Friend or Foe: Exploiting Sensor Failures for Transparent Object Localization and Classification

Viktor Seib, Andreas Barthen, Philipp Marohn, Dietrich Paulus
Active Vision Group (AGAS), University of Koblenz and Landau
Universitätsstraße 1, 56070 Koblenz, Germany
{vseib, abarthen, pmarohn, paulus}@uni-koblenz.de
http://agas.uni-koblenz.de

ABSTRACT

In this work we address the problem of detecting and recognizing transparent objects using depth images from an RGB-D camera. Using this type of sensor usually prohibits the localization of transparent objects since the structured light pattern of these cameras is not reflected by transparent surfaces. Instead, transparent surfaces often appear as undefined values in the resulting images. However, these erroneous sensor readings form characteristic patterns that we exploit in the presented approach. The sensor data is fed into a deep convolutional neural network that is trained to classify and localize drinking glasses. We evaluate our approach with four different types of transparent objects. To our best knowledge, no datasets offering depth images of transparent objects exist so far. With this work we aim at closing this gap by providing our data to the public.

Keywords: Transparent Objects, Object Recognition, Depth Images, Deep Neural Network

1. INTRODUCTION

Since their introduction, affordable RGB-D cameras [1] have gained very high popularity as imaging devices. RGB-D cameras have become essential in many applications such as object recognition [2,3] or 3D reconstruction [4,5]. However, as the functionality of these cameras is based on projecting structured light onto the environment to obtain depth information, the recognition and detection of transparent objects seems impossible with such devices. Most previous approaches for transparent object detection have exploited RGB or monochrome camera images. Some approaches need a special or elaborated setup for transparent object detection, like a special light pattern [6] or a complicated lighting setup [7]. The approaches presented in [8] and in [9] learn refraction and distortion of transparent objects and identify such regions in RGB images. So far, only few approaches proposed to use RGB-D cameras for transparent object detection. Wang et al. [10] use boundary detected techniques and a three-dimensional image graph to identify regions of interest. These are then filled with two-dimensional transparent object silhouettes at their respective depth. Lysenkov et al. [11] cover the objects with paint or paper to obtain an opaque surface and use a 3D reconstruction method [4] to obtain an object model.

We propose an approach for detecting and classifying transparent objects with Convolutional Neural Networks using depth images of RGB-D cameras (Fig. 1). The presented approach can be trained on sensor data directly, without any modifications to the objects. While many datasets including RGB-D data exist [12,13], to our best knowledge there are no datasets providing depth images of transparent objects. We aim at closing this gap and provide our dataset for download[1]. Sec. 2 presents our approach. Our contributed dataset is described in Sec. 3. The performed experiments and obtained results are presented in Sec. 4. Finally, the paper is concluded in Sec. 5.

2. TRANSPARENT OBJECT DETECTION AND CLASSIFICATION WITH A CONVOLUTIONAL NEURAL NETWORK

Deep convolutional neural networks are currently the state of the art for classifying images [14,15]. The ability of neural networks to find structures in the input data inspired us to use an unusual type of data, namely measurement errors from RGB-D cameras to localize and classify transparent objects.

Further author information: send correspondence to Viktor Seib, vseib@uni-koblenz.de

[1] Dataset with RGB-D images of transparent objects: http://agas.uni-koblenz.de/data/datasets/transparent_objects/transparent_objects.tar.gz

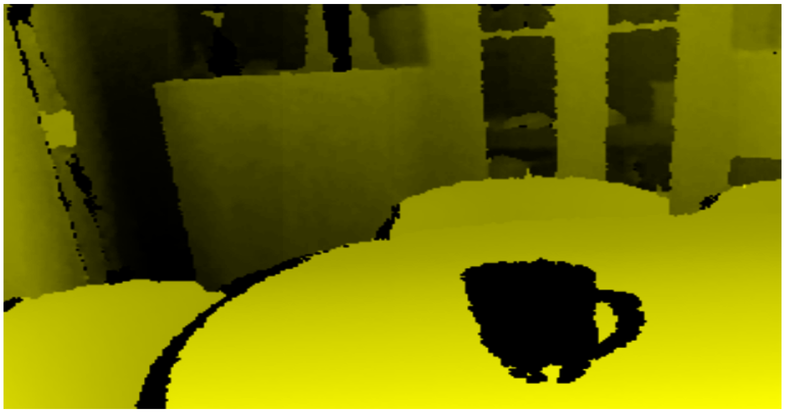

Figure 1. Example depth image of a transparent object

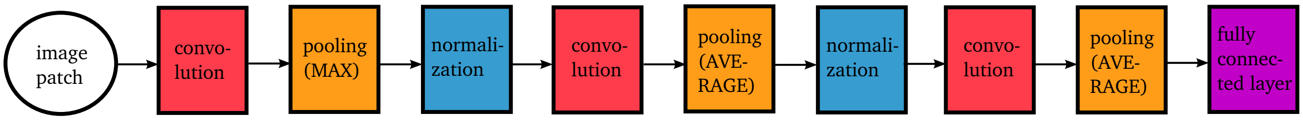

Figure 2. The design of the implemented convolutional neural network. Convolutional layers are depicted in pink, pooling layers in orange and normalization layers in blue. The violet color represents the fully connected layer.

2.1 Network Design

The network Fig. 2 was implemented using Torch [16], a wide-spread framework for deep neural networks. We used the network presented in [15] a basis for our implementation. Our first layer is a convolutional layer with 32 filters and a receptive field of 5×5. We use rectified linear units (ReLU) as activation functions for all layers. After the first convolutional layer there is a max pooling layer of size 3×3 with a step size of 2 in both directions, which is followed by a normalization layer. The second block is the same as the first, only that it uses an average pooling layer instead of a max pooling layer. The third block also uses average pooling, but its convolutional layer has 64 instead of 32 filters. Our last layer is a fully connected layer which maps the output from the previous layer to our classes. Further, we use dropout [17] with a probability of 0.5 for each neuron.

2.2 Training

Because our goal is object classification and localization, our application needs to be able to distinguish an empty image from an image containing an object. This is most easily achieved by learning empty images as negative examples. Because empty images are the least distinguished and the easiest to come by, they make up the largest portion of our dataset (see Sec. 3 for details). The creation of empty images is done by a separate script as part of our dataset preparation. This happens before training in case of both training libraries. We employ data augmentation [15] to allow for better classification of slightly different cutouts of known glasses during localization. From each 35×35 pixels bounding box image, 9 images with the sizes 32×32 and their mirrored versions are generated. This is done for all images, including the empty ones. After data augmentation, we have obtained a data basis that is 18 times larger than the original dataset.

2.3 Classification

Classifications are computed by passing the test image, which needs to be the same size as the network's input layer and thus the same size as the training images, through the trained network. As classification result, the network returns a vector which contains all possible labels with their probabilities for the given image. In our scenario, we only use the top result and return it as our classification label.

2.4 Localization

For localization we cut out image patches from the full sized image and classify them. The iteration starts in the upper left corner of the image and then moves in steps of $\frac{1}{30}th$ of the input image's width across the image

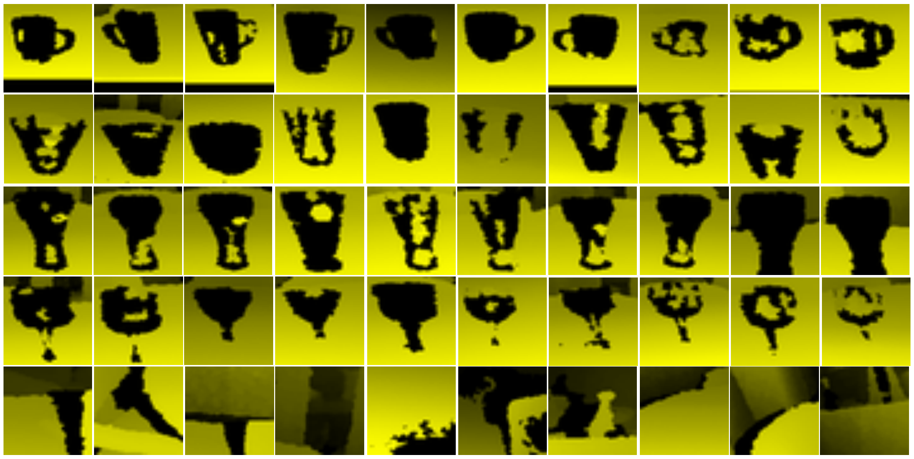

Figure 3. Images from our dataset, each row shows examples for a different class. From top to bottom: beer mugs, water glasses, white beer glasses, wine glasses and empty images.

(sliding window approach). If the confidence of the highest result is below a given threshold (0.5 in our case) or the top result is "empty", this patch is discarded. Otherwise, it is checked for the presence of multiple objects in the neighborhood. When the object found is overlapping with a previous positive match, they are assumed to be of the same object and the corresponding bounding box will be updated by making it as small as possible around the overlapping area. If positive matches do not overlap, they are assumed to be of different objects and will be treated separately. However, all patches that do not overlap with at least a given number of other positive matches are assumed to be false positives and are discarded. The class of the detected object is assumed to be the class label that was assigned most frequently to the overlapping patches. Image patches that were assumed to be false positives do not contribute to this. Our classification can not distinguish between an overlap or a partial image produced by the detection. Also, on a depth image, overlapping images would look like one object, as the black spots do not have depth values. Of course, this makes our algorithm incapable of detecting overlapping objects.

3. DATASET WITH TRANSPARENT OBJECTS

The problem with our task of classifying transparent objects is that none of the existing datasets contains such objects. Thus, we had to create our own dataset containing depth images of transparent objects. The set currently contains 440 images in 4 classes: water glasses, wine glasses, beer mugs and white beer glasses (Fig.3). The set contains full sized side by side images (RGB and depth) taken with an RGB-D camera (ASUS Xtion) from various angles and distances. Each image contains only a single glass on a table, while each class contains different glasses which we considered to be similar. For each side by side image, a bounding box is annotated in a text file. We provide scripts to split the side by side images into RGB and depth components and then obtain an image of the object by using the bounding box information provided. In addition, 8 empty images are created from each full sized image by randomly selecting patches outside the bounding box. These are used as negative examples during training. After executing the script, the full dataset contains 3960 images (3520 empty images and 440 images with transparent objects). We plan to extent this dataset in the future, especially by including images with multiple transparent objects.

4. EXPERIMENTS AND RESULTS

We have performed two types of experiments: classification of images patches (empty and object images) from our dataset and localization of transparent objects on a full-sized images.

4.1 Classification of Image Patches

We use two scenarios for classification. First, we classify unknown images of known glasses, i.e. some of the views of the objects were used only for training, while other views of the same objects were used only for testing.

Table 1. Classification results of image patches.

Test case	Correct glass label	Wrong glass label	Empty image as glass	Glass image as empty
Unknown images	80 (92%)	7 (8%)	1 (0.1%)	0
Unknown glasses	102 (98%)	2 (2%)	0	0

Table 2. Classification and localization results of full-sized images.

Test case	Glass labels assigned	Correct localizations	Correct localizations with correct label
Unknown images	271 (62%)	263 (97%)	208 (77%)
Unknown glasses	205 (47%)	205 (100%)	168 (82%)

We take about 20% of all images with glasses from the dataset to form the test set, while training is performed on the remaining 80%. Thus, the test set is formed by 816 images, 87 of which contain a transparent object. Second, we classify images of completely unknown glasses, i.e. all images of certain glasses were removed from training. We again move about 20% of all images from the dataset to the test set. Thus, the network is never trained on the glasses we use for testing. This test set is formed by 808 images, 104 of which contain glasses.

The results of both classification tasks are presented in Tab. 1. We report the following results in Tab. 1. The columns *correct* respectively *wrong* glass label indicate the number of correctly or wrongly assigned glass labels to an image containing a glass. The column *empty image as glass* shows that only one single image in both tests that did not contain a glass was assigned a glass label. On the other hand there were no images with glasses that were not recognized as such. The results in both test cases are remarkable and show the validity of our approach to make use of erroneous sensor readings for classification of transparent objects. There seems to be a discrepancy of around 6% in the assignment of correct glass labels in the two experiments. We attribute this to the small size of the testing dataset and expect the results to be closer together on a larger set. We thus plan to extend the dataset in the future and perform additional experiments.

4.2 Localization of Transparent Objects

To localize transparent objects in full-sized sensor images we use a sliding window approach. The correct localization thus heavily depends on the classification of image patches. Even though we are using a sliding window approach that produces small image patches for classification, we can not expect that these patches be the same as in the classification test above. This part is evaluated in the same manner as in the previous section: we use the same networks as in the classification test. However, in this case we use 440 full-sized images as input. An example of such an image is presented in Fig. 1. The results of this experiment are shown in Tab. 2.

The leftmost column shows the number of images that were assigned a glass label. All remaining images were marked as “empty” by our approach, i.e. no single patch in the sliding window approach was classified as an object. Both test cases yield very different results (62% vs. 47%). The second test case (unknown glasses) is indeed harder since complete object views of distinct objects were removed from the test set. In contrast, for the first test (unknown images) only distinct views of objects were removed, while the network was still trained on some images of each object. We expect that this difference becomes smaller with a larger dataset. The second column shows the number of correctly localized transparent objects in the images that were assigned a glass label. We observe that almost all cases lead to a correct localization. Finally, the last column in Tab. 2 shows that in 77% respectively 82% percent of all images that were assigned a glass label the glass was localized *and* classified correctly. It is obvious why these values are below the values reported in Tab. 1 for image patch classification. While the sliding window moves along the image different parts of a glass become part of the window and can be easily misclassified. In contrast, the image patch classification was performed on correctly cropped glass images. The class wise analysis shows a significant problem with wine glasses and beer mugs: we obtain wrong class labels in cases where the stem of the wine glass or the handle of the beer mug is outside the image patch. This shows that we need to improve our classification to make it more robust. However, it also shows that even this simple sliding window localization is sufficient, as long as the classification is powerful enough.

5. SUMMARY AND CONCLUSION

In this paper we have presented an approach of transparent object classification and localization that makes use of information that, to our best knowledge, has not been used so far: errors in sensor reading of an RGB-D camera. Our approach enables us to correctly classify and localize 4 different classes of transparent objects. Since no depth image datasets containing transparent objects are publicly available so far, we provide our collected data to the community. Our future work will concentrate on extending our dataset and including images containing multiple transparent objects, as well as improving the classification and localization. Further, we plan to extend our approach to obtain a 3D pose of the localized objects in order to include this algorithm into an existing object manipulation pipeline on our robot.

REFERENCES

[1] Smisek, J., Jancosek, M., and Pajdla, T., "3d with kinect," in [*Consumer Depth Cameras for Computer Vision*], 3–25, Springer (2013).

[2] Lai, K., Bo, L., Ren, X., and Fox, D., "Rgb-d object recognition: Features, algorithms, and a large scale benchmark," in [*Consumer Depth Cameras for Computer Vision*], 167–192, Springer (2013).

[3] Seib, V., Memmesheimer, R., and Paulus, D., "Ensemble classifier for joint object instance and category recognition on rgb-d data," in [*Image Processing (ICIP), 2015 IEEE Int. Conf. on*], 143–147, IEEE (2015).

[4] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., et al., "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," in [*Proc. of the 24th annual ACM symposium on User interface software and technology*], 559–568, ACM (2011).

[5] Henry, P., Krainin, M., Herbst, E., Ren, X., and Fox, D., "Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments," *The International Journal of Robotics Research* **31**(5), 647–663 (2012).

[6] Hata, S., Saitoh, Y., Kumamura, S., and Kaida, K., "Shape extraction of transparent object using genetic algorithm," in [*Pattern Recognition, 1996., Proc. of the 13th Int. Conf. on*], **4**, 684–688, IEEE (1996).

[7] Miyazaki, D., Takashima, N., Yoshida, A., Harashima, E., and Ikeuchi, K., "Polarization-based shape estimation of transparent objects by using raytracing and plzt camera," in [*Optics & Photonics 2005*], 588801–588801, International Society for Optics and Photonics (2005).

[8] Fritz, M., Bradski, G., Karayev, S., Darrell, T., and Black, M. J., "An additive latent feature model for transparent object recognition," in [*Advances in Neural Information Processing Systems*], 558–566 (2009).

[9] McHenry, K. and Ponce, J., "A geodesic active contour framework for finding glass," in [*Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conf. on*], **1**, 1038–1044, IEEE (2006).

[10] Wang, T., He, X., and Barnes, N., "Glass object localization by joint inference of boundary and depth," in [*Pattern Recognition (ICPR), 2012 21st Int. Conf. on*], 3783–3786, IEEE (2012).

[11] Lysenkov, I., Eruhimov, V., and Bradski, G., "Recognition and pose estimation of rigid transparent objects with a kinect sensor," *Robotics* , 273 (2013).

[12] Lai, K., Bo, L., Ren, X., and Fox, D., "A large-scale hierarchical multi-view rgb-d object dataset," in [*Robotics and Automation (ICRA), 2011 IEEE Int. Conf. on*], 1817–1824, IEEE (2011).

[13] Silberman, N., Hoiem, D., Kohli, P., and Fergus, R., "Indoor segmentation and support inference from rgbd images," in [*Computer Vision–ECCV 2012*], 746–760, Springer (2012).

[14] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., "Going deeper with convolutions," in [*Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*], 1–9 (2015).

[15] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," in [*Advances in neural information processing systems*], 1097–1105, MIT Press, Cambridge, MA (2012).

[16] Collobert, R., Farabet, C., Kavukcuoglu, K., and S., C., "Torch." `http://torch.ch/`. Accessed: 2016-02-01.

[17] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research* **15**(1), 1929–1958 (2014).