International Journal of Computer Science, Information Technology and Management
Vol. 2 No. 2 (December, 2016)

A comparison of Computer Vision Techniques for Indoor Robot Localization

Adam Mooers
Undergraduate student, Department of Electrical and Computer Engineering
James Davis
Undergraduate Student, Department of Electrical & Computer Engineering
University of West Florida
Pensacola, FL 32514, USA
Alexander Whiteside
Undergraduate Student, Department of Electrical & Computer Engineering
University of Florida
Gainsville, FL 3261 , USA

Chandra Prayaga
Professor, Department of Physics
University of West Florida
Pensacola, FL, 32514, USA
Lakshmi Prayaga
Associate Professor,
Department of Instructional, Workforce and Applied Technologies
University of West Florida
Pensacola, FL, 32514, USA

Abstract **-** This paper describes the comparison among three techniques for indoor robot localization, one based on a webcam, one on color-gradient pattern recognition, and one which uses the Microsoft Kinect system.

Index Terms **- Robotics, robot localization, computer vision, Kinect, colour sensor.**

I. Introduction

Robot localization is an important problem in several fields in robotics, including industrial, entertainment, and military applications. Several different techniques have been used to locate a robot in a plane, including optical, ultrasonic, GPS, camera-based techniques. I-Hsum Li et al [1] have recently used a webcam-based technique, and have also reviewed many of the techniques listed above. Sibai et al [2] have used wheel optical encoders, ultrasonic techniques, and WiFi signal strength to locate a robot. Borenstein et al [3] also give an exhaustive review of several methods.

II. Methodologies

Three localization techniques were implemented and compared via their resulting absolute locations, as well as control positions measured with a meter stick. The indoor location was consistent among tests, in a square area measuring 3 m by 3 m, on a matte floor with a printed gradient, which will be described later. One webcam was mounted on the ceiling providing a view of the entire arena. Another webcam was mounted on the robot itself, facing the ground. A Microsoft Kinect was mounted to provide depth data.

The first technique, Single-Camera Localization, illustrates one of the common localization techniques based on computer vision. In this scheme, robots are marked with a tag, which identifies them in the arena. The overhead camera uses blob tracking to determine the mean pixel coordinates of each tag. A perspective transform, unique to the camera's orientation relative to the floor plane, is applied to each coordinate. This process yields the blob's floor plane x and y coordinates. Since the width and length of the field are known, a conversion ratio was used to convert the transformed coordinates to cm. This method's accuracy can be improved with lens-aberration correction and higher-resolution camera sensors.

The second technique leveraged a specially-printed floor containing overlapping gradients in the RGB color plane. Printed materials have a high color resolution, leaving the sensor as the primary limitation to this method's accuracy. As shown in fig. 1, the floor gradient changes linearly along the axes in RGB space. No color combinations are repeated so that each location can be uniquely identified by its color. A robot cannot be permanently misguided by a single incorrect result.

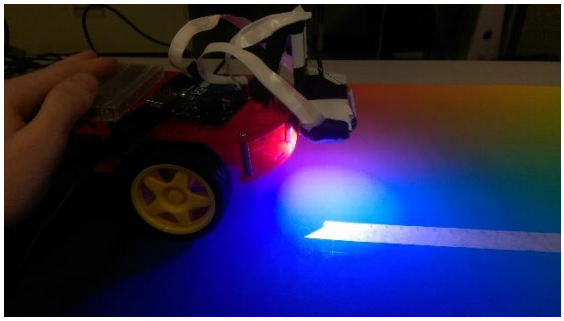


Fig. 1. Color gradient tracking using computer vision

The camera mounted on the robot was used for measuring the floor gradient's color. In order to compensate for the unpredictable effects of ambient lighting, a white LED Ring was installed around the camera. Automatic exposure and white balance correction features were disabled. This approach was implemented in Python with OpenCV. Tests were conducted on both a dedicated laptop, as well as an embedded Raspberry Pi SoC. An Arduino Uno provided real-time control of the robot's motors.

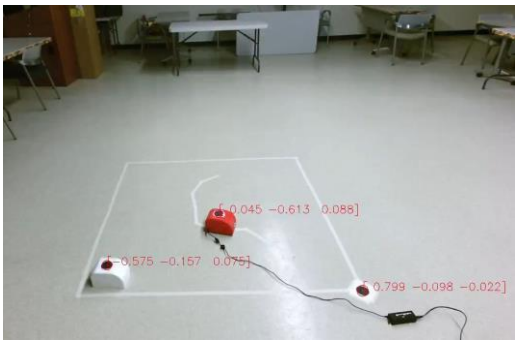


Fig. 2 Using Kinect for localization

The third technique involved a Microsoft Kinect v2, which contains both an infrared depth camera and a 1080P (1080x1920) color camera.

All Kinect data is represented by three coordinate spaces: the color space, the depth space, and the camera space. The color space is a 2D map for representing pixels from the color camera. Each coordinate contains a 24-bit BGR pixel value. The first coordinate corresponds to the upper-left pixel in the image. The last coordinate corresponds to the lower-right pixel. Internally, this space is represented as a row-major vector rather than the matrix typically associated with images. Pixel coordinates can converted to the color space as follows:

$$color\ space\ index = 1920 * y_{px} + x_{px}$$

Where, x_{px} and y_{px} are the pixel's coordinates and 1920 is the Kinect's horizontal resolution in pixels.

The Kinect's depth space describes raw depth data from the Kinect's infrared depth camera. This camera is located at the origin in figure 4. Each element in the space subtends a certain angle off the x and y axes shown in figure 4. The depth data does not represent spatial dimensions, but rather the distance from the image plane in millimeters. Much like the color space, the depth space is represented as row-major vector. There are a total of 222,600 16-bit values within the depth space, one for each pixel of the 424x525 depth image. Fig. 3 depicts the depth space.

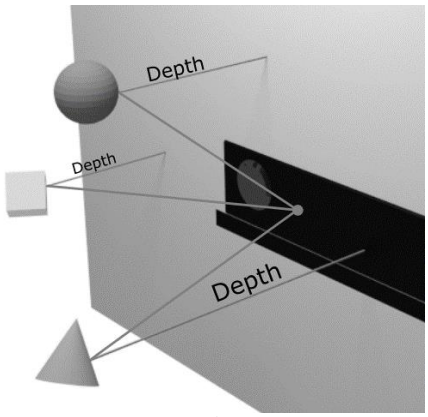


Fig. 3 Depiction of the Depth Space.

Given the x angle, y angle, and depth of a pixel it is possible to calculate its position in Cartesian space. This space is referred to as the camera space. Each point in the camera space consists of three coordinates, in meters, expressing the position of the point relative to the depth sensor. Fig. 4 shows this coordinate system.

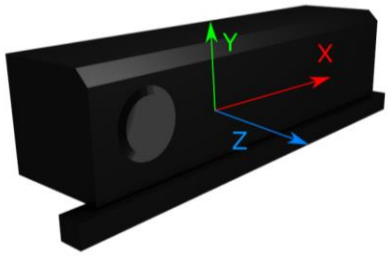


Fig. 4 Kinect Camera Space Diagram

To simplify localization, one more coordinate system is defined relative to the floor plane. The system, referred to here as the floor space, gives spatial coordinates relative to the floor plane. The floor space is equivalent to the color space with one shift and one rotation. The two spaces are identical when the when the Kinect's depth sensor is at the same level as the floor plane.

Transforming from the camera space to the floor space is described in the following derivation:

$$ax + by + cz + d = N \cdot [x, y, z] + d = 0$$

- x,y,z are camera space coordinates on the clipping plane
- d is the distance from the Kinect to the image plane/ground plane intersection line.
- N is the normalized normal of the ground plane (magnitude = 1)

$$P = [x_1, y_1, z_1]$$

- P is any point in the camera space to transform to the floor space

The first calculation is finding the minimum distance from the floor plane to P, that is, |PQ|:

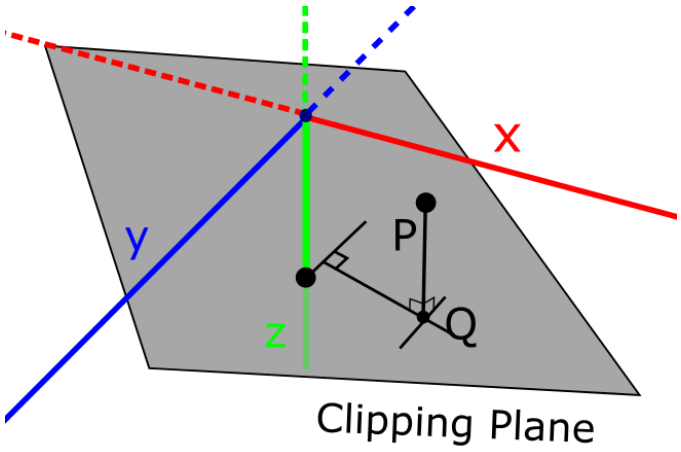


Fig. 3 Kinect data transformation

Because the minimum distance is along the normal (N), and |N|=1,

$$|PQ| = N \cdot P + d = \text{Height of the measured point}$$

Once |PQ| is known, finding Q is simple:

$$Q = P - |PQ| * N$$

At this point, the position on the plane is known but all the coordinates are still in camera space coordinates. The next step is to determine what the x and y axis are in camera space. For simplicity, the intersection between the camera plane and the floor plane was selected for x_{floor}.

$$X_{floor} = \left[1, 0, -\frac{a}{c}\right], \quad c \neq 0$$

$$x_{floor} = \frac{X_{floor}}{|X_{floor}|}$$

In the above, -a/c is the slope of floor plane along the camera space x-axis. Also c can never be zero, because the Kinect ignores all clipping planes above approx. 45 degrees because they look like walls.

In the above, -a/c is the slope of floor plane along the camera space x-axis. c is never zero because the Kinect ignores all clipping planes greater than. +-45 degrees off of the image plane.

Yfloor is the vector on the floor plane perpendicular to both the normal and xfloor.

$$y_{floor} = (x_{floor}) \times N$$

y_{floor} has a magnitude of one because both x_{floor} and N are normalized.

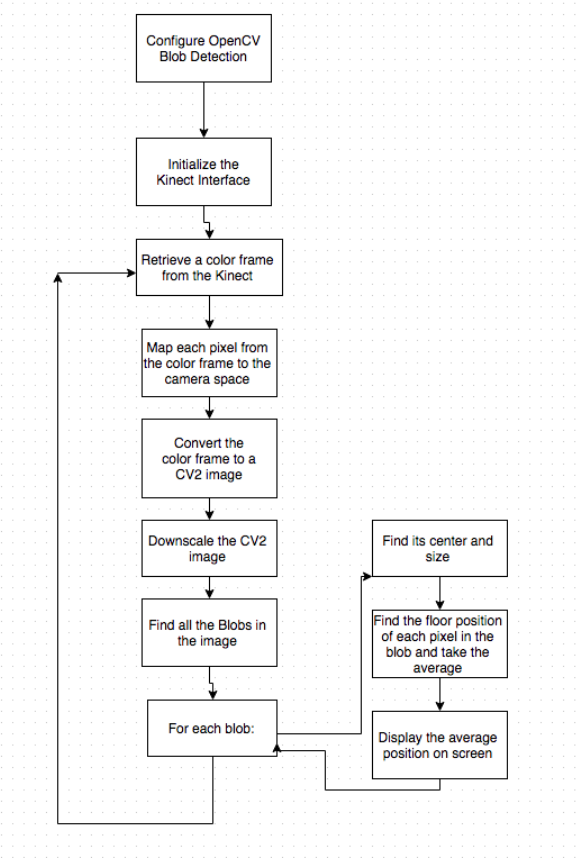
Two dot products are used to determine the x and y floor coordinates:

$$v \cdot u = \|v\| \cos(\theta) = \text{scalar projection of } v \text{ unto } u$$

This means the position of dot on the plane is given by:

$$\text{Position on ground plane} = [Q \cdot x_{floor}, Q \cdot y_{floor}, |PQ|]$$

The overall algorithm is shown in the flowchart below.



II. Results

In order to compare the results of each trial, we relied on the percent error from the of the reported location to the actual location of the robot under testing. In each test, the robot was moved in the X direction, the Y direction, and in a hybrid 2D movement, and absolute positions were obtained and compared with actual values.

The table below shows the average variation in location data for each localization system.

	Flooring	Percent Error
Overhead Camera	General Setup	4%
Robot Based Camera	Gradient A	14%
	Gradient B	
Kinect	NA	< 1%

Actual Position (+- 1mm)		Measured with Camera	
X, cm	Y, cm	X, cm	Y, cm
4.9	4.9	5.0	5.0
59.7	4.9	60.0	5.0
74.5	78.6	54.2	78.7
3.0	91.0	3.0	91.0

Fig. 4. Overhead camera data

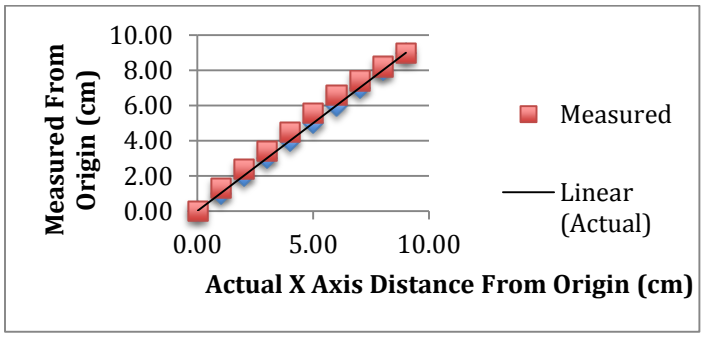


Fig. 5. Robot mounted camera data (x-coordinate)

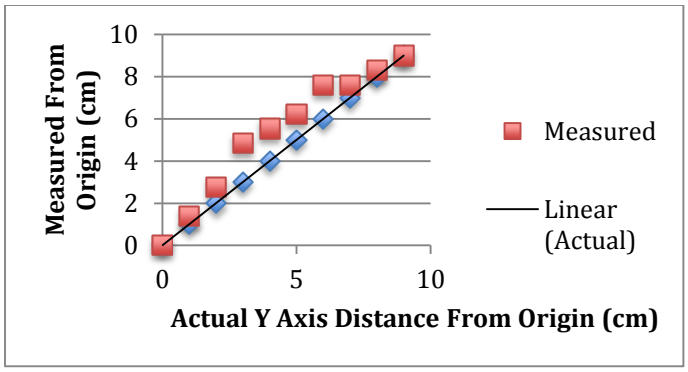


Fig. 6. Robot mounted camera data (y-coordinate)

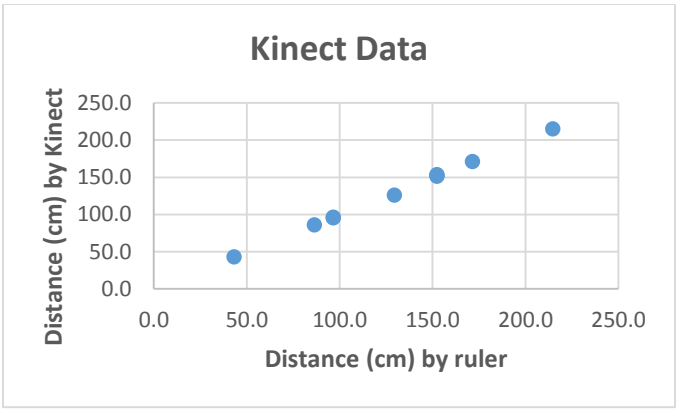


Fig. 7. Distance measurement using Kinect

III. CONCLUSIONS

The Kinect system gave the most accurate results. Both the overhead webcam technique and the computer vision based colour-gradient tracking technique had gave significant errors during testing.

While the Kinect was the most accurate, it was not the optimal solution by all metrics. The color sensing method had a marked advantage in simplicity and in situations where computing power is limited. The single webcam method also benefited from simplicity and low hardware cost.

Both had another drawback compared against the Kinect. Both required delicate calibration accounting for the size of the gradient paper and the exact placement of the ground plane sensors. These could represent a significant reliability problems in consumer, education, and other environments where maintenance is difficult to perform.

Because the Kinect is a 3D system, it is inherently unaffected by these issues. Additionally, by virtue of being a 3D sensor, the Kinect can also be used to track in the dimensions. This opens new possibilities for systems previously limited to two dimensions. Future work here involves using the Kinect system to track the motion of the robot.

IV REFERENCES

[1] I-Hsum Li, Ming-Chang Chen, Wei-Yen Wang, Shun-Feng Su, and To-Wen Lai, "Mobile Robot Self-Localization System Using Single Webcam Distance Measurement Technology in Indoor Environments", Sensors (Basel), 2014 Feb; 14(2): 2089–2109.

[2] Sibai, F.N., Trigui, H., Zanini, P.C. & Al-Odail, A.R. , "Evaluation of Indoor Robot Localization Techniques", International Conference on Computer Systems and Industrial Informatics (ICCSII), 2012, Sharjah, UAE.

[3] Borenstein, J., Everett, H. R., Feng, L., and Wehe, D., "Mobile Robot Positioning &
Sensors and Techniques", Journal of Robotic Systems, Special Issue on Mobile Robots. Vol. 14 No. 4, pp. 231 – 249.