Fast Edge Detection in RGB-D Images

Heriberto Casarrubias-Vargas[1], Alberto Petrilli-Barceló[2],
and Eduardo Bayro-Corrochano[1]

[1] CINVESTAV Unidad Guadalajara, Jalisco, México
{hvargas,edb}@gdl.cinvestav.mx
[2] Universidad Autónoma del Carmen, Campeche, México
apetrilli@pampano.unacar.mx

Abstract. The understanding of scenes is a key aspect of computer vision. Edge detection helps us to understand more about the scene structure since the edges mark a clear distinction for a transition from one region with similar properties to another one. When the edges are obtained from changes in orientation, we can use them to find key planes and describe the scene. This paper describes a method for fast edge detection in RGB-D images. The edge detection algorithm for depth images is based on the use of smooth constraints in orientation. We show experimental results that demonstrate the potential of the approach proposed for edge detection.

1 Introduction

The edges of objects are useful for describing objects in terms of their borders. If it is possible to find the boundary of an object by locating all of its edges, then we can segment it and give an interpretation of this.

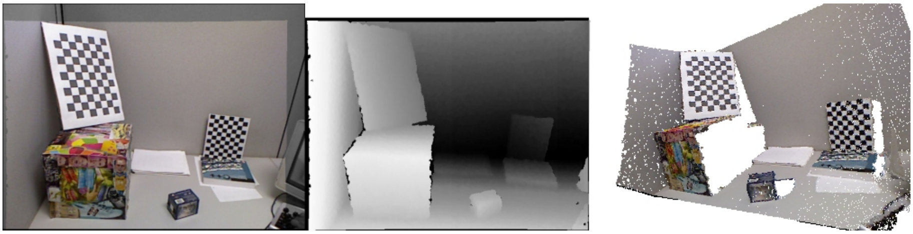
The edge-detection problem has been presented in previous publications. The edges are used as a previous step to cloud point segmentation obtained from the range images. Typically, edge detection is based on normal computations and growing regions [5], [4], [8], [11]. In [7], the authors give an algorithm for range image segmentation based on growing regions using surface normals as a measure of the local geometry, which are defined by fitting a plane to the neighborhood of the point; they detect and mark the edges as changes in the surface normal to the planes by a threshold given by τ. In [3], a method for image segmentation is based on edges, and the algorithm to detect edges is based on scan line processing. The method approximates each scan line via a set of quadratic polynomial functions; the algorithm is iterative and involves a high computational cost. The work in [10] proposes an algorithm to edge detection based on growing regions; they build a mesh from data using Delaunay triangulation and compute angles between normal vectors of meshes. If the angle change in a neighborhood is greater than a tolerance, then the point is regarded as an edge.

The segmentation process in range images can be based on edges followed by grouping the points inside the boundaries given by the edges. Edges are regions in the surface that have an abrupt change in orientation; the neighborhood of no-edge points can be considered locally as a plane. The edge points can be locally considered as a meeting of the two planes. For several incident planes, the edge point is clearly distinguished because it is locally a maximum or minimum.

E. Bayro-Corrochano and E. Hancock (Eds.): CIARP 2014, LNCS 8827, pp. 868–875, 2014.

2 The RGB-D Sensor

In this work, the Microsoft Kinect sensor is used with RGB-D images of size 640 × 480. An RGB-D image is composed of two images $I_{\text{RGB-D}} = (I_{rgb}, I_{depth})$ where I_{rgb} denotes a traditional three-channel RGB image and I_{depth} denotes a one-channel range image. The depth information of a pixel (i, j) in the RGB image is given by the pixel (i, j) in the range image. The RGB and range images are calibrated. The intrinsic parameters of the camera are known because of calibration done beforehand.



(a) RGB image. (b) Depth image. (c) Cloud of points in 3D.

Fig. 1. (a) RGB image; the pixels without depth information are shown with less brightness. b) Depth image, the dark pixels correspond to faraway points and the white to close points. (c) The cloud of points in 3D space; the geometric reconstruction of the scene.

The Kinect sensor gives the necessary information for the reconstruction process of the scene. For each pixel (u, v) in the RGB image and their depth d, it is possible to know their 3D position (X, Y, Z) in the space using the transformation

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} \frac{(u - c_{x_d})d}{f_{x_d}} \\ \frac{(v - c_{y_d})d}{f_{y_d}} \\ d \end{pmatrix} = \Pi^{-1}(u, v), \tag{1}$$

where $f_{x_d}, f_{y_d}, c_{x_d}, c_{y_d}$ are the intrinsic parameters of the depth camera. Therefore, from the RGB-D image, we can associate a cloud of points in the 3D space that is the reconstruction of the scene. Figure 1 shows the RGB image and its corresponding cloud of 3D points; we will denote the cloud of 3D points by $\mathscr{X}_{\text{RGB-D}}$, where

$$\mathscr{X}_{\text{RGB-D}} = \{P_{(u,v)} \mid (u, v) \in I_{depth} \text{ and } P_{(u,v)} = \Pi^{-1}(u, v).\} \tag{2}$$

The cloud of points $\mathscr{X}_{\text{RGB-D}}$ is indexed in a natural way by the pixels in the depth image; the indexes are determined by the transformation

$$P_{(u,v)} = \Pi^{-1}(u, v) \tag{3}$$

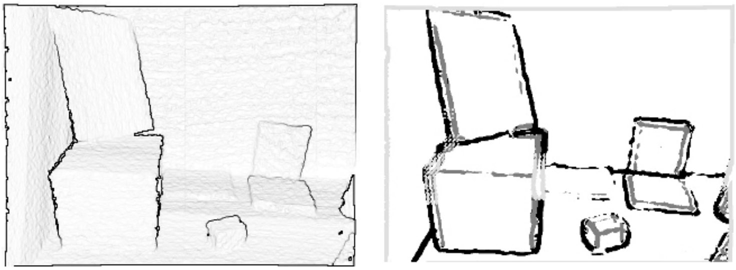
Due to the calibration assumption, the indexation is indistinct for the pixel in the RGB or range image, and for simplicity we denote the indexation only by $\{P_{(u,v)}\}_{\substack{0 \leq u \leq 640 \\ 0 \leq v \leq 480}}$. The indexation is very important because given a point $P_{(u,v)}$ in the cloud of points $\mathscr{X}_{\text{RGB-D}}$, their neighborhood points can be determined using this indexation. This property is used in the algorithms developed in this paper.

3 Edge Detector for RGB-D Images

The edge detection is an important problem in analysis of the image to recover the information about the scene. Detect the presence of these discontinuities in the image allows us to understand the scene under certain conditions. The early works in this topic are Edge detection is basically a method to divide an image into regions based on discontinuity, i.e., it allows to enhance those features where there is significant changes in the image, indicating the end of one region in the image and the beginning of another one.

In RGB images the edge detection process makes use of differential operators to detect changes in the gradients of the gray or color levels and mathematically speaking an ideal edge in RGB images is a discontinuity of the spatial gray value function of the image. In three dimensions, volumetric regions are separated by surfaces and edges become discontinuities in the orientation of surfaces that have been mapped in the depth image plane.

However, the traditional methods to edge detection in RGB images like Canny [3] or Sobel [10] can not detect the edges that are of interest in depth images. This is due the region of interest are different. In RGB images the edges are changes in color and in depth images the edges correspond to abrupt changes in orientation of the surface and discontinuity in depth to the mapped scene. The Sobel operator only detect discontinuity in depth as is show in the Fig. 2a, but not detect orientation changes in the surface.



(a) Edge Image using Sobel. (b) Edge Image using the algorithm 1.

Fig. 2. a) Shows the edges using Sobel, note that edges detected correspond to the discontinuity in depth. The Fig. b) shows the edges detected, black concave edges, dark gray convex edges and light gray edges without depth.

The algorithm proposed uses the implicit order induced by the pixels coordinates (u, v) in the RGB image over the cloud of points $\mathscr{X}_{\text{RGB-D}}$ and collinearity properties derived of the 3D scene projection over 2D image plane induced by the camera.

Using the collinearity property we can obtain the next result: Let X a point in the cloud of points $\mathscr{X}_{\text{RGB-D}}$ and suppose that X locally belong a plane. Then exist a neighborhood V compact for X such as for all $Y \in V$, X and Y are in the same plane. Hence exist a segment of line $\overline{AB}$ with $A, B \in V$ such as contain X. Due to the transformation Π

induced by the camera RGB-D is a projective transformation the segment of line $\overline{AB}$ is mapped to a segment of line $\overline{ab}$ in the image plane [6], where a and b are the projections of A and B respectively.

Let a,b and c collinear points in the depth image plane and let A,B and C their corresponds points in $\mathbb{R}^3$. If A,B and C are not aligned then there is not a plane that contain them. Hence exist an orientation change in the surface given by the cloud of points $\mathscr{X}_{\text{RGB-D}}$. This result is the core of the algorithm proposed. The next problem to solve is: how to measure efficiently changes in orientation using this result? To solve this problem, it should be noted that $\mathbb{R}^3$ is a normed space and the triangular inequality is satisfied.

Let A,B and X points in $\mathbb{R}^3$ then the triangular inequality is satisfied:

$$||A-B|| \leq ||A-X|| + ||B-X|| \tag{4}$$

and exist $\delta \geq 0$ such as

$$||A-B|| + \delta = ||A-X|| + ||B-X|| \tag{5}$$

then

$$1 = \frac{||A-X|| + ||B-X||}{||A-B|| + \delta} \tag{6}$$

Now, A,B and X are collinear if and only if $\delta = 0$. In this case

$$1 = \frac{||A-X|| + ||B-X||}{||A-B||} \tag{7}$$

If $\delta > 0$ the points A,B and X are not collinear, then we have

$$1 = \frac{||A-X|| + ||B-X||}{||A-B|| + \delta} < \frac{||A-X|| + ||B-X||}{||A-B||} \tag{8}$$

Hence

$$1 < \frac{||A-X|| + ||B-X||}{||A-B||} \tag{9}$$

Equation (9) is used to discriminate when three points are collinear. Using the order induced in the cloud of points $\mathscr{X}_{\text{RGB-D}}$;

Let $x = (u,v)$ a point in the depth image and let be $x_a = (u-\kappa, v)$, $x_b = (u+\kappa, v)$ their neighborhoods to a distance κ and let $X = \Pi^{-1}(x)$, $A = \Pi^{-1}(x_a)$, $B = \Pi^{-1}(x_b)$ their respective points in the cloud of points $\mathscr{X}_{\text{RGB-D}}$, see Fig. 3. The equation 9 is usedto determine when an orientation change is presented at the point X. A change in the orientation of the surface represent an edge in the depth image.

The equation 9 can be used to determine if an edge is concave or convex. The test is based in the relative distance to the camera. If the norm of the point $\frac{A+B}{2}$ is less that the norm of X then the edge is convex. Otherwise, if the norm of the point $\frac{A+B}{2}$ is greater that the norm of X the edge is concave.

The previous observations are summarized in the algorithm 1 that compute edges in RGB-D images.

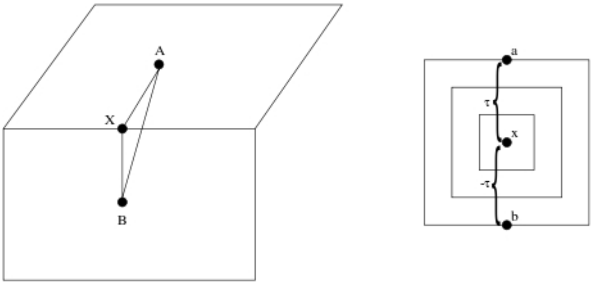
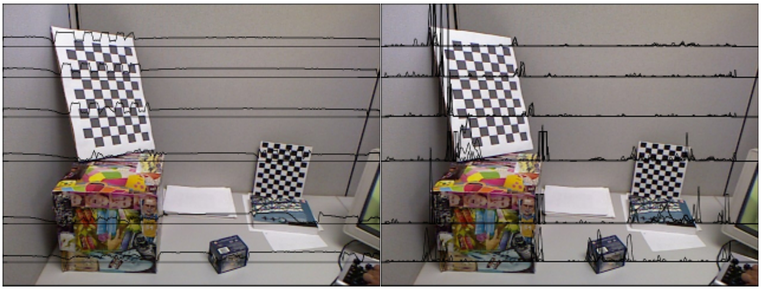


Fig. 3. The left figure outline the arrangement of neighborhood in 3D for the point X. The right side shows the neighborhood for the pixel x in 2D with a distance of τ.

The Fig. 2 shows the result after apply the algorithm 1 to a RGB-D image. The resulting image is labeled with edges convex, concave and false edges due to regions without depth information given by the sensor limitations or occlusions.

The Fig. 4 a) shows a line profile for the intensity changes in the RGB image; note that the profile line can not detect the changes in the orientation of surface. The Fig. 4 b) shows the profile line that is computed using the discriminate given in 9; this profile shows the changes in the orientation of the 3D surface associated to the depth image. The Fig. 4 b) shows the profile line to 4 orientations: horizontal, vertical y the two diagonals.



(a) Line profile for intensity image. (b) Line profile for the depth image.

Fig. 4. The line profiles for intensity and depth images are shows over a RGB image for reference

4 Experimental Results

The edge-detection algorithm 1 for RGB-D images is sensitive to noise due to the process of sensing and the reflectance of the material, illumination, etc. Therefore, to reduce the noise, an averaging filter is applied to the cloud of points $\mathscr{X}_{\text{RGB-D}}$ to use the algorithms proposed here. They can tolerate noisy points that do not actually belong to the surface sampled.

```
Data: $\mathscr{X}_{\text{RGB-D}}$ : Cloud of points
      $\kappa$ : Radio
      $T$ : Threshold
      $H$ : Height image
      $W$ : Width image
      $L$ : Edge gray value
Result: $I_{edges}$ Edge image for depth image
foreach $(u,v) \in \{0,..,W\} \times \{0,..,H\}$ do
    $X \leftarrow X_{(u,v)}$
    $\lambda_{max} \leftarrow -\infty$
    $\tau \leftarrow 0$
    foreach $(\kappa_1,\kappa_2) \in \{(\pm\kappa,\pm\kappa),(0,\pm\kappa),(\pm\kappa,0)\}$ do
        $A \leftarrow X_{(u+\kappa_1,v+\kappa_2)}$
        $B \leftarrow X_{(u-\kappa_1,v-\kappa_2)}$
        $\lambda_{max} \leftarrow \texttt{maximo}\left(\frac{\|A-X\|+\|B-X\|}{\|A-B\|}, \lambda_{max}\right)$
        $\tau \leftarrow \tau + \begin{cases} 1, & \text{if } \frac{\|A+B\|}{2} < \|X\| \\ -1, & \text{otherwise} \end{cases}$
    end
    if $\lambda_{max} > T$ then
        $I_{edges}(u,v) \leftarrow \begin{cases} L, & \text{if } \tau < 0 \\ L/2 & \text{otherwise} \end{cases}$
    else
        $I_{edges}(u,v) \leftarrow 0$
    end
end
```

Algorithm 1: Edges detection in RGB-D images

As a measurement of the quality of the edge detection, we take the dispersion of the points around the line that represent the edge after filter data. The quality is measuremed with respect to an edge line with an angle of aperture θ; theta was varied from 45-140° with a frontal viewpoint. The radius for the neighborhood was set to 11 and the distances to the edge were varied in the experiment. Fig. 5 shows the results obtained. The dispersion is considered around the edge line detected, as shown in Fig. 5 (a)-center.

All algorithms described here were tested using a Core 2 Duo processor @ 2.0 GHz using a single thread on a single core. The implementation of the algorithm 1 described in Section 3 to detect edges in RGB-D images was executed in 7 ms. The algorithms were tested using images of size 640 × 480 and processing all the points. The authors think that this time can be improved with a better implementation. The time to smooth the data was not considered in this estimated time. Using integral images, the time to compute the filter averaging was 2 ms.

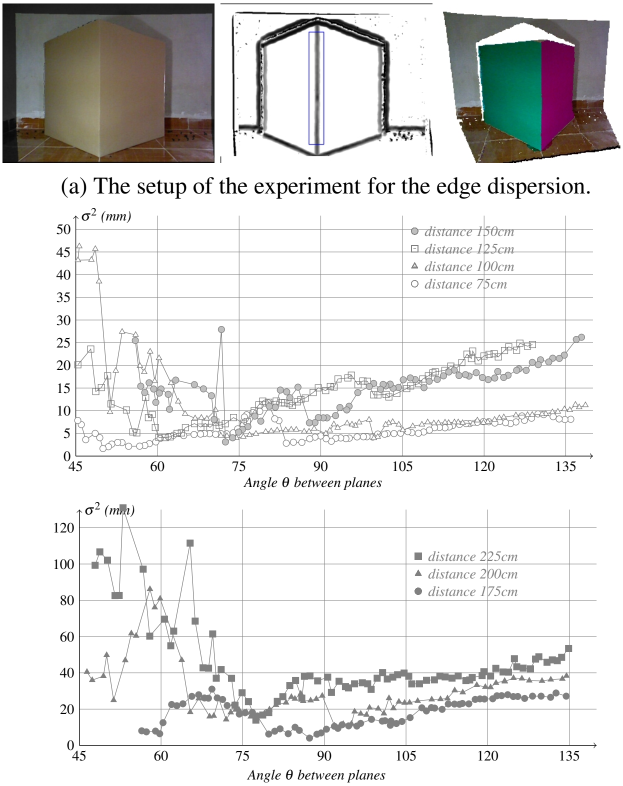

(a) The setup of the experiment for the edge dispersion.

Fig. 5. Standard deviation for a edge points generated by the meet of planes with a angle θ between them. (a)-left the RGBD image, (a)-center shows in blue the ROI used to compute the standard deviation, (a)-right the 3D view of the planes used to generate the reference edge line. (b) and (c) the results obtained.

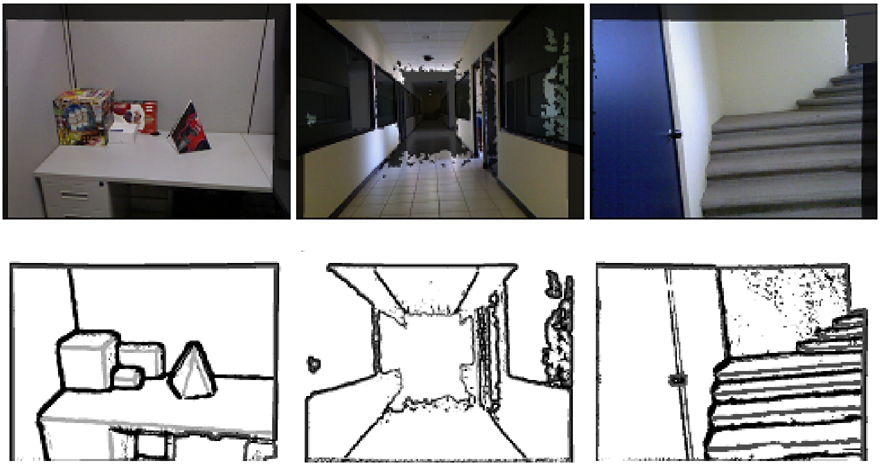


Fig. 6. Edges detected in typical scenes, the value for smoothing is 7 and that for the radius in algorithm 1 is 7

5 Conclusions

The algorithms that have been presented in this paper have the potential to recover the scene structure. They are based on an approach that uses the triangle inequality to detect changes in the orientation of the surface. The discriminant proposed for edge detection is easy to compute and responds to the changes in the orientation of the surface. Due to the noise in the data, the edges detected are thick; however, they correctly detect the orientation changes on the surface and present stability to rotations in the viewpoint for the same depth. The method is sensitive to the depth of the surface and the neighborhood uses of the algorithms.

References

1. Camillo, J.T., Cowley, A.: Fast scene analysis using image and range data. In: ICRA 2011, pp. 3562–3567 (2011)
2. Camillo, J.T., Cowley, A.: Segmentation and Analysis of RGB-D data. In: RGB-D: Advanced Reasoning with Depth Cameras. Workshop (2011)
3. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Machine Intell. 8(6), 679–698 (1986)
4. Checchin, P., Trassoudaine, L., Alizon, J.: Segmentation of Range Images into Planar Regions. In: 3DIM 1997, pp. 156–163 (1997)
5. Fu, C., Chen, C.-J., Lu, C.-J.: A linear-time component-labeling algorithm using contour tracing technique. Computer Vision and Image Understanding 93, 206–220 (2004)
6. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press (2004)
7. Holz, D., Holzer, S., Rusu, R.B., Behnke, S.: Real-Time Plane Segmentation using RGB-D Cameras. In: Röfer, T., Mayer, N.M., Savage, J., Saranlı, U. (eds.) RoboCup 2011. LNCS, vol. 7416, pp. 306–317. Springer, Heidelberg (2012)
8. Lee, Y., Park, S., Jun, Y., Choi, W.C.: A robust approach to edge detection of scanned point data. The International Journal of Advanced Manufacturing Technology 23, 263–271 (2004)
9. Marr, D., Hilderth, E.: Theory of edge detection. Proc. R. Soc. London 207, 187–217 (1980)
10. Sobel, I., Feldman, G.: A 3x3 Isotropic Gradient Operator for Image Processing. In: Duda, R., Hart, P. (eds.) Pattern Classification and Scene Analysis, pp. 271–272. Wiley, New York (1973)
11. Trevor, A., Rogers III, J., Nieto-Granada, C., Christensen, H.I.: Tables, Counters, and Shelves: Semantic Mapping of Surfaces in 3D. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010) Workshop on Semantic Mapping and Autonomous Knowledge Acquisition (2010)