Development of an Object Recognition and Location System Using the Microsoft Kinect™ Sensor*

Jose Figueroa, Luis Contreras, Abel Pacheco, and Jesus Savage

Biorobotics Laboratory, Department of Electrical Engineering,
Universidad Nacional Autonoma de Mexico, UNAM

Abstract. This paper presents the development of an object recognition and location system using the Microsoft Kinect™, an off-the-shelf sensor for videogames console Microsoft Xbox 360™which is formed by a color camera and depth sensor. This sensor is capable of capturing color images and depth information from a scene. This vision system uses *a*) data fusion of both color camera and depth sensor to segment objects by distance; *b*) scale-invariant features to characterize and recognize objects; and *c*) camera's internal parameters combined with depth information to locate objects relative to the camera point of view. The system will be used along with a robotic arm to grab objects.

Keywords: Keywords: Feature extraction, Scale Invariant Feature, Machine vision, Object detection, Pattern recognition.

1 Introduction

Autonomous mobile robots must have several capabilities to perform correctly their tasks: navigation, natural language understanding, object detection and manipulation. The ability for detecting and locating objects in the environment using vision is one the objectives of several research service robots groups. The purpose of our research is to develop a object recognition and location system which takes advantage of the Microsoft Kinect™sensor, which features both an image sensor and a depth sensor.

2 Depth Vision

The ability of locating objects in three-dimensional space is the most important ability which mobile service robot must have to interact with these objects in the environment. Most of the teams which get to Second Stage in Robocup@Home competition, as well as the teams in the top 5 in the final results have vision systems which combine depth perception and two-dimensional object recognition.

* This work was supported by PAPIIT-DGAPA UNAM under Grant IN-107609.

T. Röfer et al. (Eds.): RoboCup 2011, LNCS 7416, pp. 440–449, 2012.

2.1 Kinect

Hardware. The Microsoft Kinect^{TM} sensor for Microsoft Xbox 360^{TM} videogames console, is an input device which allows users to interact with videogames in a controller-less way through gesture and voice recognition. It was release in North America in November 4th 2010 (Fig. 1) with the retail prices of $150.00 USD, in USA, and $2,300.00 M.N., in Mexico.

The vision hardware of Kinect^{TM}is capable of making a three-dimensional reconstruction of the environment by using an infrared camera and an infrared projector, and capturing color images streams. It is based in the PrimeSensor^{TM} Reference Design de la compañía PrimeSense[8], whose vision subsystem is formed by an 640x480@30Hz RGB camera, an 640x480@30Hz infrared camera and an infrared projector.

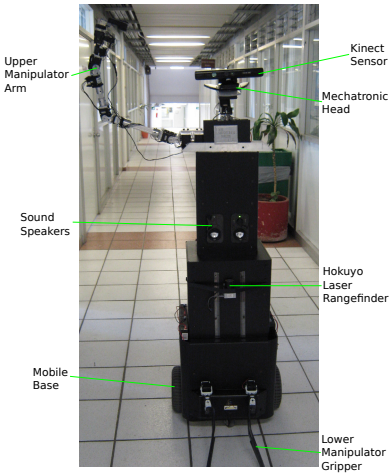

Fig. 1. Microsoft Kinect mounted on robot^{TM}

The PrimeSense's technology for depth information acquisition is based in the Light Coding^{TM}[8] technique, which works by codifying the scene's volume with near-infrared light. The depth vision system utilizes an off-the-shelf CMOS image sensor to read the coded light back from the scene. PrimeSense's SoC chip is connected to the CMOS image sensor, and executes a sophisticated parallel computational algorithm to decipher the received light and produce a depth image of the scene. The solution is immune to daylight.

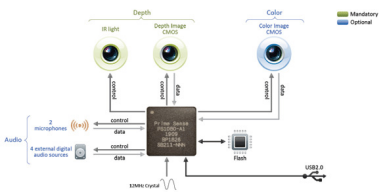

Fig. 2. PrimeSensor^{TM} Reference Design

Another capability of the PrimeSensor^{TM} Reference Design hardware is provided by a process called Registration which aligns pixel-by-pixel the elements of the depth map with the color image.

The limitations of Kinect technology are: its effective distance range is from 40 to 625 centimeters, reflective surfaces alter the readings, no depth data can be reconstructed if objects are shadowed from the infrared projector.

Software. In November 2010, Adafruit Industries offered a bounty to anyone who was able to create an open source driver for Kinect[1], which happened on November 10th, 2010 when a Spanish programmer

called Hector Martín was announced as the winner[2], who created a Linux driver which allowed using both RGB and depth camera.

The release of this driver and its corresponding forks for Windows and MacOS started a community for homebrew software which developed applications for user interfaces, robotics, artistic shows, amongst others. The popularity of Kinect in the developer community soared to such degree that PrimeSense released Kinect-enabled versions of their softwares OpenNI and NITE, allowing Kinect homebrew developers to make body tracking applications.

3 Vision System

Theory. The operations for recognizing and locating an object in a scene are represented by the first stages of the vision model of David Marr[7]. To begin with, the primal sketch stage gives the most important information about the image in two dimensions, such as edges, corners or blobs; in this stage, the SIFT (Scale-Invariant Feature Transform) algorithm[6] is used to characterize and recognize objects in a scene. Finally, the 2.5D sketch provides the depth information of the image; for this stage, the depth sensor of Microsoft Kinect™ is used to extract the depth information of an object in a scene.

Object Recognition. The objects which are going to be recognized by the vision system are represented by interest points computed by the SIFT algorithm. The training stage uses a procedure where images of an object are stored in a list. Once several images are captured, the next step consists in creating the pattern set, by getting the kd-tree[4] containing the SIFT features of each image stored in the list, this process is performed using a thread pool, which contains a loop which checks for any thread which has finished its processing and assigns a new image to the free thread. Each object is stored in the vision system as an structure which contains a set of kd-trees, a set of feature matchers and a name for the object and the set of these structures conforms the objects database.

The recognition process's preparations starts during the training stage, where a matcher is created for each SIFT feature's kd-tree. Then, when an object is going to to be recognized in a scene, a depth segmented RGB image is captured, its SIFT features are extracted they are searched for the nearest neighbors on each kd-tree of the pattern set, using the Best Bin First (BBF) algorithm[3]. The outliers in the nearest neighbor matched features are eliminated using the Random Sample Consensus algorithm (RANSAC)[5] with the homography transform as criteria for finding outliers; the matching process returns a positive result when matches are bigger or equal than four, because the homography matrix is calculated with at least four points.

In the case where an unknown object must be recognized, the recognition process is performed by matching unknown object's features against the pattern sets of each object stored in the objects database with BBF, and the greatest amount of matches is stored during the process. At the end of the matching process of each object database entry, the matches amount is stored in a histogram,

where each bin refers to an object stored in the objects database. Once matching process against all the entries of the object database is finished, the histogram is queried for the entry with the greatest amount, returning an index to an object database entry, which is used for performing an outlier deletion in the matched points using RANSAC. If the number of matches is greater than one, the set of matched points is used on the object location procedure.

Object Location. The set of matched points is used to make a bounding box in the scene image, where the recognized object is located, which is used to make a region of interest for the depth map. The object's location can be computed by using the centroid of the region of interest and get the depth value in that place or by finding the centroid of the three-dimensional point cloud generated by the depth information within the region of interest.

By taking advantage of PrimeSensor^{TM} Registration process, the only Kinect's camera which needs calibration is the RGB one, whose intrinsic parameters and distortion coefficients are obtained by using the Zhang's algorithm [9]. The color image and the pixel-aligned depth image are rectified using the RGB camera parameters and the coordinates of each three-dimensional point are computed with the following equations

$$P3D_x = (x_{RGBD} - cx_{RGB}) * \frac{f\,(depth\,(x_{RGBD}, y_{RGBD}))}{fx_{RGB}} \tag{1}$$

$$P3D_y = (y_{RGBD} - cy_{RGB}) * \frac{f\,(depth\,(x_{RGBD}, y_{RGBD}))}{fy_{RGB}} \tag{2}$$

$$P3D_z = depth\,(x_{RGBD}, y_{RGBD}) \tag{3}$$

where RGBD represents a pixel which contains both color and depth information.

4 Tests

The test started by capturing pictures for training and testing sets, using the RGB camera. The training set consisted in the SIFT features of a set of eight objects (Fig. 3a), and the testing set was formed by two objects which were rather similar to some objects of the testing set (Fig. 3b). Two tests were performed with this data: a control test and a false positives test. Input data for feature extraction was the following: depth-segmented RGB colorspace pixels from a 640x480 image of an object set at 40cm from the camera (Kinect's minimum depth value).

Two different tests were applied, all of them with segmented input pictures, so the pattern just contains objects. The objects used in these tests were a disinfectant spray (lysol), a bottle of Bonafont brand water (water), a generic brand coffee jar (coffee), a Nescafe brand coffee jar (nescafe), a cup, a jar of automotive grease (grease), a bottle of chili sauce (sauce) and a milk carton (milk) for the training set(Fig. 3a), and, a bottle of E-Pura brand water (epura_water) and

(a) Training Objects

(b) False Positive Test Objects

Fig. 3. Test Setup

a Nescafe Classic brand coffee jar (nescafe_class)(Fig. 3b). All the tests were made with the camera in a fixed viewpoint, positioned in the lower section of a domestic service robot which was used by UNAM Robocup@Home team for competition.

The first test consisted in using the pictures of the object training set to check if they can be recognized with a theoretical accuracy of 100%. Training data set was generated from the SIFT features of sixteen (16) images of each object captured at the same distance from the camera and rotated along the vertical axis to get images from all the sides of the object, and testing data set consisted of the training data set.

The second test consisted in using the pictures of the object testing set to check if false positives could be found. Training data set from former test was

used, and testing data set consisted of the SIFT features of eight (8) pictures of each object the testing data set.

5 Results

In both tests, five (5) runs were done because the randomization part of the RANSAC algorithm introduces variations in running time (Tab. 1, 8). Confusion matrices containing the amount of matches between objects, regardless if they are the same or not, were used to get the false positive counts compared to the amount of true positives. The lowest the false positives rate is, the highest is the uniqueness of the features in a data set (Tab. 3, 4, 5, 6, 7, 10, 11, 12, 13, 14).

For the control test, because the position of the camera and objects are the same during training and testing, it was expected that all the descriptors would have 100% of accuracy, and the results proved this hypothesis true (Tab. 2).

For the false positives test, one of the pictures of the E-Pura brand water bottle got a false positive with the generic brand coffee jar, and the Nescafe Classic coffee jar did not get any false positives (Tab. 9).

Table 1. Control Test: Average Running Time in Milliseconds

object	Test 1	Test 2	Test 3	Test 4	Test 5
coffee	2482.38	2598.44	2520.44	2480.44	2456
sauce	1505.44	1552.19	1573.62	1506.38	1593.19
nescafe	2051.38	2127.5	2301	2056.31	2120.62
milk	3764.5	3696.25	3752.81	3703.06	3608.5
water	1874	1857.38	1705.31	1821.31	1833
lysol	3924.38	4166.19	3857.12	4058.94	3775.25
cup	2401.44	2609.12	2300	2321.5	2363.38
grease	2014.38	2002.69	2002.69	2002.62	2048.5

Table 2. Control Test: Accuracy

object	Test 1	Test 2	Test 3	Test 4	Test 5
coffee	16/16	16/16	16/16	16/16	16/16
sauce	16/16	16/16	16/16	16/16	16/16
nescafe	16/16	16/16	16/16	16/16	16/16
milk	16/16	16/16	16/16	16/16	16/16
water	16/16	16/16	16/16	16/16	16/16
lysol	16/16	16/16	16/16	16/16	16/16
cup	16/16	16/16	16/16	16/16	16/16
grease	16/16	16/16	16/16	16/16	16/16

Table 3. Control Test 1 of 5: Confusion Matrix

object	coffee	sauce	nescafe	milk	water	lysol	cup	grease	none
coffee	16	0	0	0	0	0	0	0	0
sauce	0	16	0	0	0	0	0	0	0
nescafe	0	0	16	0	0	0	0	0	0
milk	0	0	0	16	0	0	0	0	0
water	0	0	0	0	16	0	0	0	0
lysol	0	0	0	0	0	16	0	0	0
cup	0	0	0	0	0	0	16	0	0
grease	0	0	0	0	0	0	0	16	0

Table 4. Control Test 2 of 5: Confusion Matrix

object	coffee	sauce	nescafe	milk	water	lysol	cup	grease	none
coffee	16	0	0	0	0	0	0	0	0
sauce	0	16	0	0	0	0	0	0	0
nescafe	0	0	16	0	0	0	0	0	0
milk	0	0	0	16	0	0	0	0	0
water	0	0	0	0	16	0	0	0	0
lysol	0	0	0	0	0	16	0	0	0
cup	0	0	0	0	0	0	16	0	0
grease	0	0	0	0	0	0	0	16	0

Table 5. Control Test 3 of 5: Confusion Matrix

object	coffee	sauce	nescafe	milk	water	lysol	cup	grease	none
coffee	16	0	0	0	0	0	0	0	0
sauce	0	16	0	0	0	0	0	0	0
nescafe	0	0	16	0	0	0	0	0	0
milk	0	0	0	16	0	0	0	0	0
water	0	0	0	0	16	0	0	0	0
lysol	0	0	0	0	0	16	0	0	0
cup	0	0	0	0	0	0	16	0	0
grease	0	0	0	0	0	0	0	16	0

6 Conclusions

The Microsoft Kinect^{TM} Sensor, combined with accurate object recognition algorithms, can be turned into an affordable solution for object recognition and three-dimensional location, and encourage the development of advanced robotics to people or research teams who could not afford other depth vision technologies.

Table 6. Control Test 4 of 5: Confusion Matrix

object	coffee	sauce	nescafe	milk	water	lysol	cup	grease	none
coffee	16	0	0	0	0	0	0	0	0
sauce	0	16	0	0	0	0	0	0	0
nescafe	0	0	16	0	0	0	0	0	0
milk	0	0	0	16	0	0	0	0	0
water	0	0	0	0	16	0	0	0	0
lysol	0	0	0	0	0	16	0	0	0
cup	0	0	0	0	0	0	16	0	0
grease	0	0	0	0	0	0	0	16	0

Table 7. Control Test 5 of 5: Confusion Matrix

object	coffee	sauce	nescafe	milk	water	lysol	cup	grease	none
coffee	16	0	0	0	0	0	0	0	0
sauce	0	16	0	0	0	0	0	0	0
nescafe	0	0	16	0	0	0	0	0	0
milk	0	0	0	16	0	0	0	0	0
water	0	0	0	0	16	0	0	0	0
lysol	0	0	0	0	0	16	0	0	0
cup	0	0	0	0	0	0	16	0	0
grease	0	0	0	0	0	0	0	16	0

Table 8. False Positives Test: Average Running Time in Milliseconds

object	Test 1	Test 2	Test 3	Test 4	Test 5
nescafe_class	2064.88	1989	1998.62	1996.75	1975.25
epura_water	2361.38	2464.88	2379	2353.75	2380.88

Table 9. False Positives Test: Accuracy

object	Test 1	Test 2	Test 3	Test 4	Test 5
nescafe_class	0/8	0/8	0/8	0/8	0/8
epura_water	0/8	0/8	0/8	0/8	0/8

Table 10. False Positives Test 1 of 5: Confusion Matrix

object	coffee	sauce	nescafe	milk	water	lysol	cup	grease	none
nescafe_class	0	0	0	0	0	0	0	0	8
epura_water	1	0	0	0	0	0	0	0	7

Table 11. False Positives Test 2 of 5: Confusion Matrix

object	coffee	sauce	nescafe	milk	water	lysol	cup	grease	none
nescafe_class	0	0	0	0	0	0	0	0	8
epura_water	1	0	0	0	0	0	0	0	7

Table 12. False Positives Test 3 of 5: Confusion Matrix

object	coffee	sauce	nescafe	milk	water	lysol	cup	grease	none
nescafe_class	0	0	0	0	0	0	0	0	8
epura_water	1	0	0	0	0	0	0	0	7

Table 13. False Positives Test 4 of 5: Confusion Matrix

object	coffee	sauce	nescafe	milk	water	lysol	cup	grease	none
nescafe_class	0	0	0	0	0	0	0	0	8
epura_water	1	0	0	0	0	0	0	0	7

Table 14. False Positives Test 5 of 5: Confusion Matrix

object	coffee	sauce	nescafe	milk	water	lysol	cup	grease	none
nescafe_class	0	0	0	0	0	0	0	0	8
epura_water	1	0	0	0	0	0	0	0	7

References

1. Adafruit: The Open Kinect project - the ok prize - get $3,000 bounty for Kinect for Xbox 360 open source drivers (November 2010), http://www.adafruit.com/blog/2010/11/04/the-open-kinect-project-the-ok-prize-get-1000-bounty-for-kinect-for-xbox-360-open-source-drivers/
2. Adafruit: We have a winner - Open Kinect driver(s) released - winner will use $3k for more hacking - plus an additional $2k goes to the eff! (November 2010), http://www.adafruit.com/blog/2010/11/10/we-have-a-winner-open-kinect-drivers-released-winner-will-use-3k-for-more-hacking-plus-an-additional-2k-goes-to-the-eff/
3. Beis, J.S., Lowe, D.G.: Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In: Proc. IEEE Conf. Comp. Vision Patt. Recog., pp. 1000–1006 (1997)
4. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Commun. ACM 18, 509–517 (1975), http://doi.acm.org/10.1145/361002.361007

5. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: Readings in Computer Vision: Issues, Problems, Principles, and Paradigms, pp. 726–740. Morgan Kaufmann Publishers Inc., San Francisco (1987), http://portal.acm.org/citation.cfm?id=33517.33575
6. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)
7. Marr, D.: Vision: a computational investigation into the human representation and processing of visual information / David Marr. W.H. Freeman, San Francisco (1982)
8. PrimeSense: Primesense, reference design, http://www.primesense.com/?p=514
9. Zhang, R., Tsi, P.S., Cryer, J.E., Shah, M.: Flexible camera calibration by viewing a plane from unknown orientations. In: Proceedings of the 7th International Conference on Computer Vision, pp. 666–673 (September 1999)