

# A NEW JUMP EDGE DETECTION METHOD FOR 3D CAMERAS

A. Lejeune, S. Piérard, M. Van Droogenbroeck, J. Verly

INTELSIG Laboratory, Montefiore Institute, University of Liège, Belgium

## ABSTRACT

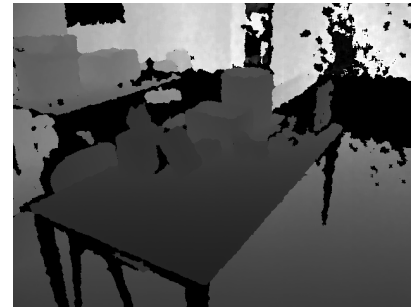
Edges are a fundamental clue for analyzing, interpreting, and understanding 3D scenes: they describe objects boundaries. Available edge detection methods are not suited for 3D cameras such as the Kinect or a time-of-flight camera: they are slow and do not take into consideration the characteristics of the cameras. In this paper, we present a fast jump edge detection technique for 3D cameras based on the principles of Canny's edge detector. We first analyze the characteristics of the range signal for two different kinds of cameras: a time-of-flight camera (the PMD[vision] CamCube) and the Kinect. From this analysis, we define appropriate operators and thresholds to perform the edge detection. Then, we present some results of the developed algorithms for both cameras.

**Index Terms**—range image, 3D cameras, edge detection, Canny, depth map, jump edge, Kinect, time-of-flight cameras

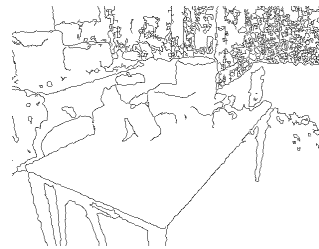
## 1. INTRODUCTION

Edge detection is one of the fundamental techniques in image processing to extract features from an image. Generally, an edge represents a discontinuity in the value of an image. For grayscale images, under common assumption about the image formation process, an edge may correspond to a discontinuity in depth, surface orientation, reflectance or illumination [8]. With a depth map or a range image, the uncertainty about the physical nature of the edge is lifted: we only have depth discontinuities. In the literature, these edges are often called *jump* edges. They are of particular importance since they correspond to boundaries of surfaces and objects. In addition, another type of edges is often defined for range images. These edges correspond to the discontinuities in the first derivative, or equivalently in the surface normals or in the surface orientation. They are called *roof* or *crease* edges. In this paper, we don't consider this type of edges.

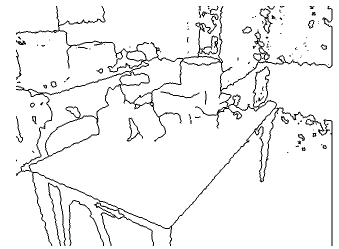
First, we briefly present existing edge detection methods for range images and explain why they are not suited to the cameras we consider. Then, we start from a standard method of edge detection, the Canny edge detector, and adapt it to our needs. We show that the assumption made on the noise by Canny is no longer valid for range images. Then, we derive a new edge detection operator that takes into account the uncertainty related to any depth value. Finally, we present



(a) Depth map.



(b) Canny edge detector.



(c) Our edge detector.

**Fig. 1:** Comparison between Canny's edge detector and our edge detector on a depth map captured with the Kinect.

results for our new edge detector.

### 1.1. Related work

There already exist edge detection methods for range images. They usually consider both jump and roof edges simultaneously. Most techniques try to fit a model on a substructure of the image to accurately compute the derivatives. Parvin *et al.* [10] fit a third-order polynomial on the neighborhood of each pixel to compute the first and second derivatives. Jiang *et al.* [5] split each line of the image into a set of quadratic polynomials. Approaches using mathematical morphology operators have also been developed [1, 7]. More recently, Coleman *et al.* [3] have developed a Laplacian operator for irregular grids to extract edges in range images. A simple criterion was used by Steder *et al.* [11] for detecting jump edges only. All these methods generally present good results on relatively simple scenes.

However, all these techniques were designed and tested for range images captured by laser ranging devices with no

concern for real-time performance since the acquisition itself is not in real-time. Also, they do not take into account the differences between each acquisition techniques and how the 3D data is obtained. Recently, real-time 3D acquisition has been made possible with time-of-flight cameras or the Kinect.

## 1.2. Canny's edge detector

The basis of our algorithm is the procedure used for the computation of Canny's edge detector [2]. It is one of the most successful algorithms developed for edge detection in grayscale images. By assuming an additive white Gaussian noise in images, Canny derived an optimal operator for edge detection that can be approximated by the derivatives of a Gaussian. The method can be summarized in four steps:

1. Noise reduction by smoothing the image.
2. Computation of the gradient.
3. Removal of non maximum values.
4. Hysteresis thresholding.

The two first steps implement a Gaussian derivative filter. First, noise is reduced in the image by convolving it with a Gaussian filter. Its standard deviation  $\sigma_{\text{filter}}$  is a parameter of the method. Then, a derivative operator is applied to the smoothed image  $I(x, y)$  for each axis. This yields the gradient  $\nabla I(x, y)$  at each pixel of the image. Any derivative operator can be used, such as a central difference or a Sobel operator.

The next step of the algorithm is to locate the pixels that are maxima of the gradient magnitude  $\|\nabla I(x, y)\|$  along the gradient of the pixel itself and to mark those pixels as possible edges. This is called non-maximum suppression because the algorithm eliminates edge points that aren't maxima. The last step, hysteresis thresholding, uses two thresholds over the gradient magnitude to only select strong edges. The two thresholds  $\tau_{inf} < \tau_{sup}$  are used sequentially. First, all maxima pixels that have a gradient magnitude over the biggest threshold  $\tau_{sup}$  are marked as edge pixels. Then, the algorithm recursively selects all locations with a gradient magnitude larger than  $\tau_{inf}$  that are adjacent to an edge pixel as edges themselves.

A raw application of Canny's algorithm on a range image captured by a time-of-flight camera or a Kinect doesn't yield good results (see Figure 1). Indeed, the algorithm assumes the same error distribution for every pixel in the image and 3D images captured with these cameras have an error distribution that changes with each pixel.

## 1.3. Overview of the proposed method

In this paper, given a range image  $D(x, y)$ , we argue that the two thresholds  $\tau_{inf}$  and  $\tau_{sup}$  should be adapted according to

the uncertainty on  $\|\nabla D(x, y)\|$  (which is related to the noise level). Here, the uncertainty on a measurement is defined as the standard deviation of the random variable that is sampled. Within a grayscale image, this uncertainty is constant, but this is not the case for range cameras. We propose to use two adaptive thresholds defined as

$$\begin{aligned}\tau'_{inf} &= \tau_{inf} + \alpha \sigma_{\|\nabla D(x, y)\|} \\ \tau'_{sup} &= \tau_{sup} + \alpha \sigma_{\|\nabla D(x, y)\|}\end{aligned}$$

where  $\alpha$  is a parameter of the algorithm, and  $\sigma_{\|\nabla D(x, y)\|}$  is the uncertainty on  $\|\nabla D(x, y)\|$ . However, in order to take this uncertainty into account also in the non-maximum suppression step of Canny's algorithm, we prefer computing an adapted value of the gradient magnitude instead of correcting the thresholds

$$\|\nabla D(x, y)\|_{\text{adapted}} = \|\nabla D(x, y)\| - \alpha \sigma_{\|\nabla D(x, y)\|}.$$

To estimate the uncertainty on  $\|\nabla D(x, y)\|$ , we start by characterizing two typical range cameras (the Kinect and the PMD[vision] CamCube) in Section 2. Then, in Section 3, we show how  $\sigma_{\|\nabla D(x, y)\|}$  is computed as a function of the uncertainty on the distance measurements.

## 2. SENSOR CHARACTERIZATION

Our goal is to determine the precision, or reciprocally the standard deviation, of each measurement made by the cameras. It is an important characteristic of the signal. It allow us to develop a better criterion for detecting edges. In the next two subsections, we briefly describe the acquisition process of both cameras and derive approximate laws for how the precision may vary inside a single range image.

### 2.1. Kinect

The Kinect can be seen as a stereoscopic camera. The part of the device responsible for depth acquisition is composed of an infrared illumination source and an infrared camera. The emitted illumination pattern is considered as the right image of a stereoscopic rig. The pattern provides a texture to every surface present in the scene. It is a matter of computing correspondences between the emitted pattern and its observation to recover depth.

Assuming that the disparity is a random variable following a normal distribution, Khoshelham [6] showed that the standard deviation of the depth  $D$  for the Kinect is:

$$\sigma_{\text{kinect}} = \kappa_{\text{kinect}} D^2, \quad (1)$$

where  $\kappa_{\text{kinect}}$  is a constant of proportionality which depends on several camera parameters and the standard deviation of the disparity.

## 2.2. Time-of-flight cameras

A time-of-flight camera derives the depth from the time of flight (round trip time) of an electromagnetic signal between the camera and the scene. In our experiment, we have used a PMD (Photonic Mixer Device) camera [12]. It works by recovering the phase shift between the emitted signal, which is modulated, and the received signal. These cameras provide three channels:

- The distance  $d$  between the camera and the corresponding point in the scene.
- The amplitude  $A$ , which measures the strength of the signal used to compute  $d$ . The amplitude is an indicator of the accuracy of the distance measurement.
- The intensity  $I$ , which measures the amount of light received by the sensor. It is similar to a grayscale image captured by a 2D camera.

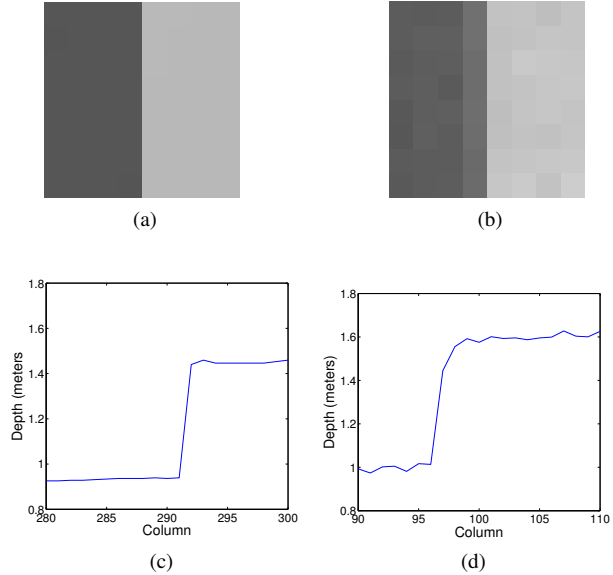
The depth precision depends on a multitude of factors including the depth, the surface orientation, and reflectance properties. Nevertheless, Frank *et al.* [4] propose a good approximation of the standard deviation of a distance measurement, expressed as

$$\sigma_{camcube}(d) = \kappa_{camcube} \frac{1}{A}, \quad (2)$$

where  $\kappa_{camcube}$  is a constant of proportionality. The precision increases with the amplitude of the signal. It should be noted that, for short distances or with long integration time, the measurements are also unreliable because of a saturation effect in the sensor. The integration time of the camera must be chosen accordingly to the considered application so that the amplitude is maximized without any saturation effect. It must be noted that the amplitude is inversely proportional to the squared depth for the same surface properties and orientation.

## 2.3. Edge characterization

Jump edges represent depth discontinuities. Depending on the method used to capture 3D information, an edge can have different physical meanings. Indeed, each range image pixel corresponds to a solid angle. Thus, the real surface represented by a single pixel depends on the depth and orientation of the surface itself. According to the method used to capture the 3D information, the depth can be sampled on a single point of the solid angle or averaged over the whole surface. So, if a depth discontinuity is present inside the solid angle corresponding to a pixel, its depth may be a mixture of the depth from both sides of the edge. Moreover, the farther the scene is, the larger the surface over which the depth is averaged will be.



**Fig. 2:** 8x8 pixels around an edge for the (a) Kinect and (b) PMD[vision] CamCube. (c) and (d) are plots of the depth values in a single row perpendicular to the edge, in the vicinity of the edge of a range image of each camera.

We have observed that the PMD[vision] CamCube time-of-flight camera displays noisy jump edges with pixels sampled from both sides of the physical edge. On the other hand, the Kinect gives clean edges. This means that the discontinuity in depth is always located between two pixels and does not spread over several intermediate pixels. An example of jump edges for both cameras is shown in Figure 2.

## 3. JUMP EDGE DETECTION

Since the error on depth may vary for each pixel, we have to adapt a step in Canny's procedure. The depth values are only used during the second step, the gradient computation. So, we adapt the gradient magnitude to the level of uncertainty of the depth values used to compute it. To achieve this, we derive an upper bound on the standard deviation of the gradient magnitude and define a new adaptive gradient magnitude.

Let  $D$  be a range image and  $D(x, y)$  denote the depth at location  $(x, y)$ . We denote the image derivatives along its first and second coordinates as  $D_x(x, y)$  and  $D_y(x, y)$ . The gradient is defined by  $\nabla D(x, y) = (D_x(x, y), D_y(x, y))$  and its magnitude by the Euclidean norm:

$$\|\nabla D(x, y)\| = \sqrt{D_x^2(x, y) + D_y^2(x, y)}. \quad (3)$$

To shorten notations, we will skip the coordinates  $(x, y)$  of the pixel unless there is a possible ambiguity. In the following developments, we assume that pixel values are correlated ( $\rho = 1$ ) if pixels are located on the same surface and independent ( $\rho = 0$ ) if pixels are separated by a jump edge.

For now, let us assume that the standard deviation of the derivatives,  $\sigma_{D_x}$  and  $\sigma_{D_y}$ , are known. Their covariance is equal to zero since the derivatives in the two orthogonal directions are independent. Therefore, we can express the standard deviation of the gradient magnitude as [9]:

$$\sigma_{\|\nabla D\|}^2 = \left| \frac{\partial \|\nabla D\|}{\partial D_x} \right|^2 \sigma_{D_x}^2 + \left| \frac{\partial \|\nabla D\|}{\partial D_y} \right|^2 \sigma_{D_y}^2$$

with

$$\frac{\partial \|\nabla D\|}{\partial D_x} = \frac{D_x}{\|\nabla D\|}.$$

So, we obtain

$$\begin{aligned} \sigma_{\|\nabla D\|} &\leq \sigma_{\|\nabla D\|}^* \\ &= \frac{|D_x| \sigma_{D_x} + |D_y| \sigma_{D_y}}{\|\nabla D\|}. \end{aligned}$$

To simplify the developments in the following, we use the upper bound  $\sigma_{\|\nabla D\|}^*$  instead of the standard deviation of the gradient magnitude  $\sigma_{\|\nabla D\|}$ . The standard deviation of the derivatives depends on the operator used to compute them. This operator is usually a linear combination of the values of the pixel and of some neighbors. Using the general equation

$$D_x(x, y) = \sum_{(u,v) \in \mathbb{N}(x,y)} c_{(u,v)} D(u, v),$$

where  $\mathbb{N}(x,y)$  is the neighborhood used to compute the derivative and  $c_{(u,v)}$  the coefficients of the derivative operator, we can compute an upper bound to its standard deviation (see the appendix):

$$\sigma_{D_x(x,y)} \leq \sum_{(u,v) \in \mathbb{N}(x,y)} |c_{(u,v)}| \sigma_{D(u,v)}. \quad (4)$$

For example, for the forward difference,

$$D_x(x, y) = D(x+1, y) - D(x, y),$$

we have:

$$\sigma_{D_x(x,y)} \leq \sigma_{D(x+1,y)} + \sigma_{D(x,y)}.$$

Now that we have derived a general upper bound for the standard deviation of the gradient magnitude, we can use it to adapt Canny's hysteresis thresholding technique:

$$\|\nabla D\| > \tau_i + \alpha \sigma_{\|\nabla D\|} \quad i = inf, sup.$$

This criteria means that the gradient magnitude must be larger than a threshold plus a certain amount of its standard deviation to be an edge. From this expression, we define the adaptive gradient magnitude:

$$\|\nabla D\|_{adapted} = \|\nabla D\| - \alpha \sigma_{\|\nabla D\|}$$

In the last equation, we adjust the gradient magnitude according to its standard deviation. In addition, we have introduced another parameter in the method,  $\alpha$ . In the implementation, it also includes the proportionality constant  $\kappa$  used in Section 2 to define the standard deviation of the depth.

From this general procedure, we describe our edge detection algorithm for the Kinect and PMD[vision] CamCube cameras. We also make some observations on the data that leads to simplification of the edge detector for the Kinect.

### 3.1. Kinect

As explained in Section 2.3, depth discontinuities in the images captured by the Kinect are clean. This means that we can obtain a single high response in the gradient magnitude by choosing the appropriate derivative operator. A forward or backward difference operator yields a single high value at the pixel preceding or following the edge. In our implementation of the algorithm, we have used a forward difference. Using such a simple derivative filter permits to compute the exact position of the edge.

Using the derivatives, we compute the adaptive gradient magnitude:

$$\begin{aligned} \|\nabla D(x, y)\|_{adapted} &= \|\nabla D(x, y)\| \\ &- \frac{\alpha}{\|\nabla D(x, y)\|} (|D_x(x, y)| (D^2(x, y) + D^2(x+1, y)) \\ &\quad + |D_y(x, y)| (D^2(x, y) + D^2(x, y+1))). \end{aligned}$$

We have noticed that the non-maximum suppression step is unnecessary with the Kinect. Indeed, the device is unable to compute the depth on surface that have a high gradient. Thus, using appropriate thresholds  $\tau_{inf}$  and  $\tau_{sup}$ , the pixels having a gradient magnitude superior to them can be considered as edge pixels. The suppression of this step speeds up the algorithm as we avoid at least two comparisons for each pixel. There is no need to apply any other noise reducing method prior to the computation of the gradient. Its adapted version is robust enough to avoid it.

### 3.2. Time-of-flight cameras

The edges are harder to detect with a time-of-flight camera. The complete algorithm requires all the steps of Canny's edge detector. First, we smooth the range image with a Gaussian. Then, we compute the derivatives with a central difference. Thus, the adaptive gradient magnitude is given by:

$$\begin{aligned} \|\nabla D(x, y)\|_{adapted} &= \|\nabla D(x, y)\| \\ &- \frac{\alpha}{\|\nabla D(x, y)\|} \left( |D_x(x, y)| \left( \frac{1}{A(x-1, y)} + \frac{1}{A(x+1, y)} \right) \right. \\ &\quad \left. + |D_y(x, y)| \left( \frac{1}{A(x, y-1)} + \frac{1}{A(x, y+1)} \right) \right). \end{aligned}$$

The non-maximum suppression must be applied before performing the hysteresis thresholding. The thresholds are determined experimentally, according to the integration time and the captured scene.

## 4. RESULTS

We would have compared our results with other methods or evaluate it with a ground-truth but, to our knowledge, there are no other publications or databases about edge extraction dedicated to the kind of 3D cameras we have considered. Therefore, we only provide qualitative results. In Figures 3 and 4, we show the results of our edge detector on an indoor scene with a large range of depths. In Figures 5 and 6, our algorithm is applied on another scene.

The extracted edges for both cameras reflect the properties of each camera. With the Kinect, the edges follow exactly the detected depth discontinuities as well as the area where the device was unable to compute the depth (black zones in Figure 3a). It must be noted that, in these zones, the depth is assumed to be zero and thus the standard deviation is also equal to zero. On the contrary, the PMD[vision] CamCube measures depth for every single pixel, some with less confidence than others. We have shown that our criteria provides good results with respect to the value of the amplitude image. For example, in Figure 4a, despite having some very noisy measurements (indicated by the red rectangle), the edges are relatively well extracted.

The PMD[vision] CamCube camera also has a physical limitation on its maximum depth measurement. Typically, for a 20 Mhz modulation frequency, the maximum range is 7.5 m. Therefore, an edge may be detected at this distance. However, it should be noted that our algorithm behaves as expected for larger distances.

Typical execution time for our algorithm on a Intel Core i7 processor is 15  $\mu$ s for the Kinect (640 $\times$ 480 pixels) and 7  $\mu$ s for the PMD[vision] CamCube (204  $\times$  204 pixels) with a Gaussian blur with a standard deviation of  $\sigma_{\text{filter}} = 1.5$ .

## 5. CONCLUSION

We have developed a new jump edge detection algorithm for 3D cameras based on the properties of the devices. Our algorithm is fast and its principle can be adapted to any other 3D sensing device. The Kinect shows better properties than the PMD[vision] CamCube camera for edge detection and thus, its version of the algorithm is more efficient. We have introduced an additional parameter,  $\alpha$ . It can be interpreted as the constant of proportionality between the standard deviation of the gradient magnitude and the values used to compute it (squared depth for the Kinect and the amplitude inverse for the PMD[vision] CamCube).

In this paper, we only have detected the jump edges and a roof edge detection technique still needs to be developed for

these cameras.

## Acknowledgements

A. Lejeune is under a contract funded by the FEDER program of the Walloon Region, Belgium.

## 6. APPENDIX: STANDARD DEVIATION OF THE DERIVATIVES

We define a derivative operator as

$$D_x(x, y) = \sum_{(u,v) \in N(x,y)} c_{(u,v)} D(u, v),$$

where

$$\begin{cases} c_{(u,v)} \leq 0 & \forall u < x \\ c_{(x,v)} = 0 \\ c_{(u,v)} \geq 0 & \forall u > x \end{cases}$$

and  $N_{(x,y)}$  is a neighborhood around the pixel  $(x, y)$  that can be divided between a left neighborhood and a right neighborhood:

$$\begin{aligned} N_{(x,y)}^{\text{left}} &= \{(u, v) \in N_{(x,y)} | u < x\} \\ N_{(x,y)}^{\text{right}} &= \{(u, v) \in N_{(x,y)} | u > x\}. \end{aligned}$$

Without making any assumptions, the standard deviation can be expressed as [9]:

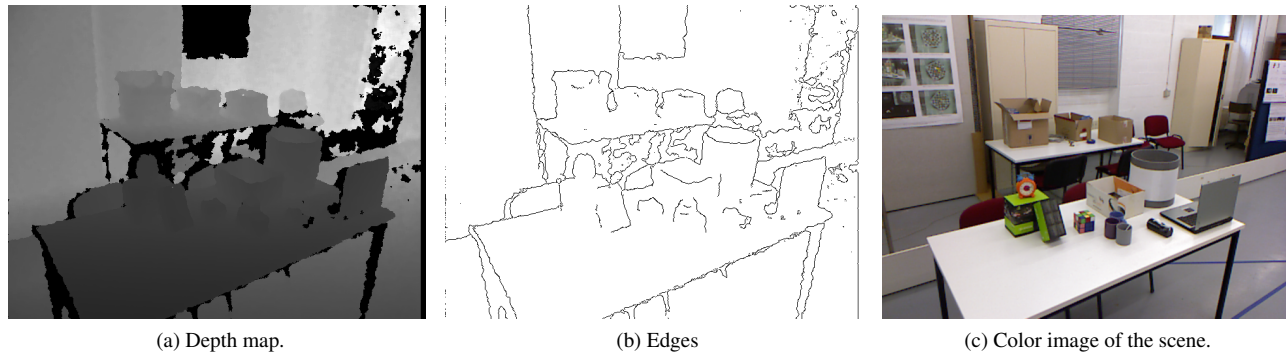
$$\begin{aligned} \sigma_{D_x(x,y)}^2 &= \sum_{(u,v) \in N(x,y)} c_{(u,v)}^2 \sigma_{D(u,v)}^2 + \\ &\sum_{(u,v) \in N(x,y)} \sum_{\substack{(m,n) \in N(x,y) \\ (m,n) \neq (u,v)}} c_{(u,v)} c_{(m,n)} \rho_{(u,v)(m,n)} \sigma_{D(m,n)} \sigma_{D(m,n)}, \end{aligned} \quad (5)$$

where  $\rho_{(u,v)(m,n)}$  is the correlation between the pixels  $(u, v)$  and  $(m, n)$ .

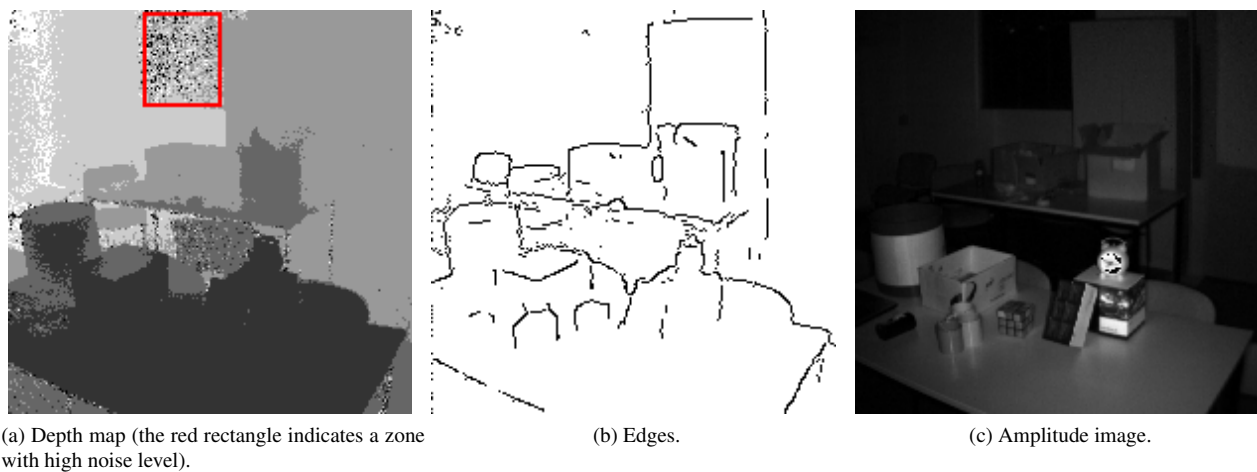
We will show that the standard deviation is maximum when there is an edge at pixel  $(x, y)$  where we are computing the derivative. Under such circumstances, the correlation between the pixels of the left neighborhood and the pixels of the right neighborhood is zero: the pixels belong to different surfaces. Moreover, if we assume that the pixels of a same surface are correlated ( $\rho = 1$ ), the standard deviation becomes

$$\begin{aligned} \sigma_{D_x(x,y)}^2 &= \left( \sum_{(u,v) \in N_{(x,y)}^{\text{right}}} c_{(u,v)} \sigma_{D(u,v)} \right)^2 \\ &+ \left( \sum_{(u,v) \in N_{(x,y)}^{\text{left}}} c_{(u,v)} \sigma_{D(u,v)} \right)^2. \end{aligned} \quad (6)$$

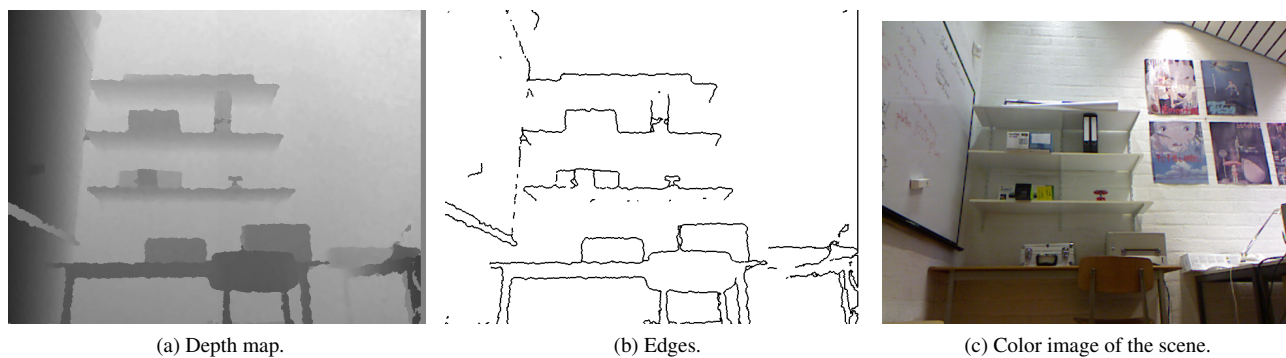




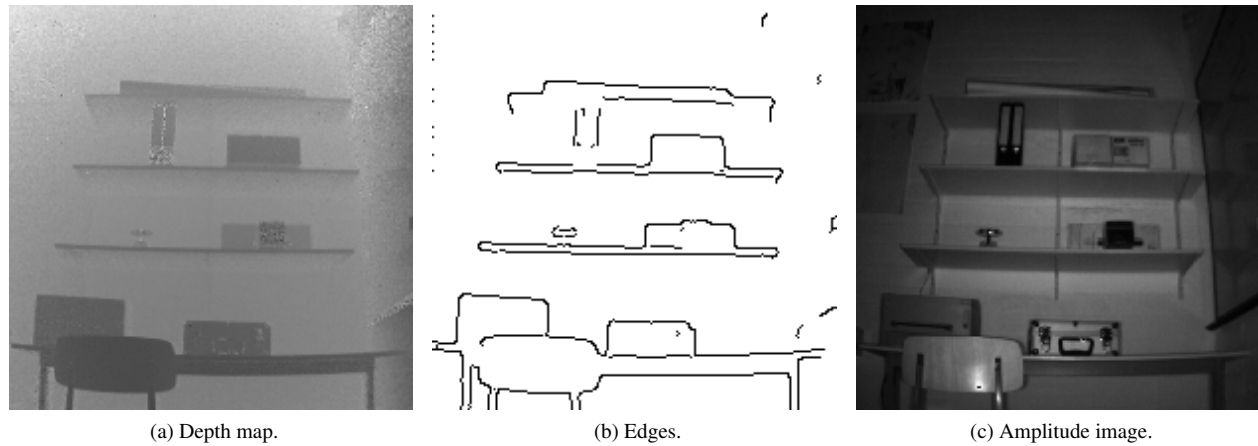
**Fig. 3:** Example of edge detection for the Kinect ( $\sigma_{\text{filter}} = 0$ ,  $\alpha = 0.004$ ,  $\tau_{\text{inf}} = 0.008$ ,  $\tau_{\text{sup}} = 0.03$ ).



**Fig. 4:** Example of edge detection for the PMD[vision] CamCube ( $\sigma_{\text{filter}} = 1.5$ ,  $\alpha = 8$ ,  $\tau_{\text{inf}} = 0.01$ ,  $\tau_{\text{sup}} = 0.02$ , integration time of  $2\text{ ms}$ )



**Fig. 5:** Example of edge detection for the Kinect ( $\sigma_{\text{filter}} = 0$ ,  $\alpha = 0.004$ ,  $\tau_{\text{inf}} = 0.008$ ,  $\tau_{\text{sup}} = 0.03$ ).



**Fig. 6:** Example of edge detection for the PMD[vision] CamCube ( $\sigma_{\text{filter}} = 1.5$ ,  $\alpha = 6$ ,  $\tau_{\text{inf}} = 0.01$ ,  $\tau_{\text{sup}} = 0.03$ , integration time of 1 ms).

Equation 6 contains all the terms of Equation 5 where  $c_{(u,v)}c_{(m,n)} > 0$ . Equation 6 is thus a superior bound on  $\sigma_{D_x}(x,y)$ . By triangular inequality, we obtain Equation 4.

## 7. REFERENCES

- [1] P. Boulanger, F. Blais, and P. Cohen. Detection of depth and orientation discontinuities in range images using mathematical morphology. In *IEEE International Conference on Pattern Recognition (ICPR)*, volume 1, pages 729–732, June 1990.
- [2] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.
- [3] S. Coleman, B. Scotney, and S. Suganthan. Edge detecting for range data using laplacian operators. *IEEE Transactions on Image Processing*, 19(11):2814–2824, November 2010.
- [4] M. Frank, M. Plaue, H. Rapp, U. Kothe, B. Jahne, and F. Hamprecht. Theoretical and experimental error analysis of continuous-wave time-of-flight range cameras. *Optical Engineering*, 48(1), January 2009.
- [5] X. Jiang and H. Bunke. Edge detection in range images based on scan line approximation. *Computer Vision and Image Understanding*, 73(2):183–199, February 1999.
- [6] K. Khoshelham. Accuracy analysis of Kinect depth data. In *ISPRS Workshop Laser Scanning 2011*, volume XXXVIII-5/W12, Calgary, Canada, August 2011.
- [7] R. Krishnapuram and S. Gupta. Morphological methods for detection and classification of edges in range images. *Journal of Mathematical Imaging and Vision*, 2:351–375, November 1992.
- [8] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30:117–156, November 1998.
- [9] A. Papoulis. *Probability, random variables, and stochastic processes*. McGraw-Hill, 1991.
- [10] B. Parvin and G. Medioni. Adaptive multiscale feature extraction from range data. *Computer Vision, Graphics, and Image Processing*, 45(3):346–356, March 1989.
- [11] B. Steder, R. Rusu, K. Konolige, and W. Burgard. Point feature extraction on 3D range scans taking into account object boundaries. In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 2011.
- [12] Z. Xu, R. Schwarte, H. Heinol, B. Buxbaum, and T. Ringbeck. Smart pixel –photonic mixer device (PMD)– new system concept of a 3D-imaging camera-on-a-chip. In *International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, pages 259–264, Nanjing, China, September 1998.