



Depth edge detection by image-based smoothing and morphological operations

S.M. Abid Hasan, Kwanghee Ko*

School of Mechatronics, Gwangju Institute of Science and Technology, Gwangju, South Korea

Received 3 December 2015; received in revised form 5 February 2016; accepted 10 February 2016

Abstract

Since 3D measurement technologies have been widely used in manufacturing industries edge detection in a depth image plays an important role in computer vision applications. In this paper, we have proposed an edge detection process in a depth image based on the image based smoothing and morphological operations. In this method we have used the principle of Median filtering, which has a renowned feature for edge preservation properties. The edge detection was done based on Canny Edge detection principle and was improvised with morphological operations, which are represented as combinations of erosion and dilation. Later, we compared our results with some existing methods and exhibited that this method produced better results. However, this method works in multiframe applications with effective framerates. Thus this technique will aid to detect edges robustly from depth images and contribute to promote applications in depth images such as object detection, object segmentation, etc.

© 2016 Society of CAD/CAM Engineers. Production and hosting by Elsevier. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Keywords: Edge; Depth Image; Smoothing; Morphology

1. Introduction

As depth edges represent object contours [1,2], proper edge detection offers a significant role in various computer vision problems. Edge maps contain part of geometric information of natural scenes; especially in the case of depth images, depth discontinuities separate foreground objects from the background and can be used for various image processing tasks such as segmentation or denoising [3]. Even though there have been some rapid progresses in the field of image patch classification [4–6] object detection is still an active research topic.

The 3D measurement technology is generally based on calculating depth information of objects in a scene. There has been some work in 3D measurement technologies, where researchers tried to acquire the depth information of objects in a scene by using stereo cameras [7–9]. However, this approach is limited because stereo cameras are able to work in the scenes containing plentiful textures. Some people used three dimensional laser range finders [10,11]. They were successful to produce

accurate depth data. But they cannot use this device to real-time applications due to their expensive apparatus. Even some pattern-based methods [12–15] were used to produce a depth map but those methods also have some limitations with respect to the cameras and object positions. Since the rapid growth of the availability of inexpensive RGB-D sensors such as Apple Prime Sense, Microsoft Kinect, Intel Real Sense, etc., a lot of breakthroughs have been achieved for several tasks such as 3D modeling [16], segmentation [17] and body pose estimation [18,19]. Though these sensors helped us to achieve depth maps but very few methods have been applied to analyze such maps in terms of edge detection.

Accurate edge detection from a depth image is essential for some object detection processes [20], which are dependent on a model of particular shape. A proper edge detection process can be used for various Human action analysis [21] problems in a real environment such as walking, spotting and sitting. Existing edge detection processes in depth images, however, cannot be applied in these types of situations due to some limitations. Some methods of edge detection in depth images [22] failed to deliver noise-free depth images; thus proper edge detection cannot be achieved.

*Corresponding author.

Some other methods [22,23] work in edge detection in depth images. However, they only work for a single frame. When they are applied to problems requiring processing multi-frames, these methods may fail as they cannot deal with newly generated noises in each frame, such as flickering problems.

In this paper, we proposed a method that can detect edges from depth images more profoundly. This method can detect continuous edges, which are very important when we try to detect a large object from an image such as a human body. To detect continuous edges, we modified the Canny edge detection method by incorporating morphological operation. The morphological operation generally deals with shape of features in an image. It has been used to remove imperfections from various types of shape such as object boundaries, skeletons, etc. This operation generally consists of two operators; erosion and dilation. The first operation, denoted as opening, smooths the contour object, breaks narrow strips and eliminates thin protrusions. The second operation, called closing, also smooths contours but in contrast with opening; it fuses thin discontinuities, eradicate trivial holes and fills gaps in the contour. This method also works in multi-frames, whereas the previous methods detect edges in a single frame.

This paper is structured as follows: Section 2 explains the overall framework of the proposed system. It also presents the way by which a depth image is acquired. In Section 3, we present a process for smoothing the depth image. Section 4 represents an edge detection process in the depth image. In Section 5 we show our experimental results for different scenes and compare our results with an existing method. Finally Section 6 concludes this paper along with some directions for possible future work.

2. Overall framework

In Fig. 1 we have illustrated the overall framework of the proposed system. Here, a depth image has been acquired by using Microsoft Kinect. The data captured from Kinect has to be processed in a certain way. After proper processing of the Kinect data we smooth the depth image for reducing the level of noises using the principle of Median filtering. Later, we move on to the edge detection process with the smoothed

depth image. Here, we use the principle of Canny Edge Detection and modified it with Morphological operations for gaining better performance. The morphological operations are opening and closing operations. In this section we discuss the technical approaches for gaining the depth image from Kinect and process it for additional applications. Microsoft provides some built-in libraries which have been used in this work for acquiring the depth image. However, there are some preprocessing task for using Microsoft Kinect. **NuiGetSensorCount** function has been used to make sure the numbers of sensors ready for use. **NuiImageResolutionToSize** function has been used to make sure the numbers of sensors ready for use. **NuiImageResolutionToSize** function has been used to acquire the width and height of the depth frame and **INuiSensor::NuiImageStreamOpen** function has used to initialize the sensor to stream out the depth data. Once we have started to get the depth stream we had started to capture the data from the next frames by using **INuiSensor:NuiImageStreamGetNextFrame** and by using **INuiSensor:NuiImageStreamReleaseFrame** each frame has been released after saving it.

Once we capture the depth data frame by frame it is essential to converse with Kinect to get the depth data, particularly one frame at a time. In this situation, we have used an abstract base class called KinectHelper which holds the functions for this type of communication. Here, we have used KinectHelper: UpdateDepthFrame method to update the frame of depth data and later by using KinectHelper: GetDepthImageAsArgb we acquired the Kinect data formatted as ARGB. Once we found this data we would be able to progress our later algorithms as per our later obligations.

3. Depth image smoothing algorithm

In this section we will explain the principle that we have used to reduce the level of noises in our acquired depth image. Here we have used the Median filtering principle [24].

There are several filters for removing noises from images. However, the median filter is different from most of the existing ones. The most important feature of Median filter is that it preserves edges while removing noises. Basically, Median filter analyzes through every image pixel and replaces

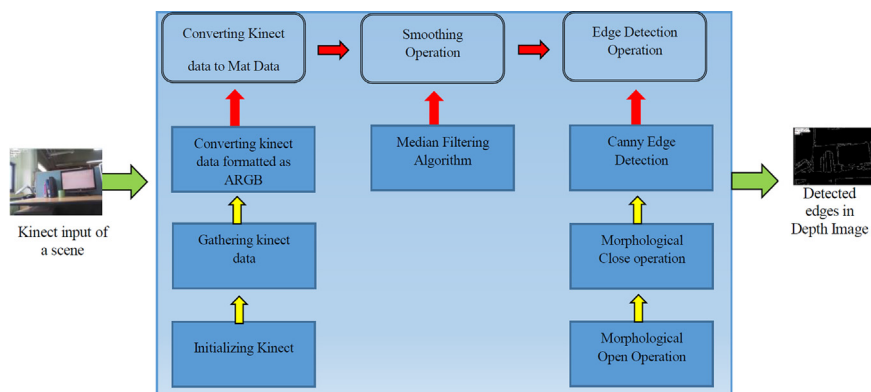


Fig. 1. Overall framework.

every image pixel by pixel by the median of the pixels in the corresponding filter region R . This process can be expressed according to the following equation:

$$I'(u, v) \rightarrow \text{median}\{I(u+i, v+j) | (i, j) \in R\} \quad (1)$$

Here, (u, v) is the position of the image pixel and (i, j) is the neighborhood size of the image region and these are specified as a two element numeric vector of positive integers. By using median filtering, each output pixel contains the median value in the i -by- j neighborhood around the corresponding pixel in the input image.

Now, as this filter replaces the pixel values with median values the median of $2K+1$ pixel values p_i can be defined as

$$\text{Median}(p_0, p_1, p_2, \dots, p_k, \dots, p_{2k}) \stackrel{\Delta}{=} p_k \quad (2)$$

In Eq. (2) the center value will be p_k if the sequence (p_0, \dots, p_{2k}) is sorted by $(p_i.C.p_i+1)$; sorted according to their numerical values. As the median filter does not generate any new valued pixels that are not in the image before Eq. (2) defines the median of an odd sized set of values. In the situation where the side length of the rectangular filters are odd, then the number of elements in the filter region becomes odd as well. For the other case where the number of elements is even ($2K$ for some $K > 0$), then the median of the sorted sequence (p_0, \dots, p_{k-1}) will be defined as the following equation:

$$\text{Median}(p_0, \dots, p_{k-1}, p_k, \dots, p_{2k-1}) = (p_{k-1} + p_k) / 2 \quad (3)$$

In this equation, the arithmetic mean of the two middle values, p_{k-1} and p_k has been considered as the median of the sequence. As this interpolation creates a new pixel value the region for the median filter is considered to be of even size.

In Fig. 2 we have shown the effects of the 3×3 pixel median filter region in a certain two dimensional image structure. Here, all the nine values extracted from this image region are sorted as a vector. The center value in the vector is considered as the median and eventually replaces the center value of the prescribed pixel region.

In our algorithm, however, we have used the same principle except that we have used a 5×5 pixel median filter in our depth image structure. In the example section we have illustrated the effect of this median filtering technique.

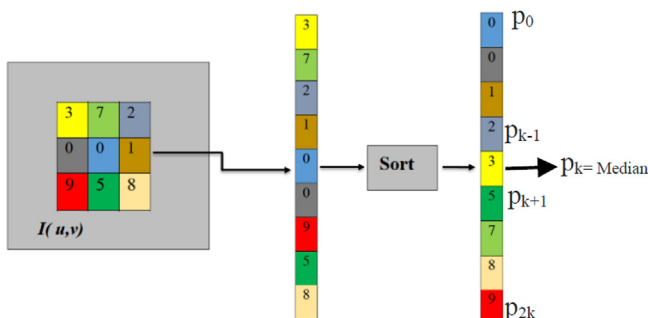


Fig. 2. Computation of a 3×3 pixel median filter.

4. Edge detection algorithm in depth image

This section illustrates the edge detection process of our system. Here, we have used the principle of Canny edge detection process. For achieving better performance we have modified it with some Morphological operations [27].

4.1. Canny Edge detection algorithm

There are several edge detection operators in the image processing field and Canny edge detector [25] is one of the most effective detectors. It is able to perceive an extensive range of edges in an image. As it meets accurately the general criteria's for edge detection and the implementation process is quite simple it has achieved well-earned recognition for edge detection. In the following sections we explain the process of Canny Edge detection algorithm [26]:

4.1.1. Process of Canny Edge detection algorithm

The process of Canny Edge detection algorithm can be categorized into five different steps:

1. *Smoothing*: To remove noise from the image, smoothing or blurring operation are performed.
2. *Finding gradients*: After having the gradients of the image, edges should be marked only in those areas where large magnitudes are obtained.
3. *Non-maximum suppression*: Only local maxima should be considered as edges.
4. *Double thresholding*: Prospective edges are determined by double thresholding.
5. *Edge tracking by hysteresis*: After suppressing all the edges that are not connected to a very certain or strong edges, those will be considered as the final edges.

In our work, we do not use all of these steps as we have already applied the smoothing operation by using Median filtering. So we skipped the first step and the other steps had been done for our method. Later on, by introducing Morphological operation we have modified this edge detection process and in terms of depth image this process is completely new.

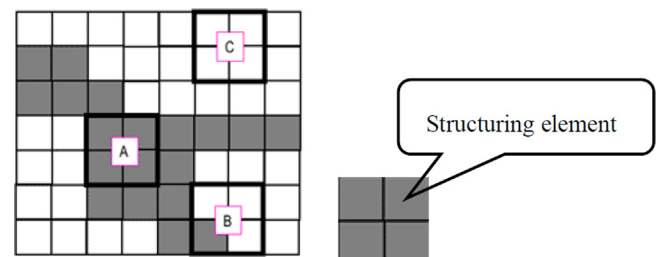


Fig. 3. Probing an image with a structuring element.

4.2. Modification of Canny Edge detection algorithm

Though Canny Edge detector has been able to detect the edges in a certain manner, to achieve more accurate edge detection from a depth image we have modified this edge detection process using morphological operations. These operations are generally a collection of nonlinear operations carried out comparatively on the ordering of pixels without affecting their numerical values. The key operators for morphological operations are erosion and dilation. In the following we explain the morphological operations [27] used in our method. Based on the shape or morphology of features in an image morphological image processing continues to do some collection of nonlinear operations. These operations depend on the relative ordering of pixel values rather than the numerical values and consequently are especially appropriate for processing binary images. However, morphological operations can be used in other types of images as well, such as grey-scale images. In grey scale images the light transfer functions are unknown and thus their absolute pixel values are of no or minor concern. Usually, Morphological techniques investigate an image by using a small shape or template, which is called the *structuring element*. This structuring element is usually placed at all probable locations in the image and then compared with the corresponding neighborhood pixels. During this operation, this technique tests whether the element “fits” within the neighborhood or whether the element “hits” or intersects the element. Fig. 3 shows how these operations are carried out over image pixels. In case of binary images, if the

test is successful at a particular location then this operation creates a new binary image in which the pixel contains a non-zero value at that particular location of the input image.

The *structuring element* used in this morphological operation is a small binary image, a small matrix of pixels where each of the pixels contains with the value of zero or one. The key explanation can be written as follows:

- The matrix dimensions specify the *size* of the structuring element.
- The pattern of ones and zeros specifies the *shape* of the structuring element.
- An *origin* of the structuring element is usually one of its pixels, although generally the origin can be outside the structuring element.

In our research, we have used a morphological operation that is represented as a combination of erosion and dilation. The first operation is called *opening* and the other is called *closing*. Opening is a morphological filter where erosion is followed by dilation. And, closing is a morphological filter where dilation is followed by erosion.

These morphological operations can be explained in terms of set-theoretic operations such as the complement of a binary image:

$$f^c(a, b) = 1 \text{ if } f(a, b) = 0, \text{ and } f^c(a, b) = 0 \text{ if } f(a, b) = 1$$

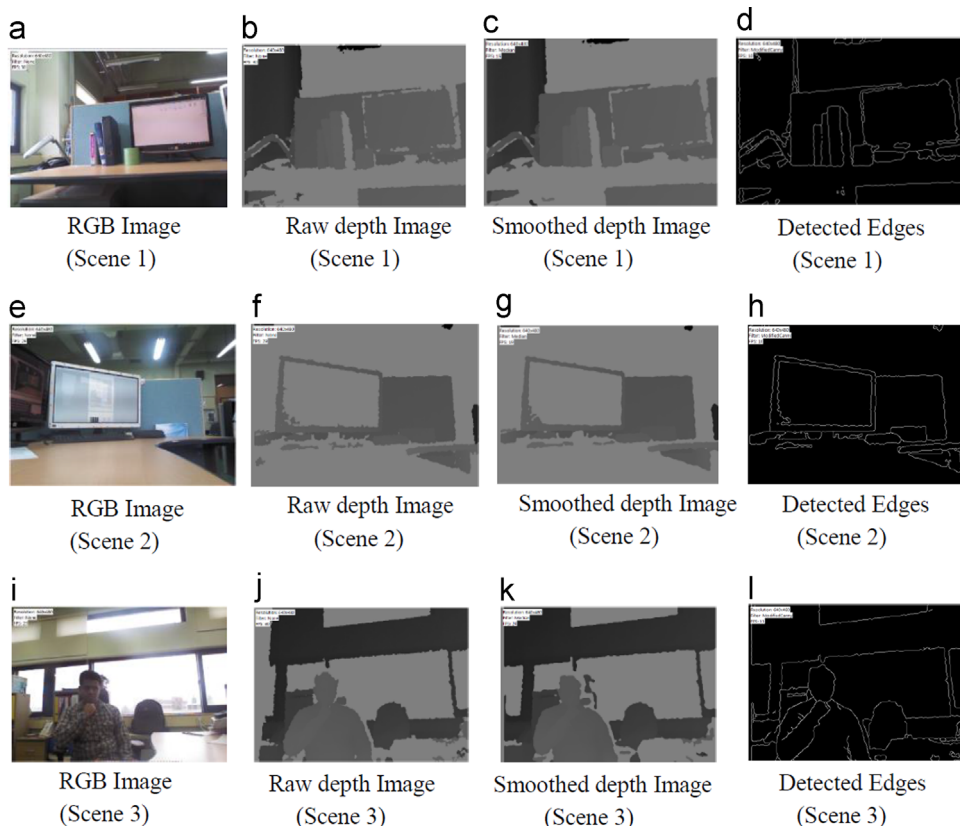


Fig. 4. Edge detection result in depth image for various scenes.

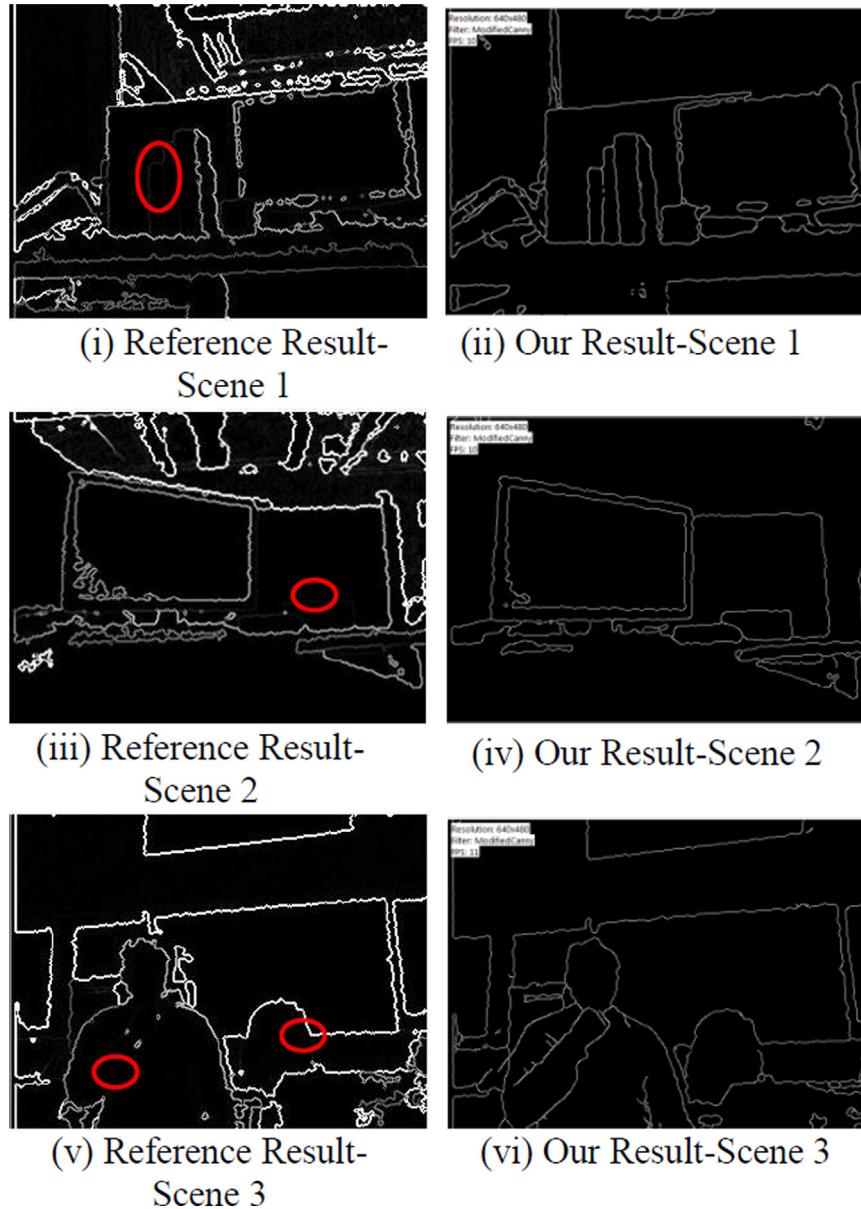


Fig. 5. Comparison of test result of various scenes.

Here, the complement, f^c is the set of elements that are not contained in image f .

Now, the *intersection* $I=f \cap g$ of two binary images f and g can be explained as the following equations:

$I(a,b) = 1$ if $f(a,b) = 1$ and $g(a,b) = 1$, and $I(a,b) = 0$ otherwise

The *union* $U=f \cup g$ of two binary images f and g can be explained as the following equations:

$U(a,b) = 1$ if $f(a,b) = 1$ or $g(a,b) = 1$, and $U(a,b) = 0$ otherwise

In the following we explain the opening and closing operations [27], respectively.

4.2.1. Opening

The opening of an image f by a structuring element s (denoted by $f \circ s$) is an erosion (denoted by \ominus) followed by a dilation

Table 1

Test scene No.	Computation time for Reference method (s) [22]	Computation time for Proposed Method (s)
01	0.725	0.1
02	0.686	0.1
03	0.687	0.1
04	0.656	0.09

(denoted by \oplus):

$$f \circ s = (f \ominus s) \oplus s$$

The reason for calling this operation opening is because it can open up a gap between objects connected by a thin bridge of pixels. As part of this operation, the regions which may have survived the erosion operation will be restored to their original size by dilation. The benefit of opening with a disc

structuring element is that it smoothens corners from the inside. And, by doing the entire operation it removes further noise, round corners from inside and abridges the image.

4.2.2. Closing

The closing of an image f by a structuring element s (denoted by $f \bullet s$) is a dilation (denoted by \oplus) followed by erosion (denoted by \ominus):

$$f \bullet s = (f \oplus s) \ominus s$$

In morphological operations, closing is originated because it is able to fill holes in the regions while keeping the initial region sizes unchanged.

This closing operation with a disc structuring element is kind of opposite to the opening operation as smoothens corners from the outside. The main objective of this operation is to smoothen the contour and maintains the shape and size of object.

As combination of both of these operations we are able to get better results for detection edges in depth image.

5. Results and comparison

In this section we have illustrated the result of our research. We have organized the results in Fig. 4 as follows: the first pictures of each row represent the RGB image of the scene (Fig. 4a, e, and i) and the following ones denote the raw depth image of that particular scene (Fig. 4b, f, and j). Later, we have shown the outcome we have achieved by applying the median filter (Fig. 4c, g, and k). The fourth and final pictures of each row show our final results obtained by the proposed method (Fig. 4d, h, and l). In this paper we have illustrated our results for four different scenes.

For comparing our results with existing methods we have implemented a framework developed by Lejeune et al. [22] and compared our results with theirs. In the following we have shown the result of an edge detection process in similar scenes. In the figures of Fig. 5(i), (iii) and (v), we have presented the edge detection method developed by Lejeune et al. [22] and in the other figures of Fig. 5(ii), (iv) and (vi) we have shown the results of our edge detection method for the same scene sequences.

It is clearly shown that by using our technique the level of noise has been reduced significantly. And, in the figures (Fig. 5(i), (iii) and (v)) from the reference method it is also found that some of the edges were not preserved, which were preserved by our technique. To understand these areas clearly we have highlighted those areas with red circles where this type of continuous edge detection was failed on the reference method. Besides this the contours of the various shape have also been improved by using our method.

We have also measured the computation time for both of these approaches; the Ref. [22] one and our proposed one in Table 1. We have shown that in terms of computation time our method works faster than the reference one. Both of the methods were tested using the same system with an Intel Core i5 CPU, 8 GByte RAM and Intel(R) HD Graphics 4000.

6. Conclusion

In this research, we developed a framework for detecting edges in depth images. This framework consists of the following processes: acquiring depth images, depth image smoothing and edge detection in a depth image.

At first, by using some Kinect libraries the depth images were attained and converted into a desired format so that the subsequent algorithm can be applied effectively. After acquiring the depth images the level of noise was minimized by using the median filtering principle. As our proposed framework works in multiframe noises generated by each new frames such as flickering issues were also taken care of by using this smoothing technique. And, while removing noises edges were preserved as well. Finally we tried to detect the edges by applying a modified canny edge detection technique. As the canny edge detection has some concerns while applied to our smoothed depth image, by using Morphological operations we have improved the canny edge detection algorithm successfully. This technique helps to remove further noises that the smoothing operation could not take care of. Moreover, while removing the further noises it also smoothens the contour and maintains the shape and size of an object; so the shapes of the various objects in the scene were preserved.

In the result and comparison section of this paper, besides showing our results we have also made some comparison with an existing technique. It noticeably shows that our method is able to detect edges more precisely along with reducing noises in a depth image. Moreover, in terms of computing performance the proposed method was better than the conventional method.

Even though our approach works fine in multiframe one of the limitations of this approach is that the framerate does not reach up to the real time level. So, if we can increase the framerate then we can apply this technique for real time application.

As part of future work we will apply this method to some object detection applications. This technique can be used for object segmentation, object detection and other computer vision applications based on depth images. For example, by using this technique we can find desired edges of an object and by using a pre-built template, we can use the chamfer matching algorithm and detect the object properly. For detecting smaller objects in a depth image the depth image should contain less noise along with the preserved edges. So, our method can be applied robustly in this type of situations.

Conflict of interest statement

The authors have no conflict of interest to report.

Acknowledgments

This research is supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Korean Ministry of Education, Science and Technology (2011-0010099).

References

- [1] Cass TA. Robust affine structure matching for 3D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1998;**20**:1265–74.
- [2] Weiss I, Ray M. Model-based recognition of 3D object from single images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2001;**23**:116–28.
- [3] Gokturk SB, Yalcin H, Bamji C. A time-of-flight depth sensor system description, issues and solutions. In: Conference on Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. IEEE; 2004.
- [4] Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D. Object detection with discriminatively trained part based models. *Pattern Analysis and Machine Intelligence* 2010.
- [5] Malisiewicz T, Gupta A, Efros AA. Ensemble of exemplar-SVMs for object detection and beyond. In: Proceedings of ICCV. 2011.
- [6] Wang X, Yang M, Zhu S, Lin Y. Regionlets for generic object detection. In: Proceedings of ICCV. 2013.
- [7] Rovira-Mas F, Zhang Q, Reid JF. Stereo vision three-dimensional terrain maps for precision agriculture. *Computers Electronics in Agriculture* 2008;**60**(2):133–43.
- [8] Orteu J. 3-D computer vision in experimental mechanics. *Optics Lasers in Engineering* 2009;**47**(3–4):282–91.
- [9] Coudrin B, Devy M, Orteu JJ, Brethes L. An innovative hand-held vision-based digitizing system for 3D modelling. *Optics Lasers in Engineering* 2011;**49**(9–10):1168–76.
- [10] Dias P, Matos M, Santos V. 3D reconstruction of real world scenes using a low cost 3D range scanner. *Computer Aided Civil Infrastructure Engineering* 2006;**21**(7):486–97.
- [11] Cole DM, Newman PM. Using laser range data for 3D SLAM in outdoor environments. In: 2006 IEEE international conference on robotics and automation. Orlando, FL; 2006. p. 1556–63.
- [12] Xu J, Xi N, Zhang C, Shi QA, Gregory J. Real-time 3D shape inspection system of automotive parts based on structured light pattern. *Optics Laser Technology* 2011;**43**(1):1–8.
- [13] Gorthi SS, Rajshekhar G, Rastogi P. Three dimensional shape measurement using high-order instantaneous moments based fringe projection method. *Optics Laser Technology* 2011;**43**(1):40–4.
- [14] Zuo C, Chen Q, Gu GH, Feng SJ, Feng FXY, Li, RB, et al. High-speed three dimensional shape measurement for dynamic scenes using high-frequency tri-polar pulse-width-modulation fringe projection. *Optics and Lasers in Engineering* 2013;**51**(8):953–60.
- [15] Wu K, Xi JT, Yu YG, Yang ZK. 3D profile measurement based on estimation of spatial shifts between intensity ratios from multiple-step triangular patterns. *Optics and Lasers in Engineering* 2013;**51**(4):440–5.
- [16] Izadi S, Kim D, Hilliges O, Molyneaux D, Newcombe R, Kohli P, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In: Proceedings of the 24th annual ACM symposium on user interface software and technology. ACM; 2011.
- [17] Nathan Silberman, Derek Hoiem PK, Fergus R. *Indoor segmentation and support inference from rgb-d images*. In: *Proceedings of Computer Vision —ECCV*. Berlin, Heidelberg: Springer; 746–60.
- [18] Shotton J, Girshick R, Fitzgibbon A, Sharp T, Cook M, Finocchio M, Moore R, Kohli P, Criminisi A, Kipman A, Kipman A, Blake A. Efficient human pose estimation from single depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2013;**35**(12):2821–40.
- [19] Shotton J, Sharp T, Kipman A, Fitzgibbon A, Finocchio M, Blake A, Cook M, Moore R. Real-time human pose recognition in parts from single depth images. *Communications of the ACM* 2013;**56**(1):116–24.
- [20] Xia, Lu, Chia-Chih Chen, Aggarwal Jake K. Human detection using depth information by kinect. In: 2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE.
- [21] Eum Hyukmin, Yoon Changyong, Lee Heejin, Park Mignon. Continuous human action recognition using depth-MHI-HOG and a spotter model. *Sensors* 2015;**15**(3):5197–227.
- [22] Lejeune A, Pienard S, Van Droogenbroeck M, Verly J. A new jump edge detection method for 3D cameras. In: Proceedings of the International Conference on 3D Imaging (IC3D). Liege, Belgium; 2011.
- [23] Chen W, Yue H, Wang J, Wu X. An improved edge detection algorithm for depth map inpainting. *Optics and Lasers in Engineering* 2014;**55**: 69–77.
- [24] Huang T, Yang G, Tang G. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech and Signal Processing* 1979;**27**(1):13–8.
- [25] Canny John. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1986;**6**: 679–98.
- [26] Moeslund T. Canny Edge detection. Denmark: Laboratory of Computer Vision and Media Technology, Aalborg University. (http://www.cvmtdk/education/teaching/f09/VGIS8/AIP/canny_09gr820.pdf) [March 2009].
- [27] Morphological image analysis. Principles and applications by Pierre Soille. ISBN 3-540-65671-5 (1999), 2nd edition (2003).