

# Fast 3D Edge Detection by Using Decision Tree from Depth Image

Masaya Kaneko, Takahiro Hasegawa, Yuji Yamauchi, Takayoshi Yamashita,

Hironobu Fujiyoshi, and Hiroshi Murase

**Abstract**— T3D edge detection from a depth image is an important technique of 3D object recognition in preprocessing. There are three types of 3D edges in a depth image called jump, convex roof, and concave roof edges. Conventional 3D edge detection based on ring operators has been proposed. The conventional ring operator can detect three types of 3D edges by classifying the response of Fourier transforms. Since the conventional method needs to apply Fourier transforms to all pixels of a depth image, real-time processing cannot be done due to high computational cost. Therefore, this paper presents a fast and reliable method of detecting three types of 3D edges by using a decision tree. The decision tree is trained under supervised learning from numerous synthesized depth images and labels by capturing depth relations between candidate pixels and pixels on a ring operator to classify 3D edges. The experimental results revealed that the proposed method has 25 times faster than the conventional method. This paper also presents some examples of 3D line and 3D convex corner detection based on results obtained with the proposed method.

## I. INTRODUCTION

The detection and localization of edges and corners are important tasks for robot vision in preprocessing. Corner and edge detection are used as the first step in simultaneous localization and mapping (SLAM) [1] [2] [3] and 3D object detection [4]. Depth information obtained from new ranging sensors such as time-of-flight TOF cameras [5] is widely used in computer vision and they have enabled breakthroughs in several tasks [6]. Our research is concerned with detecting 3D edges in a depth image. 3D edges are also helpful for robot navigation. There are three types of 3D edges called jump, convex roof, and concave roof edges. Fig. 1 outlines three types of 3D edges of a cube in a depth image. A jump edge can be observed in Fig. 1 (b) on the boundary between the object and background. A convex roof edge and a concave roof edge can be observed on the boundary between two surfaces. Fig. 1 (c) shows edges detected by a Laplacian edge detector, which is widely used for gray-scale images. As can be seen from Fig. 1 (c), it is hard to detect both convex and concave roof edges because the gradient of neighboring pixels in the depth image is small. A ring operator that detects three types of 3D edges in a depth image has been proposed by Inokuchi *et al.* [7] to overcome this problem. First, the ring operator, which is a circle of 32 pixels around the edge candidate, is applied by using spectral analysis of waveforms such as fast Fourier

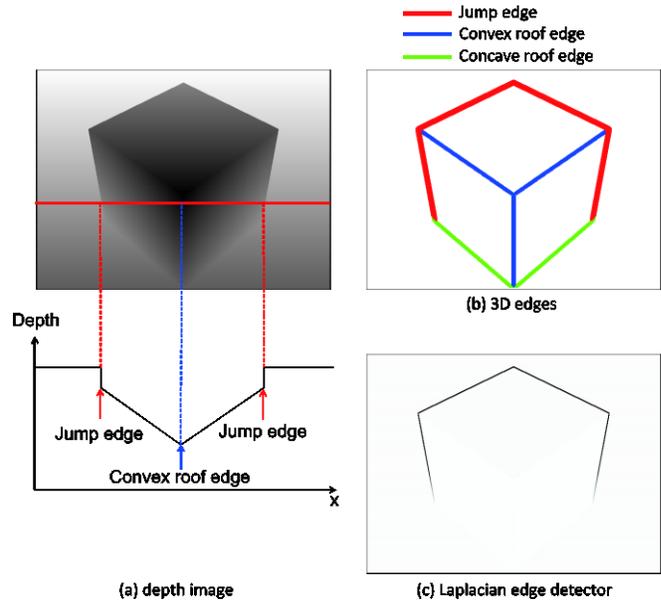


Fig. 1. 3D edges in a depth image.

transform (FFT). Then, the criterion operates that results in spectral analysis to classify waveforms into three types of 3D edges or planes. Since the ring operator needs to apply FFT to all pixels in a depth image by raster-scanning, real-time processing cannot be done due to the high computational cost of FFT. Real-time processing is required since 3D edge detection in a depth image is used in preprocessing. Features from the Accelerated Segment Test (FAST) have been used for high-speed corner detection to detect corners in gray-scale images [8], [9]. A machine learning approach has been introduced in FAST to identify corners.

We applied a machine learning approach to the 3D edge detector in a depth image because we were inspired by the work done with FAST. We propose a fast and reliable method of detecting three types of 3D edges by using a decision tree in this paper. The decision tree is trained under supervised learning using many synthesized depth images and edge labels by capturing the depth relation between an edge candidate and pixels on the ring operator in order to classify the 3D edges. The remainder of the paper is organized as follows. Section II discusses related work. Section III introduces the proposed method. Section IV presents the experimental results. Section V presents some applications based on the proposed method, and the paper is concluded in Section VI.

## II. RELATED WORK

This section briefly describes the definition of 3D edges in a depth image and related work.

Masaya Kaneko, Takahiro Hasegawa, Yuji Yamauchi, Takayoshi Yamashita, and Hironobu Fujiyoshi are with the Chubu University, Kasugai, Aichi Japan. {msy, tkhr, yuu}@vision.cs.chubu.ac.jp  
{yamashita, hf}@cs.chubu.ac.jp  
Hiroshi Murase is with the Nagoya University, Nagoya Aichi, Japan. murase@nagoya-u.jp

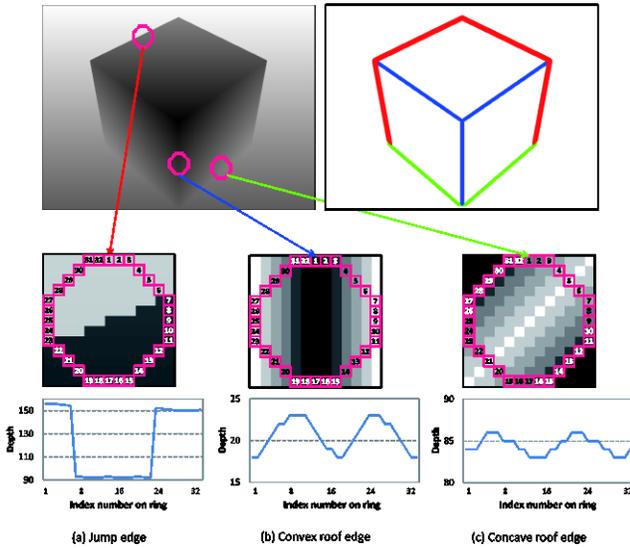


Fig. 2. Ring operator and properties of 3D edges.

### A. Definition of 3D edges in depth image

A depth image contains information relating to the distance of surfaces of objects in a scene from the viewpoint of the camera. Since each pixel in a depth image has a distance similar to the intensity of a gray-scale image, it is easy to apply conventional image processing techniques such as edge and corner detection. However, a depth image has another aspect of properties related to 3D structures compared to gray-scale images. Edges in a gray-scale image are observed at pixels in which the gradient of neighboring pixels is high. There are three types of edges in a depth image.

- **Jump edges** Jump edges are observed on the boundary between an object (surface) and the background (non-surface). The value of the gradient on the jump edge is high, as shown in Fig. 2 (a), which represents depth discontinuities.
- **Convex Roof edges** Roof edges are observed on the boundary between two surfaces. Convex roof edges have convex shapes on the cube shown in Fig. 2 (b).
- **Concave Roof edges** Concave roof edges have concave shapes in which the boundary is usually observed between the surface of the floor and the cube shown in Fig. 2 (c).

It is useful to detect these three types of edges separately to identify the 3D structure of a scene. However, conventional edge detector based on Laplacian filter can not classify these three types of edges.

### B. Edge detection methods based on surface normal

There are already methods of edge detection based on surface normal for depth images [10], [11], [12], [13]. Puili *et al.* [10] proposed surface normal-based edge detector from a depth image. Jiang *et al.* [11] split each line of the image into a set of quadratic polynomials. Approaches using mathematical morphology operators have also been developed [12]. Ye *et al.* proposed a robust edge detector with a Singular Value Decomposition (SVD) filter to smooth object surfaces [13].

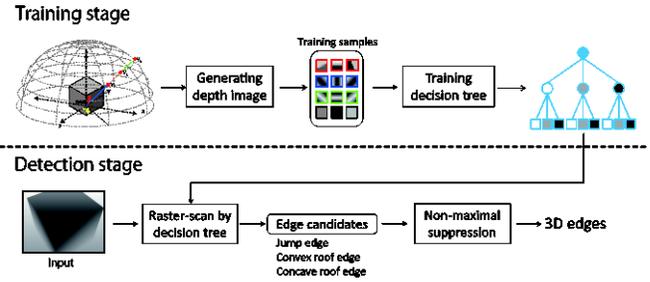


Fig. 3. Overview of proposed method.

These surface normal-based edge detectors are used for object detection via 3D geometry [14] and image segmentation [15].

### C. Ring operator

Inokuchi *et al.* proposed a method of classifying 3D edges by spectral analysis along a ring operator [7], as seen in Fig. 2. Fig. 2 shows 3D edges have different properties in the form of waveform signals converted from a circle of 32 pixels. Inokuchi *et al.* applied spectrum analysis (FFT) to waveforms by taking advantage of these different properties. They could classify them into jump edges, convex roof edges, concave roof edges, and planes by using three thresholds and the first and second components of the Fourier spectrum. Fig. 2 (b) shows an example of a 3D edge detected by Inokuchi *et al.* [7]. Real-time processing cannot be done since the ring operator needs to apply FFT to all pixels of a depth image by raster-scanning. It takes the ring operator approximately 0.7 sec to detect a 3D edge in a depth image with a VGA size of 640 by 480 pixels. The size of the ring operator needs to be set to two exponents for processing with FFT.

## III. PROPOSED METHOD

We applied a machine learning approach to a 3D edge detector for a depth image. This section describes a fast and reliable method of detecting three types of 3D edges by using a binary decision tree and a ternary decision tree. Fig. 3 overviews the proposed method.

### A. Training sample

The proposed method can deal with changes in viewpoints. As training samples that represented many different viewpoints were generated, we placed a virtual cube at the center to generate a depth image with computer graphics (CG) rendering from a viewpoint whose Euler angle rotation parameters,  $\varphi$  and  $\theta$ , were set at equal intervals, as shown in Fig. 4. The rotation ranges for the parameters were  $\varphi \in [0, 45]$ , and  $\theta \in [0, 90]$  in this work, and the interval for  $\varphi$  and  $\theta$  was 11.25. The distance from the viewpoint to the center position of the cube was set from 1.5 m to 3.0 m. We added Gaussian noise to the generated depth image of a training sample to simulate a real depth image obtained with a TOF camera. We also created edge labels for all pixels in the generated depth image, in which the labels belonged to one of four classes of jump edges, convex roof edges, concave roof edges, or planes. These four classes were determined by applying a ring operator [7]. We correctly manually labeled pixels with the ring operator. We generated 180 depth images by doing so and used them to train the decision tree.



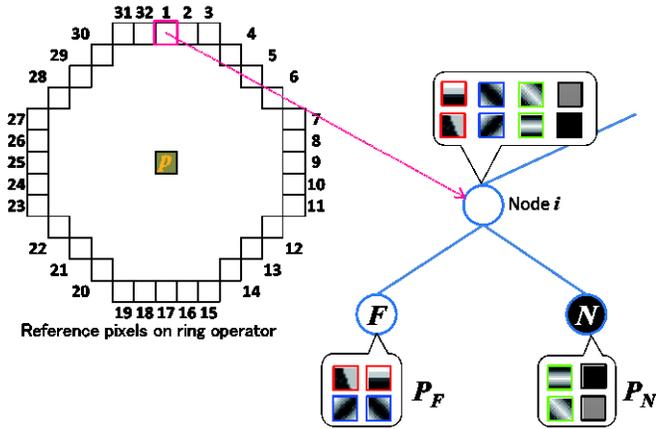


Fig. 6. Binary decision at node  $i$ .

Flexible thresholds can represent various changes in 3D edges by changing the value of the threshold. However, the computation time for the traversal decision tree can be more than that of a fixed threshold.

### C. Training of ternary decision tree

The depth of the binary decision tree described above will be large due to binary decision. It is important to create a decision tree with fewer depths to enable faster 3D edge detection. We introduced a ternary decision tree with three states to overcome this problem. Fig. 5 (b) outlines the structure of a ternary decision tree. Pixels positioned relative to  $p$ , denoted by  $p \rightarrow x$ , can have one of three states of Far, Equal, and Near for each location on the ring,  $x \in \{1, \dots, 32\}$ :

$$S_{p \rightarrow x} = \begin{cases} F & D_p + t_1 \leq D_{p \rightarrow x} & (\text{Far}) \\ E & D_p + t_2 < D_{p \rightarrow x} < D_p + t_1 & (\text{Equal}) \\ N & D_{p \rightarrow x} \leq D_p + t_2 & (\text{Near}). \end{cases} \quad (4)$$

$F$  means that pixel  $x$  is far from pixel  $p$ .  $N$  means that pixel  $x$  is near it.  $E$  means that the distance of pixel  $x$  is equal to that of pixel  $p$ .  $t_1$  and  $t_2$  are thresholds for ternary decision. We search by selecting  $x$  that yields the most information about whether the candidate pixels are edges at each node while training, which is measured by information gain:

$$G(P) = H(P) - H(P_F) - H(P_E) - H(P_N), \quad (5)$$

where  $P_F$  is the number of training samples classified as ‘‘Far’’,  $P_N$  is the number of training samples classified as ‘‘Near’’, and  $P_E$  is the number of training samples classified as ‘‘Equal’’.  $H$  is the entropy computed with (3). We train the ternary decision tree until the information gain derived by (3) becomes zero. Leaf nodes store edge labels in which the number of samples at the nodes is maximum.

### D. Edge detection by decision tree and non-maximal suppression

1) *3D edge detection by decision tree*: Raster scanning of the depth image is employed at the detection stage. The previous decision tree that was trained distinguishes pixels into one of the four classes of jump edges, convex roof edges,

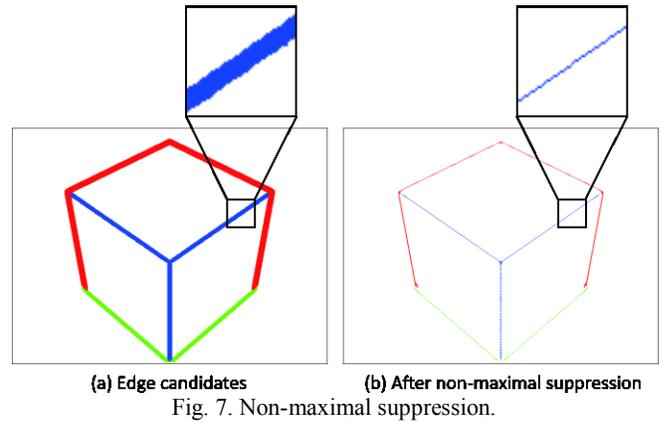


Fig. 7. Non-maximal suppression.

concave roof edges, or planes by doing tree traversal at each pixel. Fig. (7) (a) shows an example of 3D edge detection with the proposed method.

1) *Non-maximal suppression*: We can see from Fig. (7) (a) that pixels around the original edge have also been detected. We introduced an approach of non-maximal suppression to solve this problem. Since the decision tree does not directly compute edge responses, we define a pseudo-edge response by

$$response(p) = \max_{x \in \{1, \dots, 32\}} |D_p - D_{p \rightarrow x}|, \quad (6)$$

where  $p$  is the location of a detected 3D edge with the proposed method, and  $x$  is its location on the ring. We then suppress miss-detected pixels in two steps.

- 1) We compute the pseudo-edge responses with (6) for all candidate pixels detected by the decision tree.
- 2) The pseudo-edge responses in neighboring pixels (3 by 3 pixels) in which an edge label of  $x$  is the same as  $p$  are ranked at candidate pixel  $p$  according to the values of the pseudo-edge responses. If the value of a response at pixel  $p$  is within the 3rd rank, pixel  $p$  is finally stored as a 3D edge. If not, pixel  $p$  is rejected as not being a 3D edge. Please note that we do not only use the 1st rank because strong responses are observed at neighboring pixels, which are lying on the edge.
- 3) Repeat 2) for all candidate pixels detected by the decision tree.

Fig. (7) (b) outlines an example of our non-maximal suppression for 3D edges.

## IV. EXPERIMENTAL RESULTS

We experimentally compared the proposed method with conventional methods to find how effective it was.

### A. Experimental overview

We compared the ring operator [7], surface normal [10], and the proposed method (binary decision tree and ternary decision tree) regarding the  $F$ -measure obtained by the following equations:

$$F - measure = \frac{2 \times precision \times recall}{precision + recall}, \quad (7)$$

TABLE I  
F-MEASURE

	Ring operator[7]	Surface normal[10]	Binary decision tree		Ternary decision tree	
			Fixed th.	Flexible th.	Fixed th.	Flexible th.
Jump edge	0.898	-	0.768	0.884	0.780	0.911
Convex roof edge	0.759	-	0.726	0.817	0.769	0.831
Concave roof edge	0.781	-	0.689	0.764	0.717	0.797
Plane	0.999	-	0.736	0.991	0.765	0.997
Average	0.859	0.706	0.729	0.862	0.757	0.884
Processing time [ms]	683.2	61.5	47.2	35.0	22.7	26.5
Depth of decision tree	-	-	15.4	12.9	8.7	10.1

$$recall = \frac{TP}{TP + FN}, \quad (8)$$

$$precision = \frac{TP}{TP + FP}. \quad (9)$$

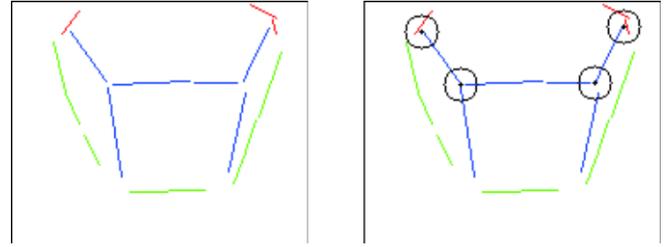
The decision trees were trained with 180 depth images. We used another 90 depth images with different viewpoints in the evaluation test, which were generated by CG and had added noise. We have also presented some examples of 3D edge detection using a TOF camera below.

### B. Comparison with conventional methods

The results for the accuracy of detection are presented in Table I, where we can see that the highest accuracy was achieved by the ternary decision tree, followed by the binary decision tree, surface normal, and finally the ring operator. Fig. 8 provides some examples obtained with the proposed method and the ring operator. The reason for this improvement is that the proposed method is more robust to noise by training the decision tree with noise-added images. However, the ring operator is affected by noise due to spectral analysis. The processing times for a depth image with a VGA size of 640 by 480 pixels are summarized in Table I. The hardware used for these experiments was a personal computer equipped with a 2.67 GHz Intel X7452 CPU. The ternary decision tree had the fastest processing time, followed by the binary decision tree, surface normal, and the ring operator. Since the proposed method based on the decision tree could decrease the number of pixels accessed on the ring, we detected 3D edges 25 times faster than the ring operator. We can also see that the processing time by the ternary decision tree is faster than that by the binary decision tree. This is because the depth of the decision tree is shallower due to the introduction of ternary decision.

### C. 3D edge detection from depth image obtained by TOF camera

Fig. 8 shows some examples of 3D edges detected with the three methods from a real depth image taken with a TOF camera (mesa SR-4000). We can see that decision trees with flexible thresholds have better 3D edge detection with fewer



(a) Line segment detector (b) Convex corner detector  
Fig. 9. Application of LSD and 3D convex corner detection

false positives.

## V. APPLICATIONS

We confirmed from our experimental evaluation that the proposed method provided fast and reliable 3D edge detection. This section presents two examples of applications based on results obtained with the proposed method. Fig. (9) (a) has some examples of line segment detection (LSD) that was applied [17] to jump edges, convex roof edges, and concave roof edges detected with the proposed method. These line segments with attributes of 3D edges should be very helpful for reconstructing 3D scenes and recognizing 3D objects. Fig. 9 (b) shows some examples of 3D convex corner detection. We trained a decision tree using the results from the proposed method to classify whether edges were 3D convex corners or not. Since 3D convex corners consist of three lines with convex roof edges, it is easy to detect them without surface normal computation.

## VI. CONCLUSION

We presented a fast and reliable method of 3D edge detection based on a decision tree. We achieved a 25 times faster than that with the conventional ring operator by using machine learning. We will attempt to identify 3D objects in a scene in future work by using the outputs of the proposed method as representations of internal features.

## REFERENCES

- [1] M. Montemerlo, S. Thrun, D. Loller, B. Wegbreit, *et al.*, "Fastslam: A factored solution to the simultaneous localization and mapping problem," in *AAAI National Conference on Artificial Intelligence*, 2002, pp. 593-598.
- [2] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99-110, 2006.

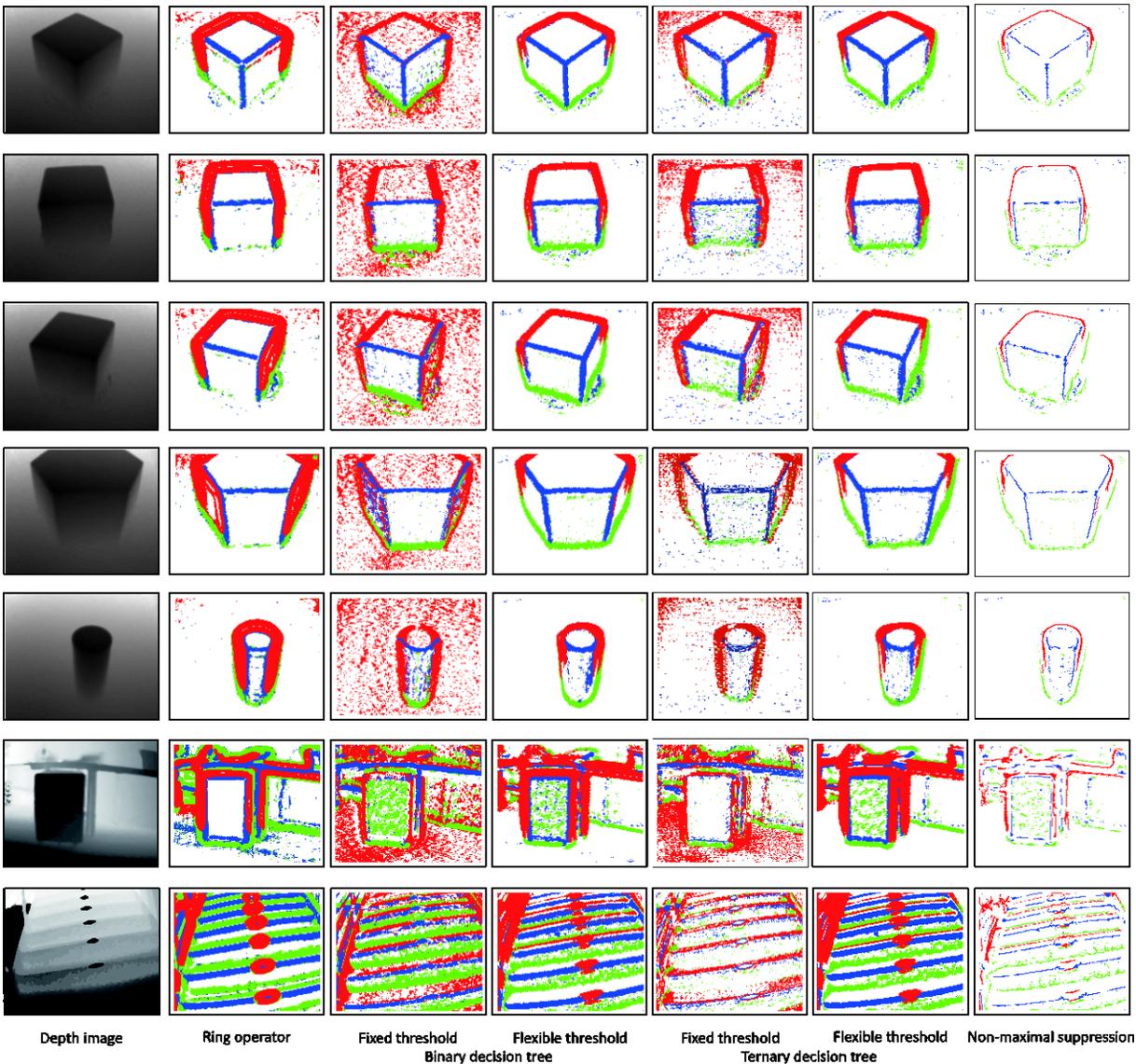


Fig. 8. Edge detection examples using depth image obtained by a TOF camera(SR-4000). Red color means jump edges, Blue means convex roof edge, and green means concave roof edge.

- [3] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (slam): Part ii," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108-117, 2006.
- [4] R. Rios-Cabrera and T. Tuytelaars, "Discriminatively trained templates for 3d object detection: A real time scalable approach," in *IEEE International Conference on Computer Vision*, 2013, pp. 2048-2055.
- [5] R. Lange and P. Seitz, "Solid-state time-of-flight range camera," *IEEE Journal of Quantum Electronics*, vol. 37, no. 3, pp. 390-397, 2001.
- [6] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE MultiMedia*, vol. 19, no. 2, pp. 4-10, 2012.
- [7] S. Inokuchi, T. Nita, F. Matsuda, and Y. Sakurai, "A three dimensional edge-region operator for range pictures," in *International Conference on Pattern Recognition*, 1982, pp. 918-920.
- [8] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on Computer Vision*. Springer, 2006, pp. 430-443.
- [9] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105-119, 2010.
- [10] P. Kari and P. Matti, "Range image segmentation based on decomposition of surface normals," in *Scandinavian Conference on Image Analysis*, IEEE, 1993, pp. 893-899.
- [11] X. Jiang and H. Bunke, "Edge detection in range images based on scan line approximation," *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 183-199, 1999.
- [12] R. Krishnapuram and S. Gupta, "Morphological methods for detection and classification of edges in range images," *Journal of Mathematical Imaging and Vision*, vol. 2, no. 4, pp. 351-375, 1992.
- [13] C. Ye and G. M. Hegde, "Robust edge extraction for swissranger sr-3000 range images," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 2437-2442.
- [14] A. Shrivastava and A. Gupta, "Building part-based object detectors via 3d geometry," in *IEEE International Conference on Computer Vision*, 2013, pp. 1745-1752.
- [15] Z. Jia, A. Gallagher, A. Saxena, and T. Chen, "3d-based reasoning with blocks, support, and stability," in *International Conference on Pattern Recognition*, 2013, pp. 1-8.
- [16] J. R. Quinlan, "Induction of decision trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 1, pp. 81-106, 1986.
- [17] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722-732, 2010.