Detection of Package Edges in Distance Maps

Elena Vasileva
Faculty of Electrical Engineering and Information Technologies
Ss. Cyril and Methodius University
Skopje, North Macedonia
el.vasileva.7@gmail.com

Nenad Avramovski
Faculty of Electrical Engineering and Information Technologies
Ss. Cyril and Methodius University
Skopje, North Macedonia
nenadavrm@gmail.com

Zoran Ivanovski
Faculty of Electrical Engineering and Information Technologies
Ss. Cyril and Methodius University
Skopje, North Macedonia
zoran.ivanovski@feit.ukim.edu.mk

***Abstract*— This paper presents a CNN-based algorithm for detecting package edges in a scene represented with a distance map (range image), trained on a custom dataset of packaging scenarios. The proposed algorithm represents the basis for package recognition for automatic trailer loading/unloading. The main focus of this paper is designing a semantic segmentation CNN model capable of detecting different types of package edges in a distance map containing distance errors characteristic of Time-of-Flight (ToF) scanning, and differentiating box edges from edges belonging to other types of packaging objects (bags, irregular objects, etc.). The proposed CNN is optimized for training with a limited number of samples containing heavily imbalanced classes. Generating a binary mask of edges with 1-pixel thickness from the probability maps outputted from the CNN is achieved through a custom non-maximum suppression-based edge thinning algorithm. The proposed algorithm shows promising results in detecting box edges.**

Keywords—edge detection, semantic segmentation, depth maps, CNN, package recognition, automatic unloading

I. Introduction

Automating the process of loading and unloading of transported goods will bring a significant reduction in financial expenses the transport industry is facing due to damaged goods. Successful partial or complete process automation will lead to improvements in the speed and accuracy of loading and unloading of transported goods. Automating package recognition is the most significant initial step towards process automation, as paper boxes of various sizes and materials are the most common types of packaging. The main steps in package recognition are precise detection and localization of packages, and successfully differentiating packages from other types of packaging, such as bags and irregular objects (cylindrical packaging etc.).

All packages share the same representation in distance maps or depth images – a package is represented as a box consisting of adjacent perpendicular planar sides marked with a gradual change in distance/depth represented with gray levels. ToF scanning is susceptible to distance measurement errors such as irregularly erroneous distance measurements of highly reflective surfaces, and different measurement values for adjacent surfaces with a sharp difference in color. Furthermore, package sides may contain irregularities due to physical damage during transport.

Edges mark the borders of two surfaces at different depth, or two surfaces with a different direction or rate of change in gray levels. Therefore, edge detection would be the most reliable way of precise localization of boxes. Edge detection as a means of package recognition should enable detection of edges that belong to packages while disregarding edges belonging to different objects and artificial edges within package sides resulting from depth measurement errors, or physical deformation of package shape. Package edges represent long, continuous straight lines separating two areas with different direction and rate of change of surface depth; as opposed to edges of other types of packaging items that are short, broken and less emphasized. Objects with irregular uneven surfaces (e.g. bags) contain a large number of edges within the object surface. Conventional DSP (Digital Signal Processing) methods for edge detection (e.g. Canny [1]) are unable to make a distinction between different types of edges. Furthermore, they are dependent on user-defined thresholds that cannot be generalized for different datasets and conditions. This creates the need for learning-based methods for reliable edge detection in the context of box recognition in distance maps.

Learning-based edge [2, 3, 4] and contour [5, 6, 7, 8, 9, 10, 11] detection algorithms designed for and tested on color photographs, which contain more edges within objects resulting from object texture, are not suitable for application on distance maps and require a large training dataset due to the model complexity. Ref. [12] presents a CNN classifying pixels of a photograph into edge and non-edge based on the local pixel environment. However, global image context is not available for the classification of a single pixel, and using a large pixel environment requires a lot of redundancy (the larger the pixel environment, the more times pixels are forwarded through the network) and thereby an excessive amount of computational time and resources, which is not allowed for an application targeting real-time performance.

The CNNs with an encoder-decoder structure are an efficient and straightforward solution for edge detection in a single feed-forward step without additional pre-processing. U-Net [13] provides state-of-the-art results in object segmentation in photographs or medical images. However, due to the large number of parameters, it requires large datasets to reach top performance. An encoder-decoder CNN with a limited number of parameters, as used mainly for medical image segmentation, would be simple enough to be trained with a limited number of samples, and simultaneously provide enough capacity to retain the crucial dataset features for correct segmentation of box edges.

The proposed algorithm presents an end-to-end trainable CNN structure for semantic segmentation of box edges from distance images with heavily imbalanced classes, followed by a custom edge thinning algorithm to generate binary edge masks with single-pixel-width edges. By precise detection of box edges and thereby localizing boxes, this algorithm represents a crucial initial step for package recognition in distance maps for automatic trailer loading/unloading.

This work is performed within the scope of the project 'Object recognition in 3D scenes based on depth and image data', sponsored by Fast Global Solutions, Inc. Information and techniques disclosed in this paper are owned and in patent-pending status by Fast Global Solutions, Inc.

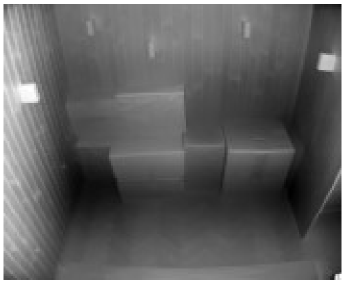

Fig. 1 Distance map with stretched value range representing a packaging scene. Lighter pixel values represent points closer to the scanner.

Fig. 2 Color photograph of the scene represented with a distance map in Fig. 1.

The rest of this paper is arranged as follows. Section II provides the description of a custom dataset of packaging scenarios. Section III provides a detailed description of the proposed algorithm. Section IV covers the post-processing methods. Section V presents the results and experiments. Finally, conclusions are presented in Section VI.

II. Description of Dataset and Annotations

A. Dataset description

A custom dataset consisting of 272 distance maps of 144x176 pixels is used for training and performance evaluation of the proposed box edge detection algorithm. An example of a distance map is shown in Fig. 1. A color photograph as a clearer visual representation of the scene in Fig. 1 is given in Fig. 2. The distance maps are obtained by averaging 20 consecutive measurements provided by a SICK Visionary-T DT infrared surface depth scanner based on ToF technology. According to the manufacturer, the distance maps obtained at a rate of 50 frames per second may contain distance errors of ±3cm for measuring distances less than 3m. During the creation of the custom dataset in a simulated trailer space, we have discovered that the distance errors are much larger than what is described in the documentation.

The dataset contains different scenes with three types of stacked packaging items: boxes, shipping bags and irregular objects (cylindrical packaging, unpackaged carpet rolls, etc.). The packaging items are arranged in one of two configurations: package walls that represent the most common scenario of carefully ordered packages, and arbitrary order that represents cases of tumbled packaging items that may occur during transportation or unloading errors. Two types of background are represented: planar floor and walls, and non-planar background where the trailer walls are fully occluded by objects of arbitrary shape. The planar background simulates most types of trailer interiors. The non-planar background materials introduce variety in the background data to reduce overfitting of machine learning algorithms on the background data, thereby causing a significant drop in model accuracy on scenes with types of trailer interiors not represented in the training dataset.

B. Distance measurement errors and effect

Several types of distance errors are observed in the dataset: rounding of inner edges, displacement of outer edges, displacement of whole box sides along the depth axis, and irregularly erroneous measurements on reflective surfaces (tape, labels). Rounding of inner edges - a result of multipath effects - has been addressed in several works [14, 15]. The closer a point is to an inner edge - the more this effect is emphasized; which causes a smooth rounding of the scanned surface. Rounding of inner edges poses a challenge in edge detection since the change in distance levels is unnoticeable in a small local environment. Proposed correction algorithms have resulted in limited success. Points on outer edges of boxes appear closer to the scanner than neighboring pixels belonging to the two box sides forming the edge. The edge displacement ranges from 20-100mm. The level of distortion (exact value of displacement) depends only on the orientation of the box - edges facing the scanner are most affected. The large outer edge displacement manifests as sharp lines with depth values significantly different from the neighboring pixels, resulting in easier detection. Irregular distance errors that are a result of reflective surfaces create sharp changes in distance levels mimicking short, jagged edges, and correct segmentation depends on the global context of the environment. Tight enclosed spaces such as trailers filled with objects contain multiple light reflection points. This results in emphasizing the distortion effects, thereby posing an additional challenge for precise object detection inside transport trailers.

C. Generating ground truth data

The ground truth data are binary masks marking edges of clearly visible and partially occluded boxes of varying sizes and orientations. The ground truth data for each distance map consists of two binary masks, one marking the inner, and one marking the outer box edges. Fig. 3 presents the ground truth data for the distance map in Fig. 1, with the inner box edges marked in green and the outer marked in yellow. Ground truth data are provided for 225 scans. The edge classes (pixels belonging to an inner box edge and pixels belonging to an outer box edge) are each represented with 2.2% of the total number of pixels in the dataset.

III. Detection of Box Edges

A. Data preparation

The input distance maps contain barrel distortion noticeable only on large planar surfaces (trailer floor, walls); therefore the box edges are represented by straight lines and applying distortion correction has little effect on the results of edge detection. However, both inner and outer edges are heavily affected by ToF distance measurement errors. Previous work has proved the ability of CNNs to learn complex perspective transformations [16], therefore we hypothesize the CNN can adapt to the heavy depth distortion of edges without the need for preprocessing. Further motivated by minimizing processing time, the raw distance maps that are normalized in the range 0 to 1 are used as CNN input without other preprocessing. Stretching the contrast emphasizes the edges (difference in pixel intensity), which proves beneficial in edge detection.

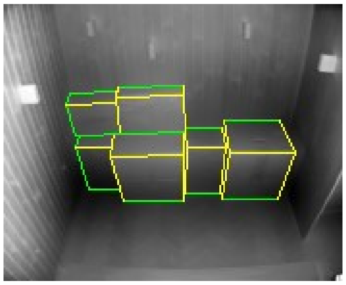

Fig. 3 Binary mask representing the ground truth data for the distance map in Fig. 1. Outer box edges are marked in yellow, and inner box edges in green.

125 randomly chosen scans comprise the training set, 20 scans are chosen for validating, and 80 for testing the CNN. Due to the small training set, a different random combination of data augmentation strategies (zoom, rotation, shifting, shear, and horizontal flip) is applied to each batch of training samples in every epoch to maximize the number of different data transformations. This augmentation strategy helps to simulate a 5 times larger dataset in the worst-case scenario.

B. *Proposed CNN structure*

As discussed in detail in Section II.B, the inner and outer edges are represented by largely different characteristics in the distance maps - inner edges are rounded, and outer edges are significantly emphasized due to distance measurement errors. The large difference in the representation of the two types of edges motivates training the model to generate two separate probability maps for the two types of edges.

The diagram in Fig. 4 presents the structure of the proposed CNN model. The encoder (contraction path) consists of 2 contraction blocks marked in yellow. The first contraction block contains a downsampling layer which reduces the feature map dimensions by 2 using a non-overlapping window. The decoder (expansion path) consists of 2 expansion blocks marked in green. The second expansion block contains an upsampling layer which doubles the feature map dimensions to generate class labels for all pixels of the input image (producing output with same dimensions as the input). All hidden layers are followed by ReLU activations and operate on zero-padded input feature maps with 3x3 filter size. The output convolution layer contains 3 filters to provide output probability maps for outer edges, inner edges, and non-edge pixels. The output convolution layer is followed by sigmoid activation. The number of pooling layers is limited to one, since reducing the feature maps further leads to discarding of valuable data containing crucial features for detecting edges. Each edge class is represented with only 2.5% of the total number of pixels in the dataset; therefore discarding a large amount of information significantly reduces model accuracy. Fig. 5 displays the probability map of outer edges, and Fig. 6 the probability map of inner edges in the distance map in Fig. 1.

C. *Optimization details*

The ADAM (ADAptive Moment estimation) optimization algorithm [17] with a learning rate of 10^{-4} and the binary cross-entropy cost function are used in the training process. The model is trained for 300 epochs with a mini-batch size of 20 samples and fine-tuned for another 50 epochs with a mini-batch size of 2 samples. The small batch size enables retaining important features present in a small number of samples, which are lost using a larger batch size (retaining general dataset features).

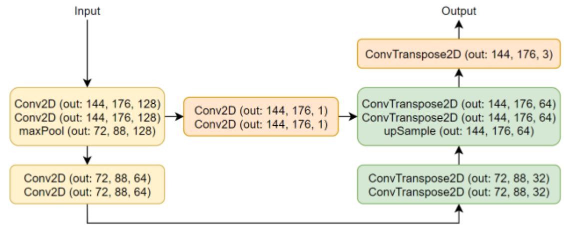

Fig. 4 Proposed CNN structure for semantic segmentation of inner and outer box edges. The first two numbers from the output dimensions represent the height and width of feature maps generated by the layer, and the last number represents the number of convolution filters in the layer.

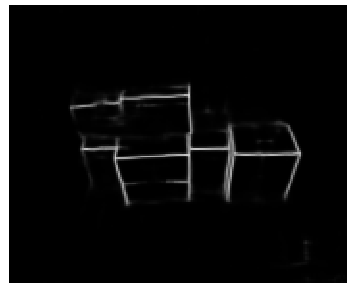

Fig. 5 Probability map of outer box edges in the distance map in Fig. 1, generated by the proposed CNN (Fig. 4).

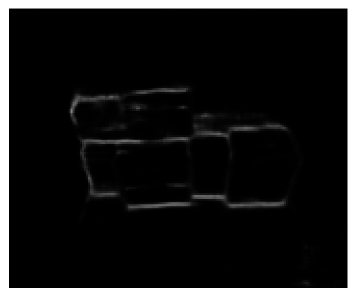

Fig. 6 Probability map of inner box edges in the distance map in Fig. 1, generated by the proposed CNN (Fig. 4).

IV. POST-PROCESSING

A. *Trailer floor and walls suppression*

Trailer walls with non-planar structure (e.g. ribbed walls, as shown in Fig. 1) and packaging types other than boxes (bags, irregular objects) contain significantly more edges than boxes. Furthermore, outer edges are emphasized by the depth scanning errors. This may result in detecting edges that do not belong to boxes as separate edges or additional edge segments connected to box edges. The position of the trailer walls and floor can be obtained with system calibration, and they can subsequently be easily removed. Other packaging types (bags, irregular objects) can be detected with different algorithms and subsequently removed. For this work, trailer floor walls and other packaging types are manually removed.

B. *Edge Thinning*

The desired output of the edge detection algorithm is continuous lines with 1px thickness for each inner and outer box edge. The edge detection CNN generates continuous probability maps for inner and outer edges for every input distance map. As observed in Fig. 7, the detection probability of the edges varies by a large margin. Inner edges or distorted outer edges may form lines as thick as 5 pixels in the probability maps. Therefore an edge thinning algorithm based on non-maximum suppression is designed to obtain a binary mask of edges with 1-pixel thickness from the output probability maps.

First, probability values lower than 0.1 are discarded from the probability map. Two binary masks are generated for each probability map, each marking the positions of local maxima along each row and column of the probability map, respectively. A binary edge mask is formed by combining the two masks of local maxima with a logical OR operation. Disconnected edge components consisting of fewer than 2 pixels are removed. The final edge mask is obtained by merging the thinned masks of the inner and outer edge maps with a logical OR operation. Edge components containing less than 5 pixels are removed. The morphological post-processing steps do not ensure edge continuity. This must be provided by the CNN output (Fig. 7).

V. Results and Experiments Analysis

The high performance of the proposed model on the validation and testing sets containing samples with large variety prove the model generalizes successfully over the different packaging types and configurations. As observed in Fig. 5 and Fig. 6, the emphasized outer edges are detected with a much larger probability in comparison to the inner edges. Cases of outer edges with a gradual change in depth due to distance measurement errors are also detected with a lower probability. Therefore, the binarization threshold is fixed to a relatively low value of 0.1. All crucial box edges are detected with probability above the threshold, and the CNN strongly rejects the background pixels and the other types of packaging (bags, irregular objects). Therefore a custom edge thinning algorithm is able to successfully produce an edge mask with edge thickness of 1px encompassing all types of box edges. Several false negative cases of inner edges as a result of too large distortion due to distance measurement errors are observed in the test data. An example of this can be seen in Fig. 7 (parts of edges touching the left wall are not detected).

Fig. 8 shows the results of edge segmentation for the scene represented with a color photograph in Fig. 9. The proposed CNN described in Section III.B is compared to

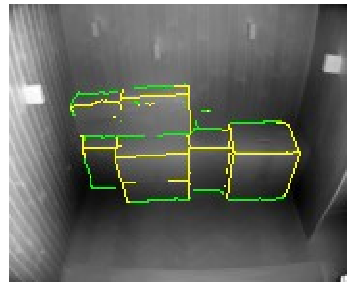

Fig. 7 Binary edge mask for the distance map in Fig. 1 obtained by merging the results of applying edge thinning to the probability maps of outer and inner edges (Figs. 5, 6). Detected outer box edges are marked in yellow, and inner box edges in green.

three algorithms described in previous works: Canny edge detector, original U-Net architecture performing binary edge classification, and a reduced U-Net architecture featuring only two consecutive convolution blocks in the contraction and expansion path with filter numbers 128 – 64 – 64 – 128 and a single downsampling layer in the first convolution block, also performing binary classification.

The algorithm performance is calculated on a test set of 80 distance maps containing packages, bags and several types of irregular objects. The performance metrics in Table I take into account false positive detection errors on bags and irregular objects. Some types of errors are more significant for the particular application of package loading/unloading. Packages at the top front of the scene are the first to be removed; therefore detecting them correctly is crucial. The removal of the front and top boxes decreases the occlusion of the back boxes and allows errors on the back packages to be corrected. Furthermore, bags and irregular objects can be detected with other detection algorithms, thereby eliminating any false positives originating from them. Visual inspection of the results confirms the front packages not affected by heavy distortion are detected correctly. Several cases of parts of reflective surfaces with erroneous distance measurements ape detected as edges mark false positive classifications by the proposed CNN (small edge components within box sides in Fig. 8d). However, such false positive segments cannot lead to box surface oversegmentation since there are no cases of false positive edges completely splitting any surface into multiple parts.

Table I compares the proposed method and experiments to previous edge detection algorithms. Accuracy, precision and recall are calculated on the binary edge masks with 1px

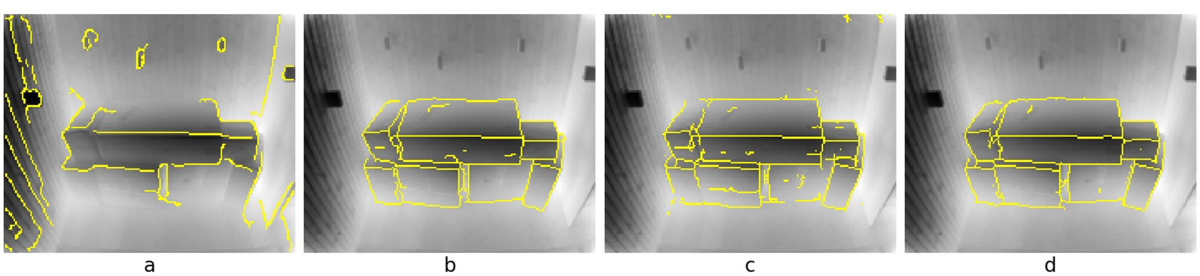

Fig. 8 Results of edge segmentation with different algorithms: a) Canny, b) U-Net, c) Reduced U-Net, d) Proposed. The resulting thinned edge masks are marked in yellow. The proposed CNN (d) produces smoothest edge lines and fewest false positive pixels within the box surfaces. All box edges are detected. Canny (a) performs the worst, producing a lot of wavy lines on the box sides and missing a large portion of the edges due to the distortions in the distance map. The edges produced by U-Net (b) contain more jagged lines and false positive pixels within the box sides, even in places with small distance measurement errors. Reduced U-Net (c) fails to detect parts of the inner edges with a gradual change in gray levels (parts of box edges touching the back wall) and shows greater sensitivity to distance measurement errors and depth discontinuities compared to (d).

Fig. 9 Color photograph of the scene represented with a distance map in Fig. 8.

TABLE I. RESULTS AND PERFORMANCE COMPARISON OF PROPOSED ALGORITHM AND EXPERIMENTS

CNN	Acc (%)	Precision	Recall	AP
Canny	93.57	0.1207	0.2381	-
U-Net	96.80	0.3738	0.3584	0.3486
Reduced U-Net	96.92	0.3976	0.3804	0.3845
Proposed multi-class	**97.13**	**0.4487**	0.5091	**0.4928**
Proposed 2 poolings	96.32	0.3520	**0.5109**	0.4394
Proposed class weight	96.63	0.3821	0.4986	0.4272

thick edges obtained with the algorithm in Section IV.B, and AP (Average Precision) is calculated on probability maps generated by the CNNs. The Canny edge detector does not provide a probability map; therefore no value is supplied for AP. Accuracy is not a very reliable metric for a dataset with heavy class imbalance such as the one in this work, therefore the superiority of the proposed algorithm is most noticeable in the AP metric results. The AP values are significantly lower compared to the accuracy values because the probability assigned to the ground truth data is 1.0, and the CNNs produce lower probability values for the edges. According to Table I, confirmed by the visual representation of the results in Fig. 8, the proposed multi-class CNN provides the best overall results in box edge detection. U-Net is too complex to be successfully trained with a limited number of samples. The poor performance of the binary classification models (U-Net and Reduced U-Net) shows that separating the learning objective into two (inner and outer edges) facilitates defining the edges. Adding pooling layers to the proposed multi-class configuration significantly reduces the CNN precision due to loss of details, and provides much worse overall accuracy despite the high recall value. Class frequency balancing during training cost calculation according to the percentage of pixels each class is represented with does not improve model performance.

VI. CONCLUSION

This paper presents an end-to-end trainable CNN structure for semantic segmentation of package edges in a distance map optimized for training with a limited number of samples and heavily imbalanced classes. As shown by the visual results and calculated metrics, the proposed CNN performs correct segmentation of box edges represented by a sufficient number of distance points, regardless of the position and orientation of the boxes and depth measurement errors characteristic of ToF depth scanning. Therefore, the CNN is able to successfully retain crucial features from the small, highly variable dataset. Separating the target objective into two classes represented by highly different features (inner and outer edges) significantly improves the segmentation performance. A simple custom edge thinning algorithm successfully creates one pixel thick box edges. Following from the presented results and analysis, the proposed algorithm represents a successful initial step for package detection for automatic trailer loading/unloading.

REFERENCES

[1] J. Canny, “A computational approach to edge detection,” In IEEE Transactions on Pattern Analysis and Machine Intelligence, v.8 n.6, p.679-698, June 1986.

[2] S. Xie, and Z. Tu, “Holistically-nested edge detection,” In ICCV, 2015.

[3] Z. Yu, C. Feng, M. Liu, and S. Ramalingam, “CASENet: Deep category-aware semantic edge detection,” In CVPR, 2017.

[4] B. Ma, C. Liu, X. Wei, M. Gao, X. Ban, H. Wang, H. Huang, W. Xue, “WPU-Net:Boundary learning by using weighted propagation in convolution network,” Preprint at https://arxiv.org/pdf/1905.09226.pdf.

[5] J.J. Lim, C.L. Zitnick, and P. Doll´ar, “Sketch tokens: a learned mid-level representation for contour and object detection,” In 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3158–3165, 2013.

[6] G. Bertasius, J. Shi, and L. Torresani, “Deepedge: A multiscale bifurcated deep network for top-down contour detection,” In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Boston, USA, 2015.

[7] J.-J. Hwang, and T.-L. Liu, “Pixel-wise deep learning for contour detection,” In ICLR, 2015.

[8] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang, “Deepcontour: A deep convolutional feature learned by positivesharing loss for contour detection,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3982-3991, 2015.

[9] M. Li, Z. Lin, R. Mech, E. Yumer, and D. Ramanan, “Photo-sketching: Inferring contour drawings from images,” In 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), 2019.

[10] K.-K. Maninis, J. Pont-Tuset, P. Arbelaez, and L. Van Gool, “Convolutional oriented boundaries: From image segmentation to high-level tasks,” In IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.

[11] L. Han, X. Li, and Y. Dong, “Convolutional Edge Constraint-Based U-Net for Salient Object Detection,” In IEEE Access, 2019.

[12] R. Wang, “Edge detection using convolutional neural network,” In International Symposium on Neural Networks, pp. 12-20, 2016.

[13] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” In Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015, pp. 234–241, 2015.

[14] Y. He, B. Liang, Y. Zou, J. He, and J. Yang, “Depth errors analysis and correction for time-of-flight (ToF) cameras,” In Sensors, vol. 17, no. 1, pp. 92, 2017.

[15] Y. He, and S. Chen, “Recent Advances in 3D data acquisition and processing by Time-of-Flight camera,” In IEEE Access, pp. 1-1, January 2019.

[16] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee, “Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision,” In Advances in Neural Information Processing Systems, pp. 1696-1704, 2016.

[17] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” In Proceedings of the 3rd International Conference on Learning Representations, 2015.