Abstract

This experimental study involves using the Microsoft Kinect as a Spacecraft Sensor. The Kinect device has the ability to detect depth information along with RGB image data of any object within its operational range. The goal of this project was to develop relative position and orientation algorithms using the Kinect’s measurement data. Results include methods in which three dimensional data of specified objects in the Kinect’s operational field can be determined. A Matlab Graphical User Interface was also developed to exploit the uses of the Microsoft Kinect Data. A thorough description of the GUI and an example of its functions are included in this report.

Introduction

This project involves using a Microsoft Kinect, the “controller-free gaming and entertainment experience”. The Kinect is a 3D scanner which interprets 3D scene information from a continuously-projected infrared structured light. The Kinect for Xbox 360 has been a popular control free game since it was released on November 4, 2010. The goal of the project will be to develop spacecraft relative position and orientation (RPO) algorithms that utilize the Microsoft Kinect measurement data.

Project Significance

All essential control systems require two types of hardware components, sensors and actuators. Sensors are used to measure the state of the system, and actuators are used to alter the state of the system. Attitude determination uses a combination of sensors and mathematical models to determine vector components in the body and inertial reference frames. These components are used in several different algorithms to determine the attitude. A body’s attitude is typically described in the form of a quaternion, Euler angles, or a rotation matrix.\[1\]

The focus of spacecraft attitude dynamics and control is essential to all spacecraft operations. Depending on the mission or task in space, there is usually a small window for error. Errors in relative position and orientation can be very costly and sometimes extremely dangerous. This is why advancement in the area of spacecraft attitude determination and control is an important investment. There are many widespread uses for which attitude determination is an essential element. GPS, weather detection, communications, and satellite TV all depend on the efficiency of their corresponding satellite’s attitude and control system.

Attitude determination is not just used for spacecraft and satellites; the same knowledge can be used for aircraft. NASA’s aeronautics mission directorate focuses on developing tools and technologies to safely and efficiently improve aviation. Enhancing attitude determination technology supports the safety and efficiency of aviation as well as space operations. Particularly, unmanned aerial vehicles (UAVs) require efficient attitude determination control systems because they are often operated by a source far from the actual vehicle.

This project involves developing relative position and orientation algorithms.
using the Microsoft Kinect, which acts as a sensor. Attitude determination sensors are a major component of vehicle control systems. Progressing on this focus of attitude determination and control will support the advancement of NASA’s mission.

**Device Specifications**

The Kinect is based on software technology developed internally by Rare, a subsidiary of Microsoft Game Studios owned by Microsoft and range camera technology by Israeli developer PrimeSense. The software produced by these developers interprets 3D scene information from a continuously-projected infrared structured light\[2\], [3].

As described in Figure 1, the Kinect device itself uses a "RGB camera, depth sensor and multi-array microphone running proprietary software"\[4\]. An RGB camera is one that uses sensors to distinguish the levels of color intensity of Red, Green, and Blue. The depth sensor uses an infrared laser projector and a monochrome CMOS sensor \[5\]. A CMOS sensor, Complementary Metal Oxide Semiconductor, uses more than one type of transistor which allows it to require less power and make it better for portable devices. A sketch of a non-idealized system illustrates the actions of the depth sensor in Figure 2. The system relates to the way the human eyes are separated to determine depth.

The device’s technical specifications are as follows:

**Sensor**
- Color and depth-sensing lenses
- Tilt motor for sensor adjustment
- USB adapter to connect to the device to computer

**Field of View**
- Horizontal field of view: 57 degrees
- Vertical field of view: 43 degrees
- Physical tilt range: ± 27 degrees

- Depth sensor range: 1.2m – 3.5m

**Data Streams**
- 320x240 16-bit depth @ 30 frames/sec
- 640x480 32-bit color@ 30 frames/sec
- 16-bit audio @ 16 kHz

**Skeletal Tracking System**
- Tracks up to 6 people, including 2 active players
- Tracks 20 joints per active player

[6]

---

Figure 1: The Xbox Kinect, advertised by www.xbox.com. This figure shows the external features of the device.

Figure 2: Sketch of non-idealized depth sensor system
Methods and Procedure

Device Installment

To install the Kinect device to the computer, middleware and hardware binary files were downloaded from the OpenNI website. OpenNI, or Open Natural Interaction, is “an industry-led, not-for-profit organization formed to certify and promote the compatibility and interoperability of Natural Interaction (NI) devices, applications and middleware.” After the Kinect device was properly installed, the next step was to allow the Kinect data to be used in Matlab. Matlab executable (MEX) files, provided by the OpenNI organization, were downloaded to allow interaction between Matlab and the Kinect device. The MEX files allow a 640 x 480 display resolution of the RGB and depth image. Both sets of data will be used to describe relative position and orientation.

Methods

Using the depth data from the Kinect device, x-y-z coordinates in the inertial frame can be determined. Once the x-y-z coordinates of specific points are determined, relative position and orientation of objects can be computed. To start, the spherical coordinates are determined from a simple conversion of the Cartesian coordinates to give a full understanding of the three dimensional space. The conversion from Cartesian to Spherical coordinates is given in Equation 1, 2 and 3.

\[ r = \sqrt{x^2 + y^2 + z^2} \]  
\[ \theta = \tan^{-1}\left(\frac{y}{x}\right) \]  
\[ \phi = \cos^{-1}\left(\frac{z}{r}\right) \]  

The definition of the spherical coordinates is also illustrated in Figure 3.

\[ v_b = R_{bi}v_i \]  
\[ Body_{axis1} = R_{11}\hat{i} + R_{12}\hat{j} + R_{13}\hat{k} \]
\[ \text{Body}_{axis2} = R_{21i} + R_{22j} + R_{23k} \]  
\[ \text{Body}_{axis3} = R_{31i} + R_{32j} + R_{33k} \]

(7c) \hspace{1cm} (7d)

Once the rotation matrix had been established, all other forms of the attitude can be computed. Leonhard Euler rationalized that the rotation from one frame to another can be visualized as a sequence of three simple rotations about base vectors \[^1\]. The three rotations that he expressed are known as Euler angles. Commonly used with aircraft, the Euler angles are known as yaw, pitch, and roll. The three Euler angles are illustrated in Figure 4.

**Figure 4: Euler angles described on an aircraft**

To define the Euler angles, a sequence of rotations must be established. Roll, pitch and yaw are sometimes defined as a 3-2-1 sequence and sometimes defined as a 1-2-3 sequence. Depending on the sequence of rotations, the attitude may be described differently. Equations 8, 9, and 10 define the individual rotation matrices about each of the three axes.

\[
R_1(\theta) = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \theta & \sin \theta \\
0 & -\sin \theta & \cos \theta
\end{bmatrix}
\]

(8)

\[
R_2(\theta) = \begin{bmatrix}
\cos \theta & 0 & -\sin \theta \\
0 & 1 & 0 \\
\sin \theta & 0 & \cos \theta
\end{bmatrix}
\]

(9)

\[
R_3(\theta) = \begin{bmatrix}
\cos \theta & \sin \theta & 0 \\
-\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

(10)

A sequence of rotations is computed by multiplying the individual rotation matrices together. The 3-2-1 rotation matrix is shown in Equation 11.

\[
R_{3-2-1}^{\text{1-2-3}} = \begin{bmatrix}
c_{\theta_1}c_{\theta_2} & s_{\theta_1}c_{\theta_2} & -s_{\theta_2} \\
-c_{\theta_1}s_{\theta_2}s_{\theta_3} + c_{\theta_3} - c_{\theta_1}s_{\theta_2}s_{\theta_3} + c_{\theta_3} & c_{\theta_1}s_{\theta_2}s_{\theta_3} + s_{\theta_1}s_{\theta_2}s_{\theta_3} & s_{\theta_1}s_{\theta_2}s_{\theta_3} - c_{\theta_1}s_{\theta_2}s_{\theta_3} & c_{\theta_2}s_{\theta_3} \end{bmatrix}
\]

(11)

After a rotation matrix has been determined, the Euler angles can be extracted by manipulating the matrix cells. Equations 12, 13, and 14 show how to calculate the Euler angles from a given rotation matrix in the 3-2-1 sequence.

\[
\theta_1 = \tan^{-1} \left( \frac{R(1,2)}{R(1,1)} \right)
\]

(12)

\[
\theta_2 = \sin^{-1} \left( -\frac{R(1,3)}{R(2,3)} \right)
\]

(13)

\[
\theta_3 = \tan^{-1} \left( \frac{R(2,3)}{R(3,3)} \right)
\]

(14)

Equation 15 shows the rotation matrix of a 1-2-3 sequence. The difference between Equation 11 and Equation 15 shows how the rotation matrix is dependent on the rotation sequence.

\[
R_{1-2-3}^{\text{1-2-3}} = \begin{bmatrix}
c_{\theta_2}c_{\theta_3} & s_{\theta_2}c_{\theta_3} & -c_{\theta_3} \\
-c_{\theta_3}s_{\theta_2}c_{\theta_3} + c_{\theta_3} & c_{\theta_3}s_{\theta_2}c_{\theta_3} + s_{\theta_3}s_{\theta_2} & s_{\theta_3}s_{\theta_2}c_{\theta_3} - c_{\theta_3}s_{\theta_2} \\
\end{bmatrix}
\]

(15)

Another way to describe a body’s attitude is with the Euler axis and principal Euler angle. Euler’s theorem states that “The most general motion of a rigid body with a fixed point is a rotation about a fixed axis” \[^1\]. The axis is call the Eigen axis or Euler axis, and the rotation is called the principle Euler angle. The Euler angle (\(\Phi\)) and Euler axis (\(a\)) can be computed using the rotation matrix, as described in Equation 16 and 17.

\[
\Phi = \cos^{-1} \left[ \frac{1}{2} \text{trace}(R) - 1 \right]
\]

(16)

\[
a = \frac{1}{2 \sin \Phi} (R^T - R) = \begin{bmatrix}
0 & -a_3 & a_2 \\
-a_3 & 0 & -a_1 \\
-a_2 & a_1 & 0
\end{bmatrix}
\]

(17)
Lastly, the quaternion can also be determined from a rotation matrix. The quaternion, also known as an Euler parameter set, is a four-parameter set with some advantages over the Euler axis/angle set \[1\]. The first three parameters can be calculated using Equation 18 and the fourth parameter using Equation 19. Together, they make the quaternion vector as shown in Equation 20.

\[
q = a \cdot \sin \frac{\phi}{2} \quad (18)
\]
\[
q4 = \cos \frac{\phi}{2} \quad (19)
\]
\[
q = [q; q^4] \quad (20)
\]

Using the quaternion to describe a body’s attitude is advantageous when compared to the other methods because it does not carry any singularities.

Results

The finalized results of this project are described through a Matlab Graphical User Interface (GUI). The GUI allows one to stream the RGB and depth data in real-time and capture a specific frame. The RGB image displayed in the GUI is shown in Figure 5. The depth image displayed in the GUI is show in Figure 6. After the steamed data is paused, any point in the frame can be described in three dimensions. The data cursor allows the user to pick points from the 680 x 460 display. A data table provides a description of each point selected. The table includes the x and y coordinates in pixels, the x y and z coordinates in the inertial frame, and the spherical coordinates: r \( \Phi \) and \( \theta \). A drop down box allows the option to change the distance units from the default (millimeters) to centimeters, meters, inches, or feet. A 3D plot is also provided next to the RGB/depth image display to illustrate the selected points.

To use the attitude determination options of the GUI, a body frame must be selected. Using the data cursor, one can pick three points; one at the origin and one on two of the body frame axes. After the three points are selected and the two reference points are distinguished, the attitude can be computed. The attitude compute function calculates the two axes, normalizes the axes, and then computes the three attitude determination methods that were previously discussed. The attitude determination section of the GUI displays the rotation matrix, the Euler angles for a 3-2-1 and 1-2-3 sequence, the Euler axis and principle Euler angle, and the quaternion.
Figure 7: Example of describing the relative position and orientation of the shelf using the Kinect GUI

Figure 7 shows an example of how the GUI works. First, the RGB frame is paused. Then, the three points on the shelf are selected. Clicking *GetStats* computes all of the three dimensional data corresponding to the selected points. To determine the attitude, or relative orientation, of the shelf, the body frame is first established. The shelf’s body frame is established by setting the selected points as reference points. In this example, the second point marks the origin of the body frame, the third point marks the 1-axis, and the first point marks the 2-axis. By clicking *Compute* in the attitude determination section, all of the attitude information is displayed.

**Future Work**

Extending from these results, there is much room for future work. An object tracking function could be added to allow attitude determination of moving objects in real time. This could be done using image processing tools such as corner, edge, or color tracking. Object tracking would allow a continuous attitude calculation instead computing the attitude for on frame. Also, more work could be done in describing the object. Being able to describe an object’s shape and attitude would be a useful application. Using many data points, it is possible to recreate an object’s shape in three dimensions.
References


