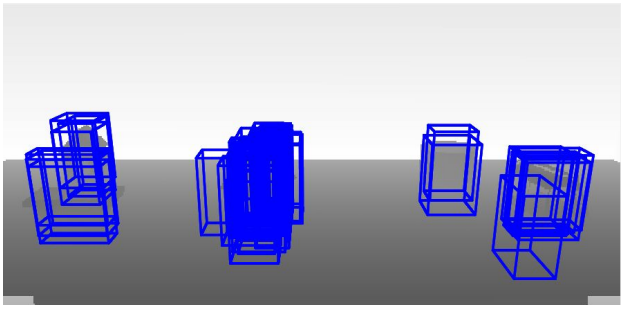

Fig. 2. Detected 3D bounding boxes before non-maximum suppression.

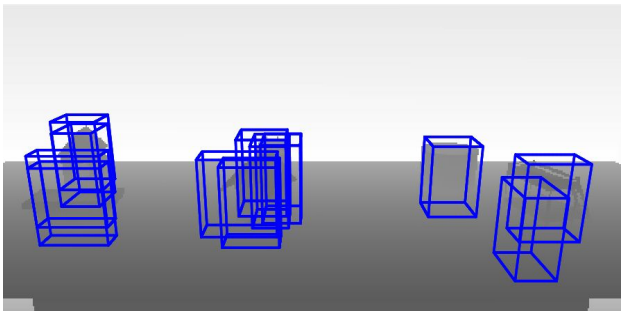

Fig. 3. 3D bounding boxes after non-maximum suppression.

III. Siamese Convolutional Neural Network for Learning the Similarity Measure

Given two candidate bounding boxes in two depth images, we need to compare their similarity based on the depth information, and decide if they contain the same object. To achieve this goal, we create a Siamese convolutional neural network to learn how to compare two depth images. The learning task is formulated as a binary classification problem. The training data are pairs of depth images generated from 3D CAD models. Each positive pair comprises two depth scenes of the same 3D object rendered under different poses and different viewing angles. Each negative pair consists of two depth scenes of two different 3D objects, also rendered with variations. In this way, we are able to train a Siamese network with a larger number of automatically generated data, and the resulting Siamese network can be used as a similarity measure to decide whether two depth images contain the same object.

A. Network Architecture

Our neural network takes a pair of stacked depth images as input. Specifically, the input size is $2 \times 112 \times 112$. The input layer is followed by several convolutional, ReLU, and MaxPooling layers as shown in Fig. 4. As for the loss function, we use mean squared error to measure the difference between the neural network output and the ground-truth label.

B. Network Training

The parameters in the network are initially filled with values sampled from a zero mean, unit variance Gaussian distribution. The learning rate is set to 0.1, and the batch size is 64. We use Stochastic Gradient Descent to optimize the parameters.

Fig. 4. Our network structure: a Siamese convolutional neural network for learning the similarity measure.

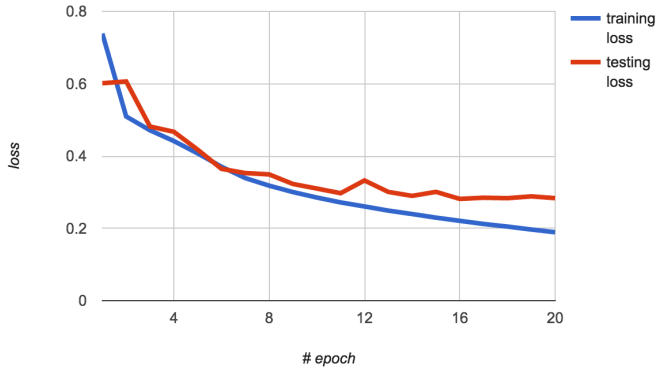

Fig. 5. training the Siamese convolutional neural network. The loss drops quickly after several epochs.

As shown in 7, the training loss drops quickly, and the loss for testing data converges after 20 epochs. An epoch means the entire training data are fully processed by the network once. We use Nvidia TitanX GPU for training, and the time needed for each epoch is roughly 750 seconds.

IV. Synthesizing 3D Objects and Scenes

We mention in the previous section that the Siamese network as a similarity measure is learned from automatically generated 3D depth scenes. In this section We describe the details of the data generation process.

A. Data Selection

We use the 3D CAD models from by ShapeNet [17]. The 3D models are stored in a hierarchical way. We do not consider some classes of objects because their depth variations are just too small when viewed at a certain distance, *e.g.* , flat objects like swords and dishes. We manually select classes whose relative dimensions (width, height, depth) are more similar to a cube. For each selected class, we randomly choose four instances for later experiments. Our ultimate goal is to find the same object instance in two scenes, therefore we discard the class information for each instances. There are total 148 instances. We further take 120 instances for training the Siamese network and the remaining 28 instances are used for testing.

B. Data Preparation

For each of the CAD model instances, we randomly choose ten locations in the scene with constraints so that the object stays in the camera view. Also, 36 rotations are applied to each placement to increase appearance variations. We create

(a) Instances sampled from the training set

(b) Instances sampled from the test set

Fig. 6. Examples of 3D CAD models for generating the training and test data.

data pairs in the following manner. We make $120 \times 120 \times 360/10 = 518400$ pairs for training. To test the matching performance, we create 100 pairs of images. For each pair, we first randomly select object instances from the test set, and place them on a table with random translation and rotation; Using the same selected objects, we repeat the above operation to render another image.

Listing 1. training pair selection scheme

```
while n<maxpairs: #518400
    if n%2==0:
        i = random(len(instances))
        sample1 = SampleFromInstance(i)
        sample2 = SampleFromInstance(i)
        label = makelabel(i,i)
    else:
        i1 = random(len(instances))
        i2 = random(len(instances))
        sample1 = SampleFromInstance(i1)
        sample2 = SampleFromInstance(i2)
        label = makelabel(i1,i2)
    aggregate(sample1, sample2, label)

def makelabel(i1, i2):
    if i1==i2:
        return 1
    else:
        return -1
```

V. Generate Matching Proposals

Given the extracted proposals from each image, we can compute the matched object proposals using the trained Siamese convolutional network. Each object proposal in the first depth image is paired with an object proposal in the second image. Therefore, we have $N_1 \times N_2$ pairs, where N_1

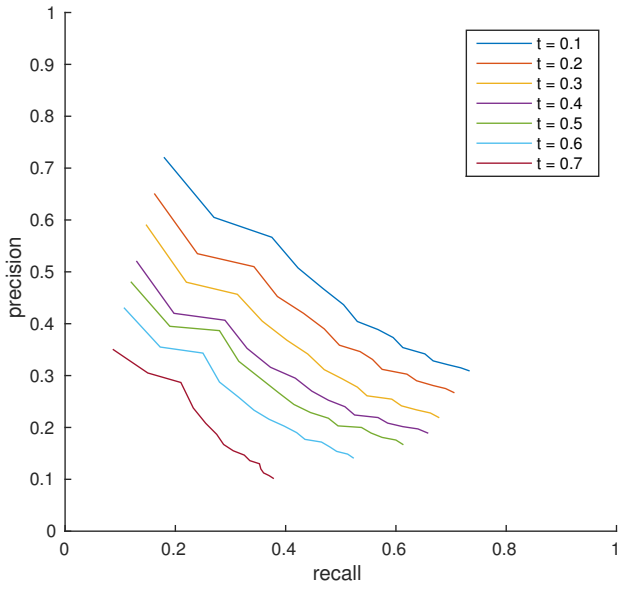

Fig. 7. The precision-recall curves under different matching criteria of the overlapping threshold t.

and N_2 are the numbers of extracted proposals from the two images. Later, we can evaluate the similarity of each pairs of proposals using the trained Siamese convolutional network. The proposals are then sorted according to their matching scores.

VI. Experimental Results

We apply our models to the virtual depth image pairs. Since the goal is to find corresponding objects in the images, we first define what a match is. For each pair of images, we have the ground-truth matching information, which provides pairs of object proposals with the bounding box information (x,y,w,h). We denote each ground-truth correspondence as g_1^i and g_2^i, where i is the i-th pair of object proposal, and the subscript indicates the first and second image of the image pair. Similarly, we denote the matched object proposal pairs as b_1^j, b_2^j, $j = \{1..K\}$, where K is the top K matches according to their matching scores produced by the Siamese network.

Given K selected proposals and the ground-truth object pairs, we can compute the precision and recall rates to evaluate performance. For the proposal pair b_1, b_2 and the ground-truth proposal pair g_1,g_2, if the IOU (intersection over union) of b_1 and g_1 is higher than overlapping threshold t and so is b_1 and g_2, then we call it a true positive match. Otherwise, we count it as a false positive. Therefore we can compute the precision and recall rates for different choices of K. Fig. 7 illustrates the precision-recall curves over different settings of overlapping threshold t. Tables I&II summarize the precision and recall rates of the top K matches for different values of the overlapping threshold t.

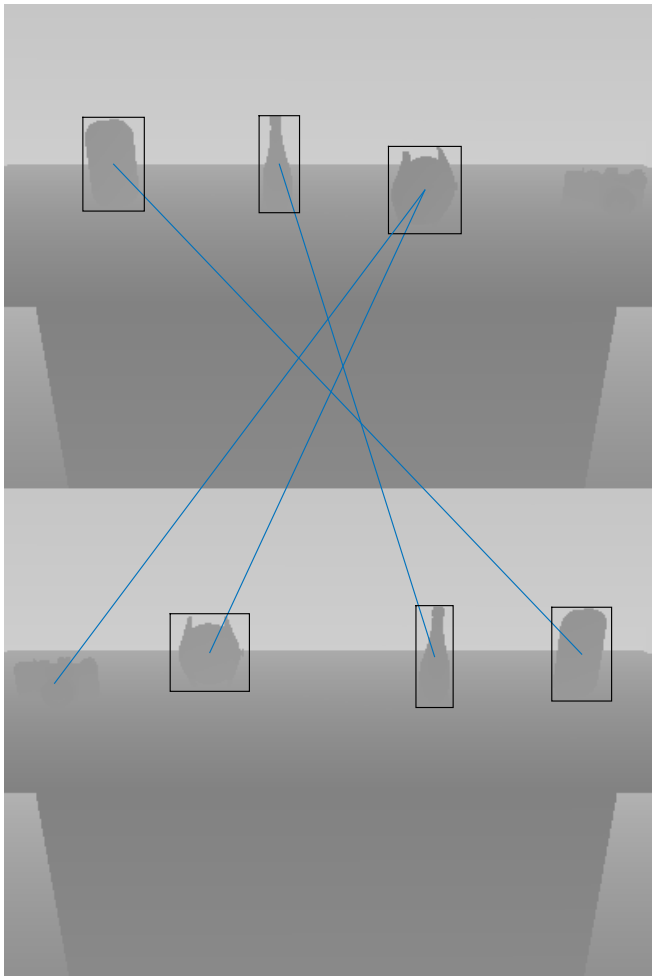

Fig. 8. The lines connect pairs of object proposals that have the highest K scores. The end points of each line are bounding box centers for each object proposal. The black box indicates a true positive proposal whose bounding box has an IOU higher than the overlapping threshold t with respect to the ground-truth bounding box. In this example, K = 4, t = 0.6.

TABLE I
Precision

	The overlapping threshold t							
	t=0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**
K=1	0.72	0.65	0.59	0.52	0.48	0.43	0.35	0.19
K=4	0.51	0.45	0.41	0.35	0.33	0.29	0.24	0.11
K=7	0.40	0.36	0.31	0.27	0.24	0.22	0.17	0.07
K=10	0.35	0.31	0.26	0.22	0.20	0.18	0.14	0.06
K=13	0.32	0.28	0.23	0.20	0.18	0.15	0.11	0.05

TABLE II
Recall

	The overlapping threshold t							
	t=0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**
K=1	0.18	0.16	0.15	0.13	0.12	0.11	0.09	0.05
K=4	0.42	0.39	0.36	0.33	0.32	0.28	0.23	0.11
K=7	0.53	0.50	0.47	0.45	0.41	0.37	0.29	0.13
K=10	0.61	0.57	0.55	0.53	0.49	0.43	0.34	0.15
K=13	0.69	0.67	0.64	0.61	0.57	0.49	0.36	0.16

VII. Conclusions

We have presented our study on how to find plausible objects in a depth image based on 3D bounding box proposals and how to associate them across two depth images using a Siamese network. The preliminary results show that, without relying on color, simply the 2.5D cues derived from depth images can provide useful information for detecting and matching objects. Furthermore, without human supervision, our system is able to learn with generative 3D data and to make sense out of the scenes.

Acknowledgment

This research was supported in part by ITRI grant F101WD2400.

References

[1] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun. 3d object proposals for accurate object class detection. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 424–432, 2015.

[2] M. Cho, S. Kwak, C. Schmid, and J. Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1201–1210, 2015.

[3] C. Doersch, A. Gupta, and A. A. Efros. Context as supervisory signal: Discovering objects with predictable context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III*, pages 362–377, 2014.

[4] S. Frintrop, G. M. García, and A. B. Cremers. A cognitive approach for object discovery. In *22nd International Conference on Pattern Recognition, ICPR 2014, Stockholm, Sweden, August 24-28, 2014*, pages 2329–2334, 2014.

[5] A. Joulin, F. R. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 1943–1950, 2010.

[6] A. Karpathy, S. D. Miller, and F. Li. Object discovery in 3d scenes via shape analysis. In *2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, May 6-10, 2013*, pages 2088–2095, 2013.

[7] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid. Unsupervised object discovery and tracking in video collections. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 3173–3181, 2015.

[8] H. Larochelle, D. Erhan, and Y. Bengio. Zero-data learning of new tasks. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 646–651, 2008.

[9] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell. Zero-shot learning with semantic output codes. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, pages 1410–1418, 2009.

[10] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015.

[11] C. Rother, T. P. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pages 993–1000, 2006.

[12] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 1939–1946, 2013.

[13] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their localization in images. In *10th IEEE International Conference on Computer Vision (ICCV 2005), 17-20 October 2005, Beijing, China*, pages 370–377, 2005.

[14] S. Song and J. Xiao. Sliding shapes for 3d object detection in depth images. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI*, pages 634–651, 2014.

[15] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. L. Buntine. Unsupervised object discovery: A comparison. *International Journal of Computer Vision*, 88(2):284–302, 2010.

[16] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 2217–2224, 2011.

[17] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1912–1920, 2015.

[18] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, pages 391–405, 2014.