Robust Object Recognition Under Partial Occlusions Using an RGB-D Camera

Yong-Ho Yoo and Jong-Hwan Kim

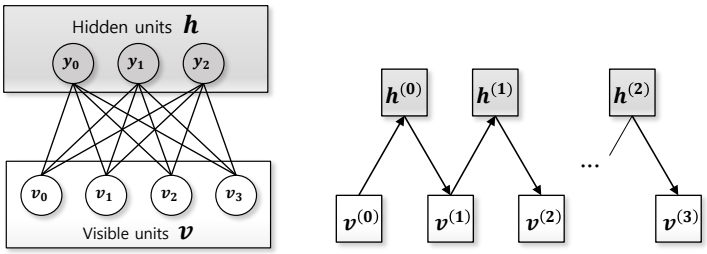
Department of Electrical Engineering, KAIST
291 Daehak-ro, Yuseong-gu, Daejeon, Republic of Korea
{yhyoo,johkim}@rit.kaist.ac.kr

Abstract. For a robot to execute a specific task, the robot firstly has to recognize what objects are in robot's view. To complete a specific task in a given time, the computation time for recognition is also important. There are much research for increasing recognition accuracy, but the recognition speed is not enough to be applied in real environment. On the other hand, there are also much research for reducing the computation time for recognition, but the recognition accuracy needs to be further improved. Nowadays, deep network has come into the spotlight due to its speed and accuracy. Deep network doesn't need to find hand-tuned features. This paper proseses a deep network-based object recognition algorithm. The main contribution is that objects could be recognized under occlusion, as objects are often laid to overlap each other. The occlusion makes object recognition accuracy worse. To overcome this problem, the dataset for training consists of not full images but partial information of images and corresponding ground truths. The object region could be found very quickly by using an RGB-D camera. By assuming that most objects are on the stable plane, object regions are taken easily. Experimental results demonstrate such consideration of contextual information (e.g. objects are on the table) makes the performance of recognition better.

Keywords: object recognition, occlusion, deep learning, deep belief network, RANSAC

1 Introduction

Recently, the progress of an RGB-D camera that provides both color and dense depth information makes a paradigm shift in computer vision. There are many applications using this RGB-D camera, e.g. object recognition [1], people tracking visual odometry and Simultaneous Localization and Mapping (SLAM) [2, 3]. In addition to computer vision, the usage of the RGB-D camera is extended to robotics [4, 5]. When a specific situation is given to a robot, this robot has to understand the situation. This situation is mainly judged by camera images. Existing cameras such as an RGB camera and a tereo camera [6] have limit to estimating the distance between camera and some objects. But, this limitation



(a) A general RBM model. (b) Gibbs sampling in RBM model.

Fig. 1: A general RBM model and Gibbs sampling process in RBM.

has been overcome by the development of depth camera that makes it possible to estimate the exact distance.

For a robot to carry out a given task, the robot has to understand a current place and situation. Objects located in front of the robot's view have to be detected and distinguished. As a scene given to the robot is changing from moment to moment, noticing this change of scenes is essential to increase recognition accuracy and improve recognition speed.

In this paper, we propose robust object recognition under partial occlusions using the RGB-D camera. A plane is easily detected using depth information that are obtained by the RGB-D camera. By assuming objects are located on a plane such as a table, object recognition accuracy could be increased and recognition speed could be faster by taking only pixels located on the plane. In particular, a database is made up of image patches and corresponding ground truths. By making the database as patch-based images and labels, robust object recognition is possible even if there are partial occlusions. A learning algorithm is based on Deep Belief Network (DBN) [7, 8]. In a multi-layer neural network, weights are pre-trained by Restricted Boltzmann Machine (RBM) [12]. Then, weights are fine-tuned by back-propagation. The advantage of DBN is that features are extracted by not hand-crafted method like SIFT [9], SURF [10], but learning. It accelerates feature extraction so that computation cost is considerably reduced.

The remainder of the paper is organized as follows: Section 2 describes some preliminaries such as RANdom SAmple Consensus (RANSAC) [11], RBM and DBN. The procedure of pre-processing and the proposed learning algorithm are detailed in Section 3, and experimental results are presented in Section 4. Finally, concluding remarks follow in Section 5.

2 Preliminaries

2.1 Restricted Boltzmann Machine

As RBM [12] is a special kind of stochastic neural network, it consists of visible units and hidden units. In RBM, there are no connections between units in the

same layer. There are only connections between units in the different layer as shown in Fig. 1(a). RBM is a kind of Markov Random Field so that it can be represented as probabilistic function as

$$p(v,h) = \frac{1}{Z}e^{-E(v,h)} \tag{1}$$

where Z is the partition function and v and h are visible units and hidden units, respectively. And energy function E in Eqn. 1 is defined as

$$E(v,h) = b'v - c'h - h'Wv \tag{2}$$

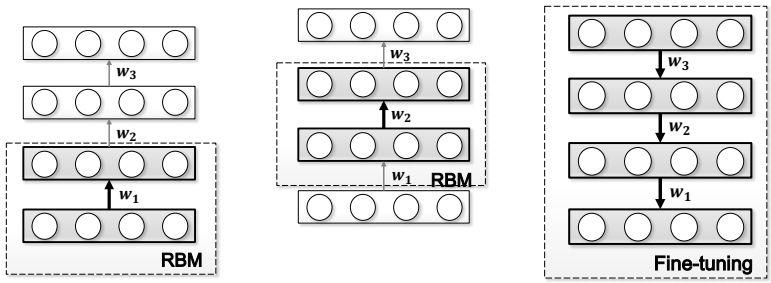
where W represents the weights between hidden and visible units and b, c are offsets of visible and hidden units respectively. Parameters in RBM are obtained by a stochastic gradients of log-likelihood. A log-likelihood gradients are as follows:

$$\frac{logp(v)}{w_{ij}} =< v_i h_j >_{data} - < v_i h_j >_{model} \tag{3}$$

In Eqn. 3, the first term could be calculated directly from given data. But, the second term is computationally impossible to obtain exactly. So, the second term is approximated by the Gibbs sampling method to sample v and h. Using $p(h|v)$ given visible units v, hidden units h are sampled. Then, visible units v could be derived by using $p(v|h)$ given hidden units h. By iterating this procedure as shown in Fig. 1(b), sampled hidden units and visible units are approached to accurate samples of $p(v,h)$. To speed up this iterative sampling process in practice, samples are obtained by only one step of Gibbs sampling [13].

2.2 Deep Belief Network

Deep Belief Network model consists of stacked RBM models as shown in Fig. 2. The first step is to evaluate hidden units in RBM in bottom of DBN. After that, evaluated hidden units are new input data in upper RBM model. By doing



(a) First pre-training. (b) Second pre-training. (c) Fine-tuning.

Fig. 2: A general RBM model and Gibbs sampling process in RBM.

this procedure iteratively, the uppermost hidden units could be evaluated. These units are linked to output units and all weights are updated by back-propagation and this procedure is called fine-tuning.

3 Algorithms

3.1 RANSAC-based plane detection

As a specific task is given to a person, he or she finds some objects that are needed to execute a given task. In an indoor environment, there are many objects on the stable plane such as table or desk. This fact could be applied to robots as well. In images given to a robot, a searching space can be limited to pixels on the plane so that recognition speed could be increased. 3D point cloud is needed to estimate a plane in a image. 3D point cloud is easily derived by the RGB-D camera.

In a 3D point cloud obtained by the RGB-D camera, there are not only pixels that lie in a plane, but also many outliers. These outliers means pixels far from the estimated plane. Sometimes, depth camera might not give exact distance information so that pixels in a plane could be outliers. To release this noisy depth information, the RANSAC algorithm that is robust to outliers is used. After finding parameters for plane equation about inliers, the distance between each point cloud and derived plane is assigned to all pixel in the image. Pixels in inlier set are derived as

$$P_{inlier} = \{p_i \,|\, |D(p_i)| < D_{th}\} \tag{4}$$

where D_i is the distance between p_i and the estimated plane and D_{th} is threshold to discriminate between inliers and outliers. Pixels on the plane are calculated as

$$P_{object} = \{p_i | D(p_i) \geq D_{th}\}. \tag{5}$$

3.2 Learning

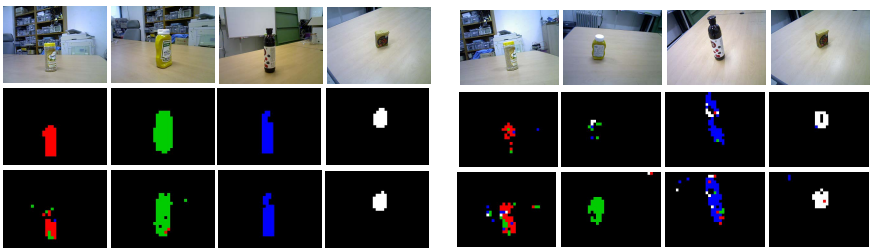
To recognize objects, input images and corresponding ground truth are necessary. A dataset for both object recognition and scene parsing consists of RGB images and ground truth images labeled for all pixels. In this paper, patch-based object recognition is executed; therefore the pixel-based dataset has to be converted to the patch-based dataset. An arbitrary patch's ground truth may have many kinds of labels. In this case, patch level's ground truth is assigned to the most frequent label. After patch-based database is constructed, parameters could be obtained by training. Neural network has two hidden layer of size H and a soft-max output layer of dimension C that is total number of classes. Parameters are pre-trained by RBM and these are trained using a back-propagation and a cross-entropy loss function.

Algorithm 1 Object region estimation using RANSAC

INPUT: 3D point clouds D
OUTPUT: Binary image P
Binary image means whether each pixel belong to object region or not.
Get 3D point clouds D using depth image
for $i = 0 \rightarrow$ max iterations **do**
    $S_i \Longleftarrow$ Random Sample(D) ▷ To estimate a plane, at least 3 points are needed.
    $M_i \Longleftarrow$ Compute Model Parameters(S_i)
    $\{e, D_{inlier}, D_{outlier}\} \Longleftarrow$ VerifyingModelParameter(D, M_i)
    if $e_i < e_{min}$ **then**
        $e_{min} \Longleftarrow e_i$
        $M \Longleftarrow$ ComputeModelParameters(D_{inlier})
        if $e_{min} < \epsilon$ **then**
            Return M_i
        end if
    end if
end for
Parameters are determined and discriminate points on the plane.
$P^* \Longleftarrow$ AbovePlane(D) ▷ Object region is restricted on the plane.

4 Experiment results

Primesense Carmine 1.09 that is a kind of RGB-D camera was used to capture real indoor scene and objects. This camera's depth ranges from 0.35m to 1.4m. Objects in a dataset were four kinds of bottles that contain pepper, mustard, vinegar and sesame. The reason why we chose these objects was that robot may work in a kitchen. There were total 80 labeled pairs of RGB and depth images in the dataset. We used the 40 odd-numbered images as training set, and the remaining 40 even-numbered images as the test set. For learning, two models were used. These models were similar except output units' activation function and error criteria. The first model used a sigmoid function with mean square error. The second model used soft-max function with cross entropy error. The



(a) Recognition result in training dataset. (b) Recognition result in test dataset.

Fig. 3: Object recognition result in training set and test set.

Algorithm 2 Learning Algorithm

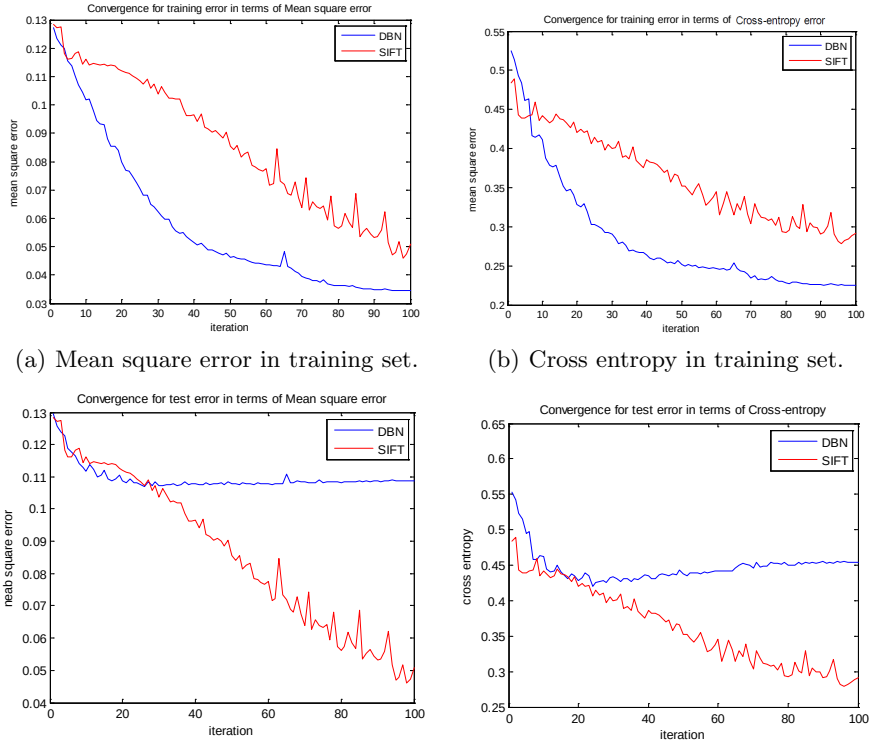
INPUT: Patch-based RGB images X and ground truth Y in a training set
OUTPUT: Parameters W in deep neural network
Initialize all parameters in Deep Network.
Derive hidden units by training RBM.
for $i = 0 \rightarrow$ the number of hidden layer-1 **do**
  for $j = 0 \rightarrow$ max iteration **do**
    $v_i^1, v_i^1 \Longleftarrow \text{GibbSampling}(p(v_i^0, h_i^0))$ ▷ Iterate Gibbs sampling only 1 time.
    $W_{i,i+1}^{new} \Longleftarrow W_{i,i+1}^{old} = W_{i,i+1}^{old} + \mu(< v_i^0 h_i^0 > - < v_i^1 h_i^1 >)$ ▷ Update weights
    $v_{i+1} \Longleftarrow h_i$ ▷ Let hidden units h_i be visible units v_{i+1}
  end for
end for
for $i = 0 \rightarrow$ max iteration **do**
  for $j =$ the number of layer-1 $\rightarrow 1$ **do**
    $W_{j,j+1}^{new} = W_{j,j+1}^{old} + \mu\delta_{j+1} v_i$ ▷ v_i Back-propagate error to lower layer.
    ▷ v_i means current layer's unit values.
    ▷ δ_{i+1} means error back-propagated from upper layer.
  end for
end for

results of object recognition are described in Fig. 3. In the figure, images in the first row are original RGB images. Images in the second row and third row are the the results of DBN and neural network with SIFT descriptor, respectively.

Figs. 4(a) and 4(b) compare training images' convergence in terms of mean square error and cross entropy error, respectively. In these graphs, the model learned by DBN is better than the model that used the SIFT descriptor in terms of convergence error. As shown in Fig. 4(c) and 4(d) that show the results of test images, the model that uses SIFT descriptor is better than the proposed model.

5 Conclusion

In this paper, robust object recognition under occlusion was proposed using RANSAC with the RGB-D camera. Deep Belief Network are used for training and this model was compared with neural network with the SIFT descriptor. Using RANSAC for finding a plane in 3D point cloud captured from depth information, it was possible to recognize objects much faster. In addition, our algorithm could be applied to object recognition under occlusion. Our future work includes the object recognition for the increased number of objects. Although the performance of the proposed method is better than neural network with the SIFT descriptor in training set, the performance of the proposed method in test set is not better. This overfitting problem in DBN has to be solved in the near future.



(a) Mean square error in training set.

(b) Cross entropy in training set.

(c) Mean square error in test set.

(d) Cross entropy in test set.

Fig. 4: Convergence of error in training set and test set.

Acknowledgements

This work was supported by the Technology Innovation Program, 10045252, Development of robot task intelligence technology, funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).
This work was also supported by the Software Computing Technology Development Program, 14-824-09-012, Technology Development of Virtual Creatures with Digital Emotional DNA of Users, funded By the Ministry of Science, ICT and Future Planning.

References

1. Lai, Kevin, et al. "A large-scale hierarchical multi-view rgb-d object dataset." Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011.
2. Huang, Albert S., et al. "Visual odometry and mapping for autonomous flight using an RGB-D camera." International Symposium on Robotics Research (ISRR). 2011.

3. Deok-Hwa Kim, and Jong-Hwan Kim. "Image-Based ICP Algorithm for Visual Odometry Using a RGB-D Sensor in a Dynamic Environment." Robot Intelligence Technology and Applications 2012. Springer Berlin Heidelberg, 2013. 423-430.
4. Henry, Peter, et al. "RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments." In the 12th International Symposium on Experimental Robotics (ISER. 2010.
5. Lenz, Ian, Honglak Lee, and Ashutosh Saxena. "Deep learning for detecting robotic grasps." arXiv preprint arXiv:1301.3592 (2013).
6. Helmer, Scott, and David Lowe. "Using stereo for object recognition." Robotics and Automation (ICRA), 2010 IEEE International Conference on. IEEE, 2010.
7. Bengio, Yoshua. "Learning deep architectures for AI." Foundations and trends in Machine Learning 2.1 (2009): 1-127.
8. Hinton, Geoffrey E. "Deep belief networks." Scholarpedia 4.5 (2009): 5947.
9. Lowe, David G. "Object recognition from local scale-invariant features." Computer vision, 1999. The proceedings of the seventh IEEE international conference on. Vol. 2. Ieee, 1999.
10. Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." Computer VisionECCV 2006. Springer Berlin Heidelberg, 2006. 404-417.
11. Fischler, Martin A., and Robert C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." Communications of the ACM 24.6 (1981): 381-395.
12. Salakhutdinov, Ruslan, and Geoffrey E. Hinton. "Deep boltzmann machines." International Conference on Artificial Intelligence and Statistics. 2009.
13. Bengio, Yoshua, and Olivier Delalleau. "Justifying and generalizing contrastive divergence." Neural Computation 21.6 (2009): 1601-1621.