Design and Implementation of an Object Oriented Motion Sensing Camera

ECE 4600 Group Design Progress Report

Group 09
Grant Williamson
Trent Erskine
Jared Tarr
Cameron Whitton

Presented to:
Department of Electrical and Computer Engineering
Faculty of Engineering
University of Manitoba
Winnipeg, Manitoba, Canada

Faculty Supervisor:
Pradeepa Yahampath, Ph.D., P.Eng.

Reporting Dates:
September 6, 2013 - January 13, 2014
Submission Date:
January 13, 2014
# Table of Contents

1.0 Introduction ..................................................................................................................... 1  
2.0 Progress ............................................................................................................................ 1  
   2.1 Motor Controller ........................................................................................................... 1  
   2.2 Image Processing System ......................................................................................... 2  
   2.3 Physical Structure ..................................................................................................... 4  
3.0 Future Work ....................................................................................................................... 4  
   3.1 Arduino - Image Processing System Communication ............................................... 4  
   3.2 Motor Controller Calibration .................................................................................... 4  
   3.3 Image Processing System ........................................................................................ 5  
   3.4 Enclosure for Electrical Components ...................................................................... 5  
4.0 Conclusion ......................................................................................................................... 5  
Appendix A - Parts Ordered/Received ................................................................................ 6  
Appendix B - Revised Gantt Chart ...................................................................................... 7
1.0 Introduction
Object Oriented Motion Sensing Cameras (OOMSC) are used throughout industry for security purposes. Many cameras found in industry today record while remaining in a fixed position, only recording what is in their current field of vision. This project is to design and implement an OOMSC which is different from these standard security cameras. As an object travels through a small room, the camera will physically pan and tilt to center the object within its vision. When the object moves across the room the camera follows it until the object moves out of the camera’s range of motion.

The OOMSC utilizes a Microsoft Kinect for capturing images which are sent to a laptop computer with an Intel I7 processor and 8GB of ram for processing. The computer sends an updated positional coordinate to an Arduino Uno micro-controller which acts as the motor driver, adjusting the servo motors to pan and tilt the Kinect.

This project is as shown in Appendices A and B, currently under budget, on schedule and will be completed on time.

2.0 Progress
For the purpose of this report, the progress of the project can be separated into three sections. These include progress on the motor controller, image processing system and the overall system and its structure. All progress in the individual sections has been made by the team as a whole as the group has adopted a democratic approach to the entire project.

2.1 Motor Controller
The Arduino motor controller is programmed to control the movement of two servo motors based on an input received from the image processing system. The image processing system passes data to the Arduino via communication over a USB cable. Currently, the computer passes a number value pertaining to the pulse width required in order to rotate the shaft of each servo motor to its required position. This number is passed as a number ranging from 550-2350 with each value in between representing a different position of the servo motor shaft over its 180 degree range. There are two principal functions in the main loop of the Arduino code. Each function controls a different servo motor so that we can update both the pan and tilt servo motors
simultaneously. To do this, the software on the image processing system sends the Arduino two separate values separated by a pre-defined delimiting character. In the Arduino program, we have chosen to use a comma to separate the pan and tilt values. An example scenario of a value which the current image processing system can pass the Arduino is “700,1600”. The Arduino interprets this as a position value of 700 for the pan servo motor, and a position value of 1600 for the tilt servo motor. This task was started at the beginning of November and was completed mid-December.

2.2 Image Processing System
Three separate methods of image processing have been implemented and tested to date. These methods take images from the Kinect, analyze and alter them accordingly, then return the updated image for output. A simple flow chart of the image processing system can be seen below. While none of the three current methods are adequate for this project in their current state, the integral parts of a final method are contained in the code we created for these methods.

The first of these methods implemented was optical flow. Optical flow worked by finding important points contained in the area of interest and finding the location of these points in concurrent images. The locations of the important points are then be compared to find the distance that each important point has moved in the time it takes for the Kinect to capture two images. The average of all of these distances was then found and used to calculate a simple two-dimensional vector which is sent to the motor controller so that
it can adjust the servo motors as needed. Consequently, when testing this method, the results were found to be erratic. Too many points of interest were used and the outputted vector was rarely correct. Changes will have to be made if this method is to be used going forward, using less points of interest in a more centralized region to normalize the outputs we receive.

The next method implemented was facial recognition and tracking. This worked by comparing the image taken from the Kinect with a database of facial descriptors. If anything in the image matched one of the facial descriptors from the database, then a green rectangle was drawn around the area of interest. This rectangle can then be tracked and followed as it moves. However, upon testing we discovered that this has some drawbacks. Facial recognition and tracking is extremely resource intensive, as it noticeably slows the computer down. This method only works if there is a direct view of a face in the captured image as if the face is rotated too much or no face is detected, then the method fails. Facial recognition and tracking will not work as a primary form of user detection but can be used to supplement one of the other methods.

The final method that has been implemented to date is Motionblob tracking. It works by creating a reference image. The reference image is the mean difference of every image that has been captured up to the current image. The absolute difference is then taken of the current image and the reference image, where the difference represents movement in the image. By taking the moments of the difference, it is possible to determine the center point of the movement. This can then be used to determine how far the movement is from the center of the Kinect’s viewing area, which will allow for a vector to be created and sent to the motor controller to update the system coordinates. In spite of the promise this method showed, when tested one fatal flaw was found. We have determined that this method will not work with a moving camera as our system requires. Motionblob tracking is only viable with a stationary camera; therefore this method cannot be used going forward.

While none of these methods will work for our project, each of the methods contain code that can be used in the final iteration of the image processing system. The Motionblob tracking’s moment analysis and center of mass calculations will prove
valuable moving forward. Also, the facial recognition and tracking method may be deemed to be an integral secondary method for object detection. At this stage, there are no further plans to use the optical flow method. Work on the image processing system commenced half way through September and will continue until the end of January.

2.3 Physical Structure
A physical structure has been constructed to allow movement of the Kinect in a pan and tilt motion. The structure has been created using aluminum to reduce weight, which will prevent excess strain on the servo motors. Due to the large weight of the Kinect, compensation was required to allow the tilt servo motor to freely move the Kinect with its available torque output. Instead of the Kinect being directly attached to the tilt servo motor, the servo motor rotates a D-shaft in the structure which, in turn, rotates the Kinect’s mount. This D-shaft is connected to the tilt servo motor at one end and a ball bearing at the other. The Kinect’s mount is fixed at the central point of the D-shaft. In this configuration, the weight of the Kinect is distributed between the structure and the tilt servo motor. The pan servo motor is fixed to the channel housing the D-shaft (for the tilt servo motor), and rotates the entire structure. The design and construction of the physical structure took place throughout the month of December.

3.0 Future Work
Over the next two months, there are several tasks which must be completed. These include establishing communication between the image processing system and Arduino, calibrating the motor control code, further development of the image processing system and building an enclosure for the electrical components.

3.1 Arduino - Image Processing System Communication
In the coming week, we plan to develop a dictionary which will allow the image processing system, which is coded in java, to communicate with our Arduino motor controller, which is coded in a C variant. Communication will be established over a serial channel via USB.

3.2 Motor Controller Calibration
Once communication has been established between the image processing system and the Arduino, we need to find a way to translate distances, in terms of pixels in the
images, to degrees of rotation of the servo motors. This translation will be dependent on the environment in which the system is established and will require finding the correlation between motor controller commands and actual servo motor output movement in degrees.

### 3.3 Image Processing System

A final choice for the structure of the image processing system must be made. Facial recognition and tracking has proven not to be viable as a main source of object tracking, but could be used as a secondary system if needed. Motionblob tracking has also proven not to be viable for our needs but may be restructured to a form that could suit the needs of the project. Finally, our final method can be supplemented with the Infrared sensor of Kinect to further improve our results by creating a depth map that can be used in conjunction with portions of the former implementations of each of the methods.

### 3.4 Enclosure for Electrical Components

If time permits, our final plans for the project are to enclose all the wiring and electrical components for an aesthetically clean look. Ideally only the power supply cable for the electrical components enclosure, Kinect to image processing system USB cable and image processing system to Arduino USB cable should be visible outside the enclosure. The rectangular enclosure will be constructed using mild steel, aluminum or a plastic/polyurethane compound.

### 4.0 Conclusion

So far in this project we have completed three methods of image processing, established control of the servo motors using the Arduino motor controller and have designed and constructed a structure to integrate the servo motors and Kinect. In the next week we will establish communication between the image processing system and the motor controller followed by a calibration of the motor control system. Over the next month we will choose and refine one, or a combination of the image processing methods for the purpose of this project. If time permits we will then construct an enclosure for the electrical components. We are satisfied with the progress thus far and are confident that this project will be completed on time.
Appendix A - Parts Ordered/Received

Our updated budget to date is as follows:

Table 1: Budget

<table>
<thead>
<tr>
<th>ITEM</th>
<th>UNITS</th>
<th>UNIT COST</th>
<th>TOTAL COST</th>
<th>SUPPLIER</th>
<th>ORDERED</th>
<th>RECEIVED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micro-controller – Arduino Uno part #1050-1024-ND</td>
<td>1</td>
<td>$31.75</td>
<td>$31.75</td>
<td>DigiKey</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Power Supply (5VDC) part #285-1890-ND</td>
<td>1</td>
<td>$19.80</td>
<td>$19.80</td>
<td>DigiKey</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Power Supply (12VDC) part #285-1891-ND</td>
<td>1</td>
<td>$21.51</td>
<td>$21.51</td>
<td>DigiKey</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Microsoft Kinect</td>
<td>1</td>
<td>$109.83</td>
<td>$0.00</td>
<td>Group Members</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Motor Drivers part #497-1395-5-ND</td>
<td>4</td>
<td>$5.19</td>
<td>$20.76</td>
<td>DigiKey</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Servo Motors (6VDC) part #900-00005-ND</td>
<td>2</td>
<td>$14.44</td>
<td>$28.88</td>
<td>DigiKey</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Schottky Diode part #1N007</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>U of M Elect. Tech. Shop</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Prototyping Circuit Board</td>
<td>1</td>
<td>--</td>
<td>-</td>
<td>U of M Elect. Tech. Shop</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Board Mountable Terminal Blocks (120V Rated) part #ED1609-ND</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>U of M Elect. Tech. Shop</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Device Enclosure</td>
<td>1</td>
<td>$100.00</td>
<td>$100.00</td>
<td>U of M</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Pan/Tilt Camera Movement Structure</td>
<td>1</td>
<td>$50.00</td>
<td>$0.00</td>
<td>Group Members</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Terminal Blocks (120V Rated) part #ED3006-ND</td>
<td>2</td>
<td>$3.18</td>
<td>$6.36</td>
<td>DigiKey</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Cable Gland part #377-1631-ND</td>
<td>6</td>
<td>$2.08</td>
<td>$12.48</td>
<td>DigiKey</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>PC</td>
<td>1</td>
<td>$700.00</td>
<td>$0.00</td>
<td>Group Members</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Machine Shop Time</td>
<td>8 Hours</td>
<td>-</td>
<td>-</td>
<td>U of M</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Subtotal</td>
<td></td>
<td>$241.54</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parts Contingency</td>
<td></td>
<td>$100.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Miscellaneous Items (15%)</td>
<td></td>
<td>$51.23</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>$392.77</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix B - Revised Gantt Chart

The Gantt chart has been continuously revised as the project has progressed to account for any delays or changes in the initial proposed project schedule. We are pleased with the progress of the project thus far and feel that this updated schedule provides sufficient time to meet the stated external and internal deadlines.
Figure 2: Gantt chart created on January 12, 2014