International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 09 | Sep-2016 www.irjet.net p-ISSN: 2395-0072

Object Segregation by an Autonomous Robot using Microsoft Kinect

Shantanu Ingle[1], Madhuri Phute[2]

[1]*Department of E&TC, Pune Institute of Computer Technology, Pune, Maharashtra, India*

[2]*Department of E&TC, Pune Institute of Computer Technology, Pune, Maharashtra, India*

Abstract - *Detecting and recognizing objects are fundamental components of robotic navigation. In order to effectively interact with its environment, a robot must be able to classify and distinguish between multiple objects, of same or distinct colors, and the surrounding environment. It should eventually identify the position as well as structure of these objects, and update the status in real-time. We propose an idea that uses Microsoft Kinect along with a 3 wheel drive autonomous robot to achieve these goals. The algorithm that was developed for this project was able to accurately segment objects of different shapes and colors placed at different distances from the Kinect and place the objects using a robot to its desired location. In this project, a self driving autonomous robot, with a gripper attached to it and the Kinect mounted on top, is used to detect, recognize, pick up and place objects at their predefined locations. Using the RGB-D information obtained from Kinect, we are able to identify objects of a variety of colors and shapes placed at different distances from the robot.*

Key Words: **Autonomous Robot, Kinect, Object Recognition, MFCC, Neural Network, Bluetooth, Ultrasonic Sensor.**

1.INTRODUCTION

Ever since Microsoft launched the Kinect Xbox 360 [1], back in 2010, researchers and hobbyists have widely experimented with it to exploit its potential for applications other than gaming. An intriguing aspect of the Kinect is that it provides distance data, i.e. a depth component, along with RGB data. This allows a 3-Dimensional interpretation of the captured 2-Dimensional image.

Taking that aspect into account, this project aims at developing an autonomous robot, capable of recognizing voice commands and identifying objects based on them.

The goal behind our idea is to develop a general framework for classifying objects based on RGB-D data from Kinect. Using this framework, a robot equipped with Kinect will take the name of an object as an input from an inbuilt microphone array of the Kinect sensor, scan its surroundings, and move to the most likely matching object that it finds. As a proof of our concept, we demonstrate our algorithm in an office/school environment.

The scope of this project currently spans 3 basic objects, namely 'cube', 'box' and 'sphere' of 3 basic colors; 'red', 'green' and 'blue'.

Being in its inchoate stage, the objects for classification are limited, but further detailed design would lead to applications spanning a variety of fields. One such example of its use is in home automation as a voice enabled personal assistant. Designed such that the physically disabled people or the aged, who are incapable of moving around on their own volition, can instruct the robot to fetch certain objects or carry out certain tasks. Military applications may also find use of this project, such as deployment of a dummy robot in a war field to identify any ammunition planted and eventually diffuse it, or maneuver around mapping the area and ultimately gauging threats. Medical assistants such as a real-time patient's body monitoring nurse are also in consideration, which will keep track of the patient's condition and send information about the patient's health to the doctor in real time, and also administer any emergency remedies if required. Use of such robots at a warehouse for its management and functioning is another feasible future for this project. Logging of consignments, retrieving packages from their locations and sorting of packages based on a variety of parameters can easily be achieved. Not only will it be efficient, but also highly cost effective.

2. RELATED WORK

Object recognition and segregation is a widely researched topic in the field of cognitive robotics. The detection of objects and maneuvering of intelligent autonomous robots is the motive behind development of a plethora of techniques, each with its own merits and limitations along with its intended area of application.

Most techniques for object recognition today rely on a set of descriptors. In essence, a picture is taken from a camera and individual entities within the picture are determined by computing certain values of descriptors. The relations between them, or the 'distance' between these descriptor vectors is what determines the object[2]. A major focus of this project is to combine the depth data along with the RGB data from the Kinect to simulate a 3-Dimensional environment and identify not only the presence of an object but also its location in terms of lateral distance from the robot. Kinect is now being used for applications far beyond

that of gaming, for which it was initially designed. Boston dynamics, acquired by Google in 2014, has done research and development in robotics enabled with deep neural nets[3], but has mostly developed robots for the United States Department of Defense. One of the early products is Atlas, which is a high mobility, humanoid robot designed to negotiate outdoor, rough terrain. Six wheeled robots developed by Starship Technologies delivers meals[4]. When they are deployed, the 4mph robots will operate as a "last-mile" solution, delivering food to customers within a 2-3 mile radius with help from its on-board GPS system and various sensors. When a robot arrives at its destination, customers simply need to type in a code that has been sent to them via the mobile app to open the lid and collect their food.

3. PROBLEM STATEMENT

The project is primarily aimed at distinguishing objects based on their color and shape and segregating them using a 3 wheel drive autonomous robot. Kinect, a Gripper arm, the HC-05 Bluetooth Module, a Li-Po Battery, all the circuitry (Motor Drivers, Arduino Mega) and a Laptop, for processing images acquired from Kinect, will be mounted on the robot. The user will provide a voice command of the object to be found. Once the voice input is analyzed and the object to be found is known, the Kinect will capture a frame using both, its RGB and IR sensors, and MATLAB will process them. It will identify the color, shape and distance of the objects in the frame and check whether the desired object is present or not. If the object is found, MATLAB will transmit the location of the object to the HC-05 Bluetooth module which is in turn connected to Arduino Mega. Each object will have a fixed position, which will be pre-programmed in the Arduino Mega. Mega will then maneuver the robot towards the object and pick up the object with the gripper arm attached to it, move to the predefined location and place the object. The robot will then return to its original position.

4. ALGORITHM & IMPLEMENTATION

The flow of the project would be to first identify the object from the voice input, then analyze the image from the Kinect to check whether the said object is present, and if so then at what distance. The algorithm for its execution is as follows:

1. Speech Analysis
    A. Word Detection and Splitting
    B. Coefficient Extraction
    C. Word Recognition
2. Object Detection
    A. Color Mask
    B. Depth Mask
    C. Superposition
3. Object Identification
    A. Bounding Box Technique
    B. Identify the Location
    C. Calculate the Distance
4. Transmitting Information via Bluetooth
5. Robot Actuation
    A. Check for Obstacles
    B. Actuate the Robot

5. SPEECH ANALYSIS

The speech is recorded in MATLAB with 8 kHz sampling frequency[5]. It is processed to identify the object mentioned. The procedure involves 3 major steps:

5.1 Word Separation

The recorded speech signal is split into two separate words. This is done by detecting a region of silence in between two words and using it to split the audio sample.

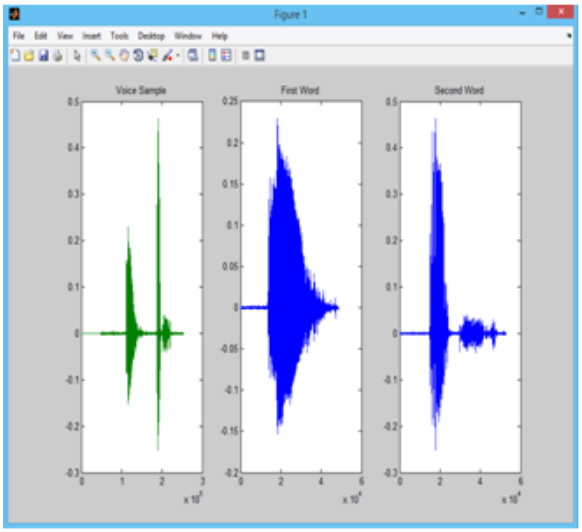

Figure 1: Word Separation in MATLAB

5.2 Extracting coefficients -

The Mel-Frequency Cepstrum (MFC) is a representation of the short- term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear Mel scale of frequency[6].

Mel-Frequency Cepstral Coefficients (MFCCs) are coefficients that collectively make up an MFC. In an MFC, the frequency bands are equally spaced on the Mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands. This frequency warping allows for better representation of sound.

The Mel-Frequency Cepstral Coefficients are calculated by the following formula[7]:

$$Mel(f) = 2595 * \log_{10}\left(1 + \frac{f}{700}\right)$$

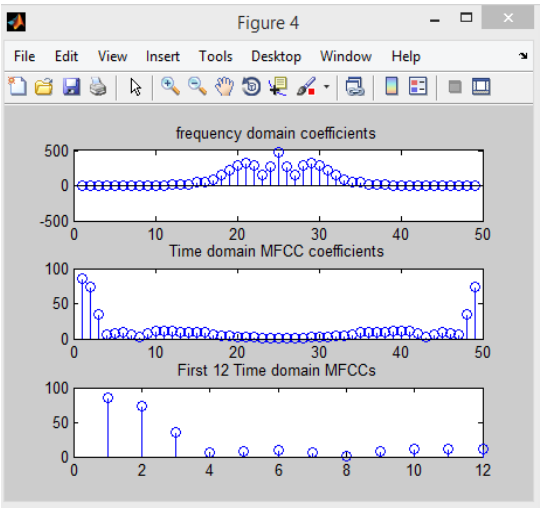

Figure 2: MFCC Extraction

5.3 Testing through the neural networks-

Once extracted from the recorded audio samples, the MFCCs are compared to those extracted in ideal conditions using a neural network[8].

The coefficients are passed through a neural network which is trained to identify words spoken by a particular person. Once the object is identified from the voice sample, MATLAB commences analysis of the image.

6. OBJECT DETECTION AND RECOGNITION

6.1 Object Detection

Object detection is done using an algorithm which combines the feeds from the RGB and depth sensors to apply masks which determine whether the given object is present or not.

Color Mask

Change of Color Space

The HSV (Hue, Saturation & Value) color space separates the color component of a pixel from its intensity, thus making it easier to detect the color of an object in conditions of varying light. In comparison to other color spaces capable of doing the same (e.g. YIQ, YCbCr), RGB to HSV conversion is simpler and thus it is used here.

One way of visualizing the HSV color space is as an inverted conical structure as shown below[9].

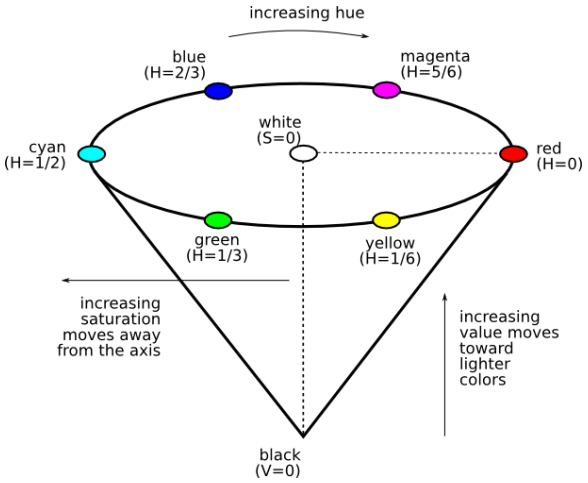

Figure 3: RGB to HSV Conversion

The 'Hue' represents the general color of the pixel, i.e. its position on the color wheel. The 'Saturation' shows how close the pixel is to grayscale, while 'Value' represents the intensity or brightness of the pixel.

Conversion from RGB to HSV color space is done as follows:

First, scale the RGB values from 0 to 255 range to a 0 to 1 range:

R' = R/255; G' = G/255; B' = B/255

Then, calculate the maximum and minimum of the 3 components:

M = max(R',G',B'); m = min(R',G',B')

The value of Delta or Chroma is defined as the difference between M and m:

Δ = M - m

The calculations for Hue, Saturation and Value are done as:

$$H = \begin{cases} 0°, & \Delta = 0 \\ 60° * \left(\frac{G' - B'}{\Delta} \bmod 6\right), & M = R' \\ 60° * \left(\frac{B' - R'}{\Delta} + 2\right), & M = G' \\ 60° * \left(\frac{R' - G'}{\Delta} + 4\right), & M = B' \end{cases}$$

$$S = \begin{cases} 0, & M = 0 \\ \frac{\Delta}{M}, & M \neq 0 \end{cases}$$

$$V = M$$

Determination of Color Mask

The Color Mask places constraints on the Hue, Saturation and Value components of the pixels, thus filtering them on the basis of their color. Another added advantage of the HSV color space is that the basic color identification can be done by analyzing a single (Hue) frame, as opposed to 3 (R, G, B) frames in the RGB color space. Although the Hue component shows only slight change with the variation in ambient light, the Saturation and Value components vary significantly and thus they have to be adjusted according to the light conditions.

The color mask essentially places '1's in the output image for all the pixels which satisfy certain conditions placed on the Hue, Saturation and Value components and places '0's for the remaining pixels, thereby producing a binary image.

Depth Mask

The depth mask is a filter which includes all the pixels having the depth attribute to be within a predefined range, or above or below a predefined value. It is applied to the frame captured from the IR sensor. The output contains all those pixels which are at the desired distance or within the desired range. Similar to the color mask output, the depth mask output is also a binary image.

Superposition

The term 'Superposition' implies performing logical AND operation of the color and depth mask output images. The resultant image displays all the pixels of a desired color placed within a desired range of distance. The output frame contains white sections corresponding to the front view of the object[10].

6.2 Object Recognition

Once the output of the superposition is available, the next task is to determine the shape of the object in the frame. The technique used here is a rudimentary one, meant for only 3 basic shapes; a cube, cuboid and a sphere. The procedure to identify the shape is as follows:

Bounding Box Technique

MATLAB'14 provides an image property called 'Bounding Box' for binary images. This property draws a box around the '1's or any white sections in the image. These sections of pixels are called 'blobs'. Once all the blobs are identified and bounding boxes are drawn around them, the properties of that bounding box (area, side length, etc) are considered to identify the image.

For a sphere, the bounding box is a square, enclosing the circular front view of the bounding box. For a circle of radius 'r' enclosed by a square of side length '2r', the ratio of the area of the circle to the area of the bounding box is a constant, given as:

$$A = \frac{\pi r^2}{4r^2} = \frac{\pi}{4} = 0.7854$$

For a cubical object (square as its front view) or a cuboidal object (rectangle as its front view), the ratio of area of the blob to area of the bounding box will be 1. So the differentiation is done based on the side length of the bounding box; if two adjacent sides are of same length, then the object is a cube and if the adjacent side lengths differ considerably then the object is a cuboid.

Identify the Location

Once the object has been identified, it is necessary to place the object in the frame so that the robot can navigate towards its location. Since there are 3 objects placed for this demonstration, the frame has been divided into 3 sections namely : Left, Center and Right. The 640 columns of the total frame size of 640 x 480 has been divided into 3 columns spanning from [1, 210], [211,430] and [431,640]. The section in which the object lies is then found as being either the Left, Center of Right.

```
Enter object to find: 'Green Cube'
```

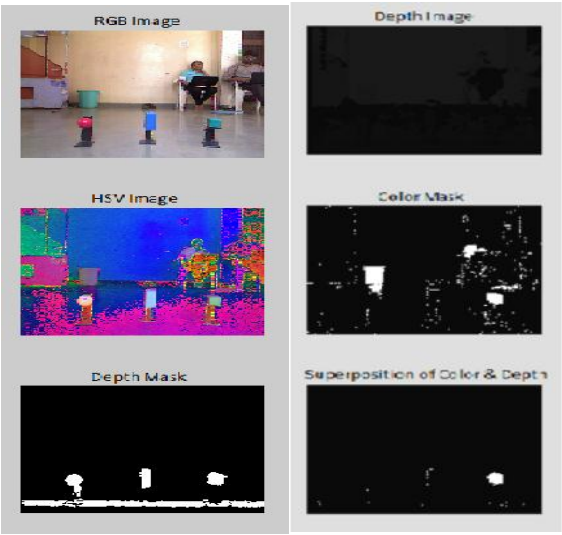

Figure 4: Image Processing for Green Cube

Calculate the Distance

The centroid of the blob corresponding to the desired object is calculated. The distance of the desired object is given simply as the depth attribute of the pixel corresponding to the centroid.

Figure 5: Identification of Cube

```
Cube Found!
Left

distObj =

    936

Continue? y=1/n=0 : 0
>>
```

Figure 6: Location of Cube and Distance

Note: Images from the Kinect are displayed in an inverted manner, i.e. flipped along the vertical axis. Thus, the cube which is seen on the right side in the frame is actually on the left in the demonstration. Thus while displaying the location and while commanding the robot to move towards the object, the opposite of what is determined from the frame is provided.

7. ROBOTIC ACTUATION

Information about the object and its location is transmitted to the ATmega microcontroller via Bluetooth. An HC-05 Bluetooth module is interfaced with Arduino Mega microcontroller board which receives following data about an object:

- The color of the object i.e. Red, Green or Blue; (R/G/B)
- The shape of the object i.e. Sphere, Cube or Box (cuboid); (S/C/B)
- The location of the object i.e. Left, Center of Right; (L/C/R)
- Distance of the object

The robot actuation process is as follows:

7.1 Check for Obstacles

Checking for obstacles is done using HC-SR04 ultrasonic transceivers. Before actuating the robot, the ultrasonic transceivers check for any obstacle within the distance of the desired object. If it senses an obstacle, the microcontroller will wait for 5 seconds before checking again. If it senses that the route is free of any obstacles, the motors will be actuated. A maximum of two attempts will be made in the case of obstacle detection. Then the process will be aborted and no actuation will occur.

7.2 Actuate the Robot

The routine corresponding to the location in the frame, i.e. Left, Center or Right, will be executed and the robot will pick up the object with the gripper and drop it at the predefined location.

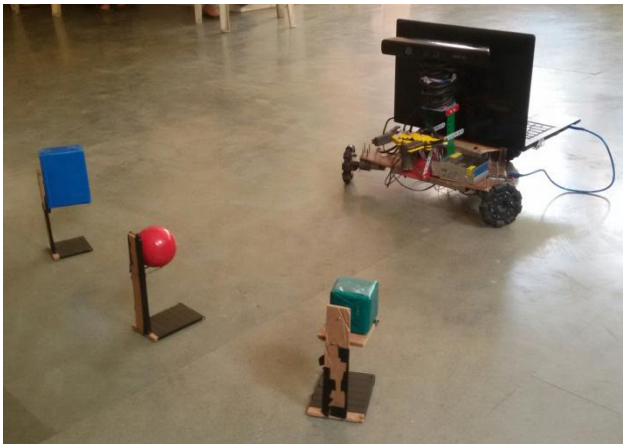

Figure 7: Final Assembly

8. EXPERIMENTAL RESULTS

8.1 Test Cases

This project was designed to recognize any of the 3 objects (sphere, cube, cuboid) in any of the 3 colors (red, green, blue), thus making a total of 9 identifiable objects.

During testing, it was found that all 9 objects are successfully identified and segregated into their respective positions by the robot.

Expansion of the code to include a fourth object, a cylinder, was also successful in specific ambient light conditions.

8.2 Power Requirement

A typical conundrum for any commercial product is of achieving maximum work output by consuming proportional, if not less, amount of power. This automated robot requires one 12V battery which can supply power to the motors, the gripper, the Arduino and the Kinect.

Component	Power Requirement
4 Johnson DC Motors (3 for the wheels of the Robot, 1 for the Gripper)	12V, 700mA each - Battery
Arduino Mega	5V - Laptop's USB Port
Kinect	12V, 2A - Battery; 5V -Laptop's USB Port,

9. CONCLUSION

In this project we have presented a novel and easy approach to object recognition, enabling a 3 wheel drive autonomous robot to segregate objects based on their color and shape. The 'Depth' attribute available from the Microsoft Kinect has been used to obtain a 3-Dimensional mapping of a 2-Dimensional frame, thus not only locating an object's position in the frame but also obtaining its lateral distance. Voice command recognition is done with Mel Frequency Cepstrum Coefficients. The top 12 coefficients for each of the 6 words are used as outputs for the neural net, which is trained to recognize those words in noisy and ideal environments. HSV color space is used to determine the colors and a depth range is set to identify all objects within it. The superposed object obtained is recognized using 'Bounding Box' properties like area, side length, and area ratio. Arduino Mega controls the robot - which sorts the objects - and maneuvers it to the dedicated locations.

The field of cognitive robotics is developing by leaps and bounds every day due to advancements in AI and optimization of hardware. Incorporating Self Learning Algorithms will allow the robot to interact with its surroundings and maneuver flawlessly by taking into cognizance all external factors.

REFERENCES

[1] Microsoft; http://www.xbox.com/en-us/kinect.

[2] Antonio Sgorbissa , Damiano Verda, "Structure-based object representation and classification in mobile robotics through a Microsoft Kinect", IEEE ,2013

[3] Boston Dynamics_ Dedicated to the Science and Art of How Things Move; http://www.bostondynamics.com/robot_Atlas.html#.

[4] Azwan Jamaluddin, "10 Creative And Innovative Uses Of Microsoft Kinect - Hongkiat"; http://www.hongkiat.com/blog/innovative-uses-kinect/.

[5] MathWorks; http://in.mathworks.com/help/

[6] Koustav Chakraborty, Asmita Talele, Prof. Savitha Upadhya, "Voice Recognition Using MFCC Algorithm", International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-2163, Volume 1 Issue 10 (November 2014)

[7] Dr. ShailaApte, "Speech and audio processing", Wiley India Publication

[8] LaureneFausett, "Fundamentals of Neural Networks: Architectures, Algorithms And Applications", Pearson Education, Inc, 2008.

[9] RGB to HSV conversion | color conversion - RapidTables.com: http://www.rapidtables.com/convert/color/rgb-to-hsv.htm

[10] Jos´e-Juan Hern´andez-L´opez, "Detecting objects using color and depth segmentation with Kinect sensor", Iberoamerican on Electronics Engineering and Computer Science- IEEE, 2012.