Object Detection and Comparison of Different Shapes and Materials using Kinect

Nancy Velasco[1], David Rivas L.[2] and Eddie E. Galarza[2]
[1]*Departamento de Ciencias Exactas,*
[2]*Departamento de Eléctrica y Electrónica,*
Universidad de las Fuerzas Armadas ESPE, Sangolqui, Ecuador.
ndvelasco@espe.edu.ec

***Abstract*—This paper presents an algorithm for object detection and an evaluation of response with different shapes and materials using Kinect sensor. In order to develop this work, a new icon is done using LabVIEW. The depth image of the Kinect is processed by Artificial Vision Toolkit to indicate the distance to each object. Additionally, the application has an audio output in English and Spanish indicating whether an object is in the trajectory. Several tests were done, through which the performance of the proposal was verified.**

***Index Terms*—Kinect; Object Detection; LabVIEW Vision Toolkit; Microsoft Kinect SDK.**

I. INTRODUCTION

Artificial vision, also known as computer vision simulates the vision sense of a human by programming a computer to get all the information from a digital image to perform a specific task. Many methods are implemented for object detection. In [1] object recognition is made by a selective search. Reza Oji in [2] presented a method to recognize objects with full boundary detection. This algorithm combines a region merging and affine scale invariant feature transform. In [3] a machine learning approach for visual object detection was made. It is capable to process images in an extremely rapidly way with high detection rates. In [3], a machine learning approach for visual object detection was made, which is capable to process images in an extremely rapid way, achieving high detection rates. C Papageorgiou & T Poggio [4] presented a general and trainable system for object detection in unconstrained, cluttered scenes. All of these researches use a camera without depth information.

The appearance of Microsoft Kinect has been a resounding success in the computer world; many research teams are developing different applications to increasingly varied fields, away from the field of videogames that was their initial objective [5]. Thanks to this sensor that is powerful and cheaper, many people make productive developments, such as the motion detection to create a method to distinguish between obstacles and the manipulator when they share the same workspace [6], human detection using depth information by Kinect [7], study on the use of Microsoft Kinect for robotics applications [8],[9], mobile robots navigation in indoor environments [10], method to recognize gestures through the Kinect depth camera [11], depth-supported real-time video segmentation with the Kinect [12], or to use it in the education [13] and health [14] areas.

Many of the systems that take computer vision to detect objects do not do it very well and has time lag. Some of them are not able to define the distance where these objects are located.

The algorithm that was implemented in this research uses the edges of the objects that are present in the image to detect them and then, after detected, the algorithm uses special conditions to define the distance where the objects are located. The use of Kinect is fundamental to define the presence of the objects and the distance of their location.

The algorithm could give assistance to mobile robots operated in indoor environments and is able to support blind people to make the recognition of its environment offering a greater independence to the user.

This work is divided into four sections: Section II deals with the depth capture and the implementation of the algorithm. Section III presents the results, and finally the conclusions are presented in Section IV.

II. EXPERIMENTAL

A. Technique for Capturing the Object Depth

The camera uses an infrared laser projector that casts a pattern of 50000 points invisible to the human eye. With the bouncing of the objects in the scene, the pattern of dots is captured by the infrared camera that is 7.5 cm of separation from the projector. The integrated circuit of Kinect analyzes the disparity caused by the objects in the scene.

Depth information is returned in a map of pixels with a maximum frequency of 30 images per second. If the value of a pixel in the map is zero, this means that the sensor was unable to estimate the depth in that region because of limitations of the sensor. Each pixel has the Cartesian location and the distance (millimeters) from the plane of the camera to the nearest object in a particular position with coordinates (x, y), which represent the pixel location in the image depth, but no the physical units in the room. The depth data size is 12 bits that is displaced to the left by 3 bits. [15] (Figure 1). The Microsoft Kinect SDK is the main development kit released by the Microsoft Corporation. The Microsoft documentation stablishes that the range of distances for a valid depth sensor is from 4 to 11.5 feet (1.2 to 3.5 meters).

Figure 1: Location of the depth bits in Microsoft Kinect frame

B. Algorithm Phases

The algorithm is divided in seven phases, which are concatenated consecutively to achieve the objectives of the project. These are summarized in Figure 2.

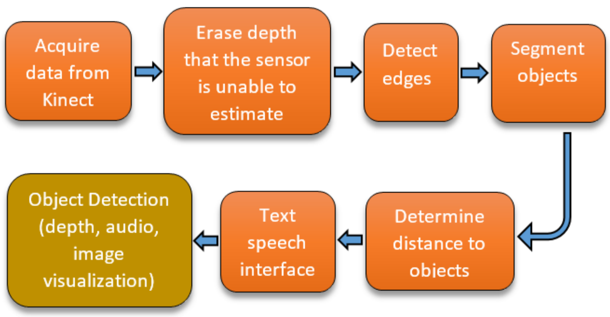


Figure 2: Block diagram of the processes of the algorithm

The algorithm used in this work include the following steps:

i. Acquire data from Kinect.
ii. Erase value that the sensor is unable to estimate.
iii. Detect Edges
iv. Score object detection based on the edges detection
v. Calculation of distance of the object

Acquire data from Kinect. The Kinect interface with LabVIEW is based on the Kinect SDK v1.0.zip [16]. We used only the tools that are necessary for the project: initializing the Kinect, setting the angle of tilt, getting RGB images, calculating depth and Stopping the Kinect. Data were three positions rotated. Then, only the bits that represent the depth excluding those that represent the number of players were extracted.

Erase value that the sensor is unable to estimate. The object position where distance could not be calculated by the sensor was eliminated (pixels with zero values).

Detect edges. A binary image was obtained by thresholding the depth image. This digital image has only two possible values for each pixel. They were displayed with colors: black(value=0) and white(value= 1 or 255). A conceptually simple but effective noise-smoothing algorithm was used for edge detection. The filter [17] conducted by Lee, Jong-Sen was motivated by the sigma probability of the Gaussian distribution, and it smooths the image noise by averaging only those neighborhood pixels which have the intensities with a fixed sigma range from the center pixel. Image edges were kept, and some thin lines and details were retained.

Segmentation and distance to object determination. After edge detection, using LabVIEW tools, each closed edge was filled for segmenting and counting. The algorithm separated the coordinates of the center and corners of each object. Then, it read the depth information of the central pixel in the object. Rectangles that represent the objects were placed over the original image.

Text speech interface. The text to speech interface [18] used ActiveX controls and property nodes to convert a written text in LabVIEW to an audible voice. Previously, the texts were created and they were available in Spanish and English, the text information was about total obstacle, track free and front obstacle.

C. LabVIEW Icon

The LabVIEW icon has the following entries.

- Maximum distance to be detected with 10 cm steps.
- Range image division to consider above space as the center of the path.
- Boolean control to use text to speech.
- Control to select the available voices in the PC to be used.
- Control to select English or Spanish voices.

The algorithm has the following outputs:

- A numeric indicator that shows the number of obstacles.
- An array with distances of each object from the Kinect lenses and the coordinates of the corners of each object inside the image.
- An image with program results.
- A group of indicators showing information about position of objects: left, right and center.

The image of the icon is shown in Figure 3.

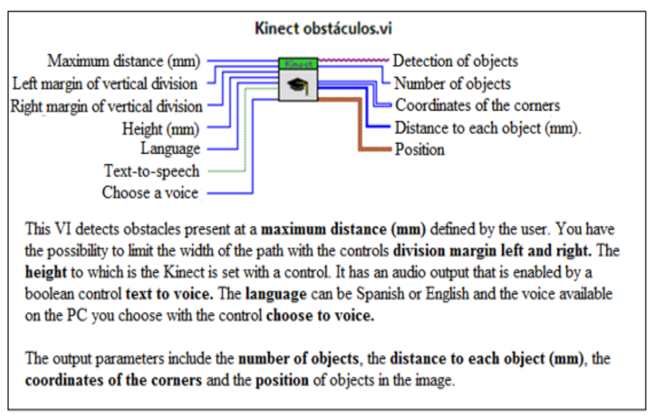


Figure 3: LabVIEW Icon ready to be used

III. Results And Discussion

The entire process of detection was performed on the depth image and a RBG image was used for visualization.

The tests were conducted for object location considering the trajectory: center, left, right, center-left, center-right, and total obstacle. Table1 presents the response rate of the algorithm. The tests results are shown in Figure 4. The results are shown in Table1. They include object shapes as a) rectangular, b) circular, c) cylindrical, and d) others such as triangular and elliptical. Also the tests include the analysis of the size of objects as e) small and f) large objects. In the depth test g), objects are placed at different distance from the Kinect sensor.

For the material test we used h) leather, ceramic, plastic, cloth, cardboard, f) wood, b) synthetic, c) wax and metal, i) transparent and j) reflective objects.

Table 1
Response Rate Algorithm for Different Object Detection

Type	Objects	Objects detected	Percent effectiveness
Location	10	10	100%
Shape	10	10	100%
Size	10	9	90%
depth	10	10	100%
materials	10	10	100%
transparent	10	5	50%
reflective	10	7	70%

The implemented algorithm detected the objects and calculated their location with a 100% of effectiveness.

The shape of objects was not a limitation to detect the objects using the algorithm in this work. Object with holes also were detected.

The algorithm discarded very small objects to avoid confusions with particles or false objects that are the product of the Kinect acquisition. Materials tests that include leather, ceramics, plastic, fabric and cardboard were segmented satisfactorily.

The test result indicates that the transparent and reflective objects are detected partially, segmenting only part of them.

IV. CONCLUSION

The Kinect sensor provides reliable information under certain conditions of the environment situation and enough depth information to classify a particular event with fast computational analysis. LabVIEW simplifies the programing and let do an easy-to-use interface.

The algorithm that we developed can detect objects of materials such as plastic, leather, fabric, cardboard, porcelain, synthetic that are the most common. The algorithm can also detect object with shapes as circular, rectangular, cylindrical or irregular forms object. Small detected objects cover an approximate height of 10cm. The algorithm cannot detect completely transparent, reflective or joined object. In addition, a new icon was created in LabVIEW.

REFERENCES

[1] Uijlings, J. RR, 2013. et al. Selective Search for Object Recognition. International Journal of Computer Vision, 104(2):154-171.

[2] Oji, R. 2012. An Automatic Algorithm for Object Recognition and Detection Based on ASIFT keypoints. Preprint arXiv:1211.5829.

[3] Viola, P., Jones, M.. 2001. Rapid Object Detection using a Boosted Cascade of Simple Features. Proceedings of the 2001 IEEE Computer Society Conference on In Computer Vision and Pattern Recognition CVPR2001. IEEE, 1:I-511-I-518

[4] Papageorgiou, C., Poggio, T. 2000. A trainable system for object detection. International Journal of Computer Vision, 38(1):15-33.

[5] Iralde L. I. 2012. Desarrollo de Aplicaciones Con MicrosoftKinect. Pamplona.

[6] Rakprayoon P., Ruchanurucks M., Coundoul A. 2011, Kinect-based Obstacle Detection for Manipulator, SI International, 68-73,

[7] Xia, L. C., Chia-C. Aggarwal, Jake K. 2011. Human Detection using Depth Information by Kinect. 2011 IEEE Computer Society Conference onComputer Vision and Pattern Recognition Workshops (CVPRW).15-22.

[8] Riyad A. El-I., Jidong Huang, Michael Yeh. 2012. Study on the Use of Microsoft Kinect for Robotics Applications. IEEE/ION Position Location and Navigation Symposium (PLANS). 1280 - 1288

[9] Ekelmann J., Butka B. 2012. Kinect Controlled Electro-Mechanical Skeleton. Mechanical and Electrical Engineering Departments Embry-Riddle Aeronautical University, FL USA.

[10] Ganganath N. and Leung H. 2012. Mobile Robot Localization using Odometry and Kinect Sensor. Department of Electrical and Computer Engineering Schulich School of Engineering. 91-94.

[11] Biswas K. K., Basu, S. K. 2011. Gesture Recognition using Microsoft Kinect®, Conf. on Automation, Robotics and Application. 100-103.

[12] Abramov, A., Pauwels, K., Papon, J., Wörgötter, F., & Dellen, B. 2012. Depth-supported Real-Time Video Segmentation with the Kinect. In IEEE Workshop on Applications of Computer Vision 2012. 457-464

[13] Lange, B., Koenig, S., McConnell, E., Chang, C. Y., Juang, R., Suma, E., & Rizzo, A. 2012. Interactive Game-Based Rehabilitation using the Microsoft Kinect. In 2012 IEEE Virtual Reality Short Papers and Posters (VRW). 171-172

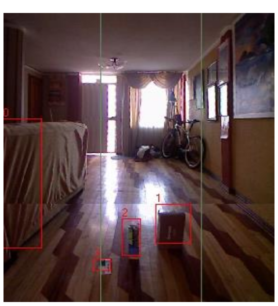
[14] Hui-mei Justina Hsu, 2011. The Potential of Kinect in Education. Int. Journal of Information and Education Technology, 365-370.

[15] Microsoft Kinect SDK. [Online] From: http://msdn.microsoft.com/en-us/library/hh973078.as px#Depth_Ranges. [Acessed on 2 Octubre 2015].

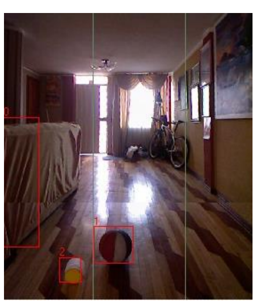
[16] Kinect SDK v1.0. [Online] From: https://decibel.ni.com/content/servlet/JiveServlet/download/2253-3-47391/Kinect%20SDK%20v1.0.zip. [Acessed on 10 July 2015].

[17] Lee, J.-S.. 1983. Digital Image Smoothing and the Sigma Filter. Computer Vision, Graphics, and Image Processing, 24(2), 255-269.

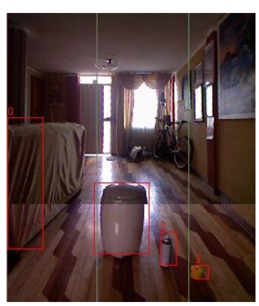
[18] Text to speech. [Online] From: https://decibel.ni.com/content/docs/DOC-2263. [Acessed on 10 December 2015].



(a) Rectangular cloth, cardboard



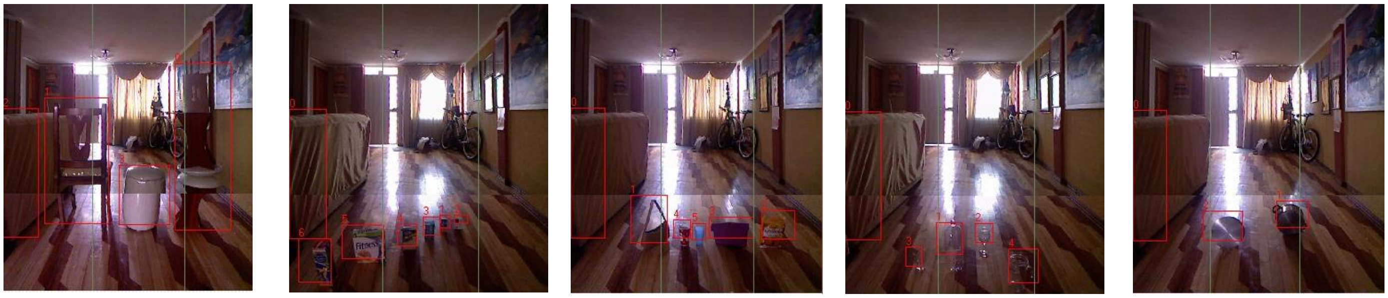
(b) Circular cloth, plastic, synthetic



(c) Cylindrical cloth, metal, wax

(d) Other forms cloth, plastic

(e) small cloth, cardboard



(f) Large wood, cloth

(g) Depth cardboard

(h) Materials leather, ceramics, plastic, clothes and cardboard

(i) Transparent

(j) Reflective

Figure 4: Phases of the playing card detection