ICRAFT

Eye-Controlled Robotic Feeding Arm Technology

Members

Pedro Lopes
Ryan Lavoie
Rishi Faldu
Nick Aquino
Jason Barron
Mohamed Kante
Basel Magfory
Waleed Meleis (Advisor)

TABLE OF CONTENTS

Introduction and Related Work

Assistive robotic arms are one of the most expanding markets in robotics today. According to research conducted by the Christopher & Dana Reeve Foundation (CDRF), approximately 1 out of 50 people are living with paralysis (about 6 million people). 36% of those who reported being paralyzed said they had "a lot of difficulty" in moving; 29% said "some difficulty"; 17% said "a little difficulty"; and 16% said they were "completely unable to move." Our project aims to improve the quality of life of people with disability by implementing an autonomous robotic arm feeder. Assistive feeding devices such as the Meal Buddy and My Spoon are commercially available and have been successful but they are very expensive and they move according to preset positions and trajectory. We developed a prototype of an assistive robotic feeding arm that is cost effective. Also, we aim to enhance the flexibility of previous feeding arms by using eye-tracking technology. This will allow the user to select which bowl of food they desire to eat out of in real-time, without outside assistance or the need to push a button. This is a particularly advantageous improvement over previous designs as it allows fully paralyzed users to start and stop the robot arm all by themselves. Using a monitor with a fixed position, the user will be able to use his or her eye movement to grasp food from one of the three bowls s/he decides to eat from. This provides an easy and effective system for those who may not have the use of their body from the neck down, giving them more self-sufficiency.

The Meal Buddy developed a "one click approach" to start and pause the robotic arm whenever the user wishes to. This is supposed to reduce the interaction between a user and the manipulator. However, if the user decides to switch from one bowl to another, a third party would have to assist and preset the motion accordingly, which takes away the desired sense of independence. Fortunately, our system will perform not only autonomous object handling, but also autonomous serving. The user will be able to look at a “rest spot” on the portable screen if s/he decides to pause the arm as well as look at the appropriate bowl number to command the robot to feed from it.

We will deliver a prototype of an autonomous feeding arm that will bring various foods to the user's mouth while using simple and inexpensive hardware and software. iCRAFT will cost approximately $800, which is significantly cheaper that the competition which cost at around $3500. This can happen because it only requires Windows, PC to operate, which we assume a majority of the people have.

Design Specifications

Our product will use a robotic arm kit that has base dimensions of 11.42 x 6.85 x 9.52 cm. The arm itself will closely resemble the dimensions of an adult arm, with segments of length 24.4 cm (shoulder to elbow) and 32.5 cm (elbow to wrist). The total weight of the entire robot arm is approximately 1.2 Kg and will be able to hold a maximum of 500 g at full horizontal reach. This amount is more than the amount of food that we would ever need to scoop, allowing this limit to not be an issue. The maximum horizontal reach is 60 cm and the maximum vertical reach is 69.4 cm. The robot arm has 4 degrees of freedom allowing rotation at the shoulder, elbow and wrist. Each of the servos has a maximum rotation of 1260 degrees, which reduces to 252 degrees at the final output shaft when it is attached to the 5:1 ratio gearbox. The structure of the arm allows for the wiring to be run through the inside of the arm with no interference with the gears. This will also allow us to run any other wiring through the arm if the need arises.

To control the servo motors on the robot arm will be an attached servo controller board. The board will serve as an interface between the users commands and the robot arm, being attached to the computer by way of a serial (RS232) cable. The servo controller board comes preprogrammed and loaded with the software and source code required to gain full control over the movement of the robot arm.

The main control of the arm is through a conjunction of the computer and the eye tracking technology. Our product will come with software that can be installed on any windows computer or laptop with the only requirement being that the computer has a USB port for the camera and a serial connector that is used to connect the computer to the servo control board and send commands to the board from the computer. All the movements of the robot arm are preprogrammed allowing the assistant to only need to set an eating position and the rest of the movement is carried out automatically once the user selects a bowl location.

The interface with the eye-tracking technology occurs with a computer monitor with the webcam mounted on top of it. The camera has an IR light that shines in the users eye and which allows the eye-tracking software to monitor the movement of the users eye. This program will seize control of the computer, acting as a mouse would for a normal computer and allow the user to select from the three bowl positions and initiate and conclude the feeding process. The feeding process works in the following manner, with the user first defining the feeding position, or the location of the users mouth where the food is to be brought to. The user then selects between the three food locations corresponding to different foods with the eye-tracking GUI appearing on the monitor behind the arm. The robot arm then acts out a scooping and feeding motion through preprogrammed sequences. This ensures the maximum amount of smoothness and a comfortable feeding speed to be achieved. The automatic program will bring the food to the defined feeding location so the user can move their head forward slightly and eat the food off the spoon. In between food selections the robot arm will return to a rest position, which is also the starting position the robot arm will be in upon first powering the unit up. The user also has the opportunity to select drinking while they finish their bowl selection.

Some of the requirements of iCRAFT is that the user must have minimal neck and head movement, which is usually not a problem even for C1 tetraplegics. Initial calibration assistance is required, which should't be a problem since the user should have someone already cooking the food for them. Solid foods need to be cut up into bit-sized pieces. Finally, the users would need a Windows based PC.

Design Overview

iCRAFT has two main parts. The computer which contains the interface for the eye tracking as well as the interface for the robotic arm. The computer includes the USB camera, and the external monitor. The robot arm includes the micro controller and the bowls and spoon.

The process starts with the user interacting with the eye tracking arrangement. The eye tracking arrangement will calculate where the user is looking on the monitor using a USB webcam and an IR LED which will shine in the user's eye. The computer is connected to the robot by a servo controller, which is connected by a serial connection. The servo controller is connected to the robot arm by a servo controller wire. The eye tracking will send a signal to the robot arm which will move to either of the three bowls or the rest position. The arm will then perform an automated action to get food onto a spoon. The spoon will then come to the eating position that is set up by the user and the assistant. The user will eat and can then choose to select another bowl, put food back, drink, or move the arm to the rest position again. Once a bowl is selected, we implemented a scooping algorithm that will scoop from different sides of the bowl so that the food is scooped evenly.

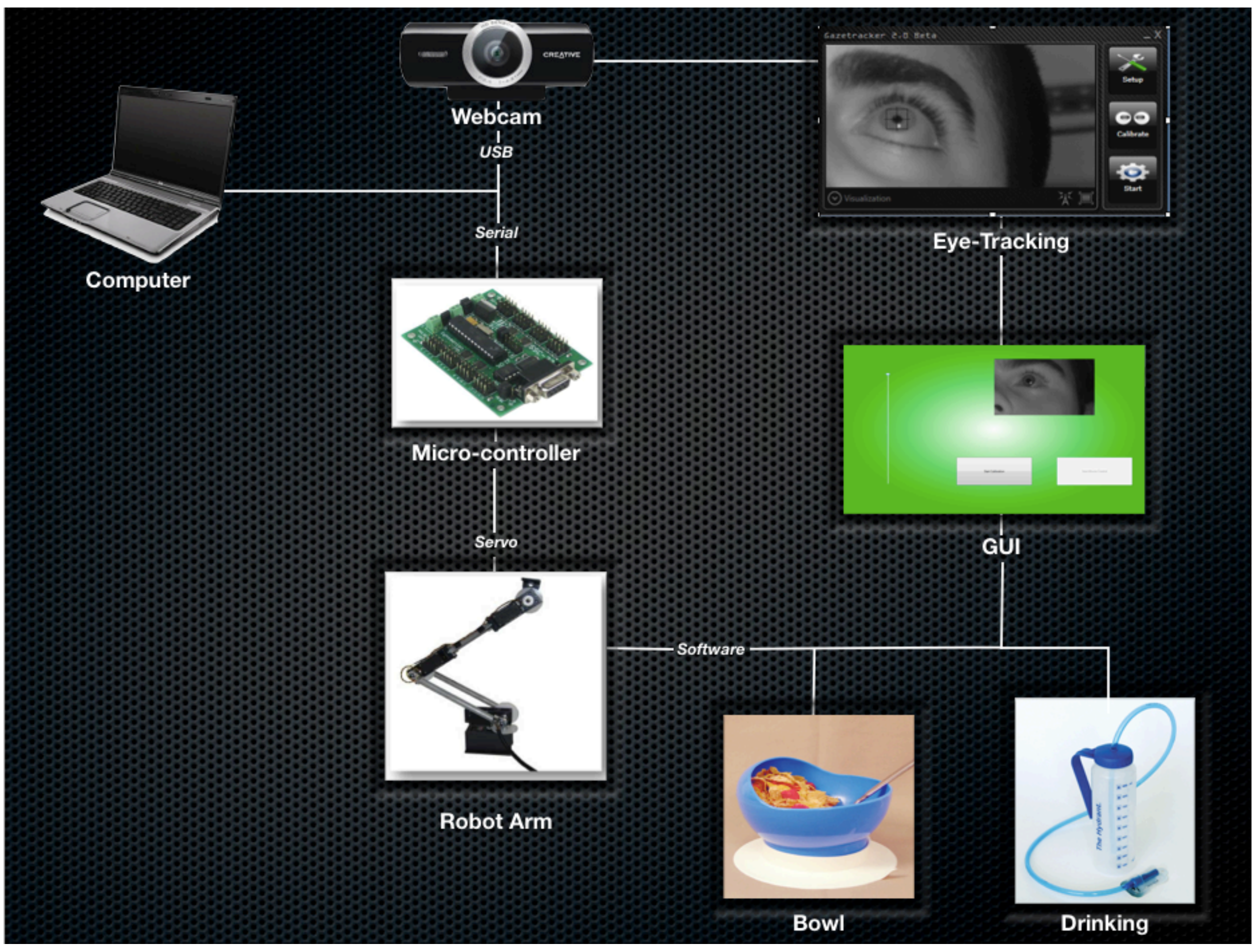

Figure 1: Design Overview

Using can be broken down into four steps. The first is the hardware setup which requires an assistant to the user to place the robotic arm, screen, bowls, and webcam in the correct places for the use of iCRAFT. The assistant can then put the food in the bowls and power on the robotic arm and PC. The second step is the software setup which will start automatically on the PC that has had iCRAFT software already installed on it. The eye calibration will be the first part of the setup and will be required for any usage beyond this point. Once the user has control over the mouse with eye tracking they can choose whether to use a new feeding position or a previously set feeding position. If the user wants to set a new feeding position they can do so using the eye-tracking UI so that there is less need for the assistant. The third step is food selection where the user controls the arm with eye-tracking. Once in the feeding UI (Figure 4) the user can select which bowl to eat from. While looking at the frame for bowl selection an indicator will begin to fill up to represent selection. Once full the robotic arm will perform the selected action. The final step is food retrieval. In this step the robotic arm performs the task from food selection with automated motions. If the option selected is to rest the arm then the robotic arm moves to rest position and waits for another selection. If the user selects a bowl to eat from the base of the arm will turn toward the bowl and check to see if that bowl has been eaten from three times. If so the arm will use the spoon to push food toward the center of the bowl before performing the scooping motion. The scooping motion is designed to pick up excess food then remove the excess with a bar installed on the scooper bowl. Afterward it brings the spoon back to the feeding position and waits (Figure 5) for the user to make another selection, return food, or select a drink. After the user is done eating the assistant can just remove the bowl and spoons to be washed.

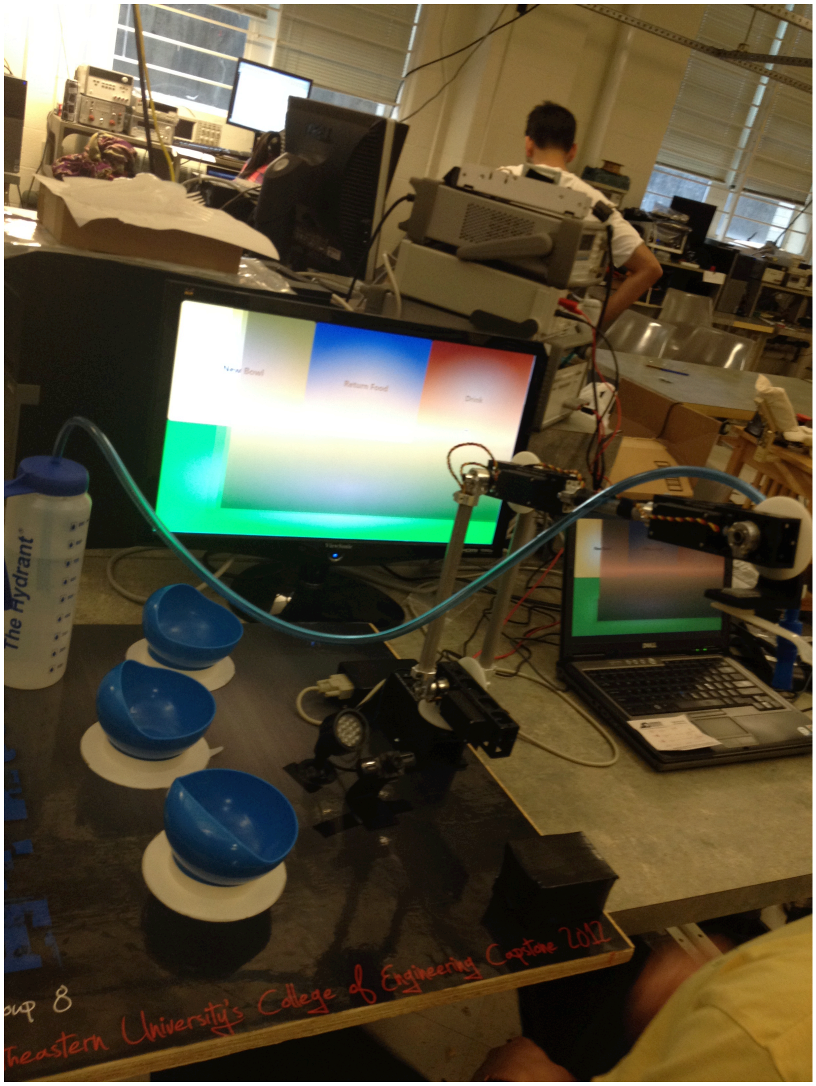

Figure 2: Representation of iCRAFT setup

Design Details

Robot Arm

The robot will be used in conjunction with a computer to control the interface between the eye tracking and the robot control. Since the program for eye control with be run on a PC, its integration into the control of the robot arm will be in the same program, meaning that when a selection is made a command will be sent out from the PC to the Lynxmotion SSC-32 Servo Controller (Figure 3b) which will take its commands by way of a serial cable. The servo controller is then able to make a move by taking the difference between its current position and the commanded position and uses that error to figure out how far of a move it needs to make. The servo controller board will allow us to control the rotation of the servos through creating PWM signals from the commands issued through conjunction with the provided software. Pulses between 600 microseconds (us) and 2400 us correspond to a position, while 500 us causes full speed rotation in the positive direction and 2500 us causes full speed rotation in the negative direction. The servo controller also has a “group move” function that will allow us to control more than one servo at the same time, which is necessary for creating the proper feeding motion. The board will require 6 V to power the on board micro-controller, and 6 volts will also be needed to ensure maximum torque in the servos on the robot arm. The servos are included in the RobotShop M100RAK Modular Robotic Arm Kit (Figure 3a). We will however need to select a proper power supply that can provide the 6 V to both the servos and the control board micro-controller, while also being able to handle a somewhat higher current rating, possibly even as high at 10 A peak. This current rating applies to the servo control board as well, but since the board is rated at 15 A continuous and 30 A peak it easily meets these requirements. The Lynxmotion servo controller is designed to work with Hitec servos, and they make the servos we plan on using so compatibility is ensured between the control board and the servos on the arm.

Included are the servo motors and the 5:1 ratio gearbox needed to ensure the torque is as high as required for our project. The servo motors included are 4 Hitec SH-785HB motors which will be connected to three SPG-785A-5.0 gearboxes at each joint, as well as one SPG785-BM gearbox at the base. The 5:1 ratio of the gearbox allows the torque at the output shaft to be five times higher than it originally is, while still providing 252 degrees of rotation.

Figure 3a: M100RAK Modular Robotic Arm.

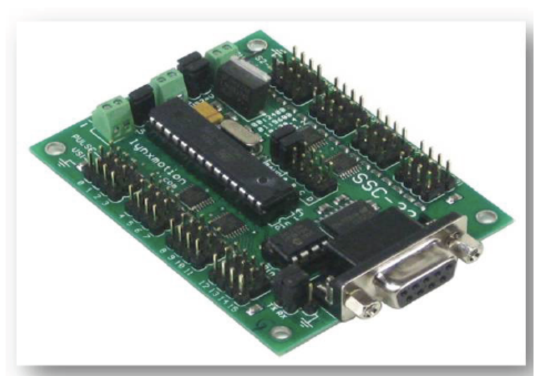

Figure 3b: Lynxmotion servo controller

Power Supply

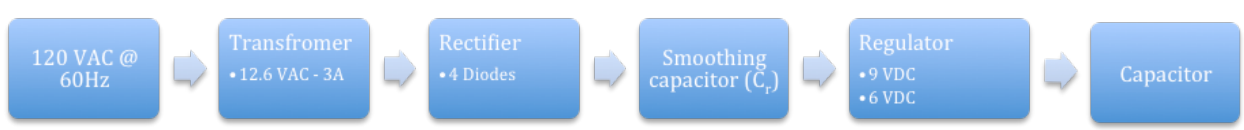

Figure 3c: Regulated Power System

Transformer:

We bought a heavy-duty 12.6V CT 3.0A Transformer that features tabs, pins and wire leads for easy mounting.

- Primary Voltage: 120VAC
- Includes tabs, pins and wire leads for easy mounting
- Secondary Voltage: 12.6CT
- Maximum Output: 3.0A

Rectifier:

4 individual diodes are connected as a bridge to convert AC to DC and to produce full-wave varying DC. Each diode is Rated 6 amps and 50 V.

Smoothing:

The varying DC output is not suitable for electronic circuits unless they include a smoothing capacitor. Smoothing is performed by a large value electrolytic capacitor (C_r) connected across the DC supply to act as a reservoir, supplying current to the output when the varying DC voltage from the rectifier is falling. The capacitor charges quickly near the peak of the varying DC, and then discharges as it supplies current to the output.

$$C_r = \frac{5 \times I_o}{V_s \times f} = \frac{5 \times 3A}{12.6v \times 60Hz} = 19841\mu F$$

Regulator:

The LM317 is a 3 Terminal Adjustable Regulator capable of supplying more than 1.5A over a 1.2V to 37 V output range. They are exceptionally easy to use and require only two external resistors to set the output voltage. Additionally, the chip includes current limit, thermal overload protection and safe area protection. The desired output voltage is calculated as followed:

$$V_{o1} = 1.25 \times \left(1 + \frac{R_2}{R_1}\right) = 1.25 \times \left(1 + \frac{1k\Omega}{160\Omega}\right) = 9.06\,V$$

$$V_{o2} = 1.25 \times \left(1 + \frac{R_4}{R_3}\right) = 1.25 \times \left(1 + \frac{380\Omega}{100\Omega}\right) = 6V$$

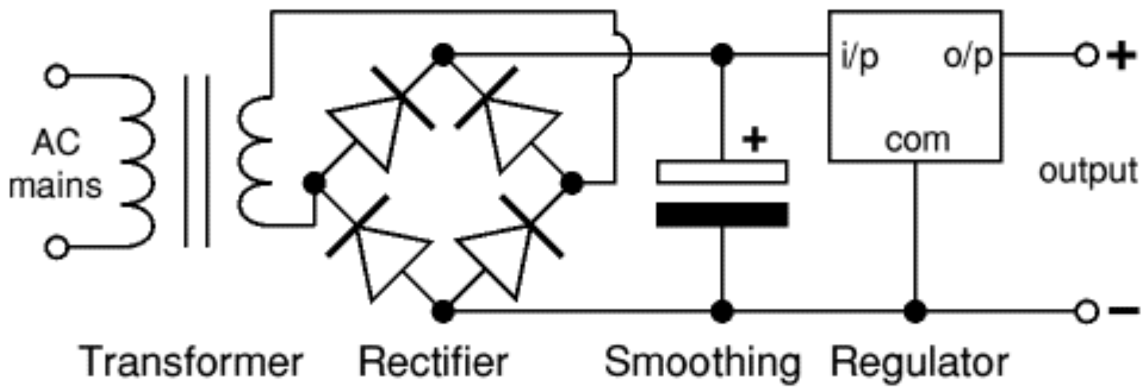

Figure 3d: Schematic

Capacitor:

Optional capacitor is added after each regulator to improve the transient response. The electrolytic capacitor (3300µF each) is used to provide improved output impedance and rejection of transients.

Eye Tracking

In order to eliminate the need to mechanically start the feeding process by pushing a button, our product will make use of eye-tracking technology accompanied by a webcam and computer interface/ portable screen. We will modify a webcam by installing a lens that does not filter infrared (IR) light. We will mount an IR light emitter on the table that will shine IR light on the user's pupils. The user's pupils will glow and reflect the IR light back towards the webcam. The webcam can then pick up the reflected light and estimate where the user is looking on the screen. Open source software (ITU Gazetracker) exists that can be used to calibrate the user's eye movement to control a mouse pointer on the screen. There is an option for nine, twelve, or sixteen points to depending on how much accuracy the user wants with his or her eyes. The cursor on the screen goes in a random order to avoid early movement of the eyes which will lead to inaccurate calibration. The screenshot below shows the cursor that the user follows with his or her eyes during calibration.

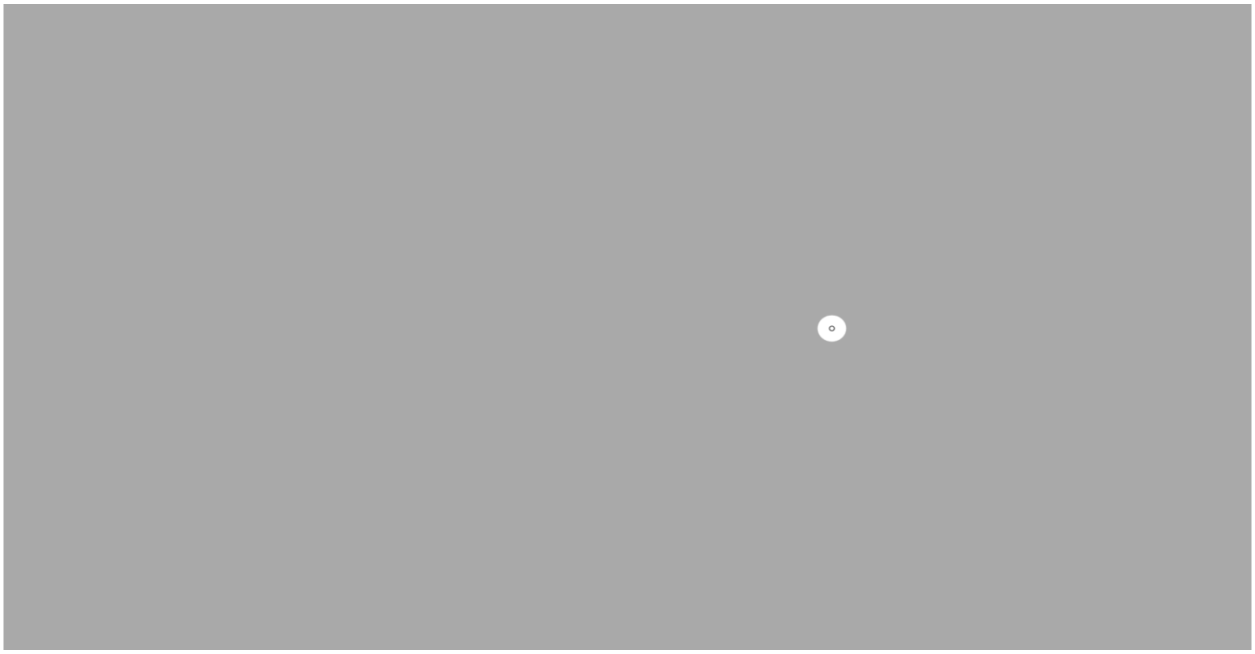

Figure 3e: ITU Gazetracker 2.0 Calibration (Start)

Once calibration is completed, a screen with how accurate the readings will come up, as shown below.

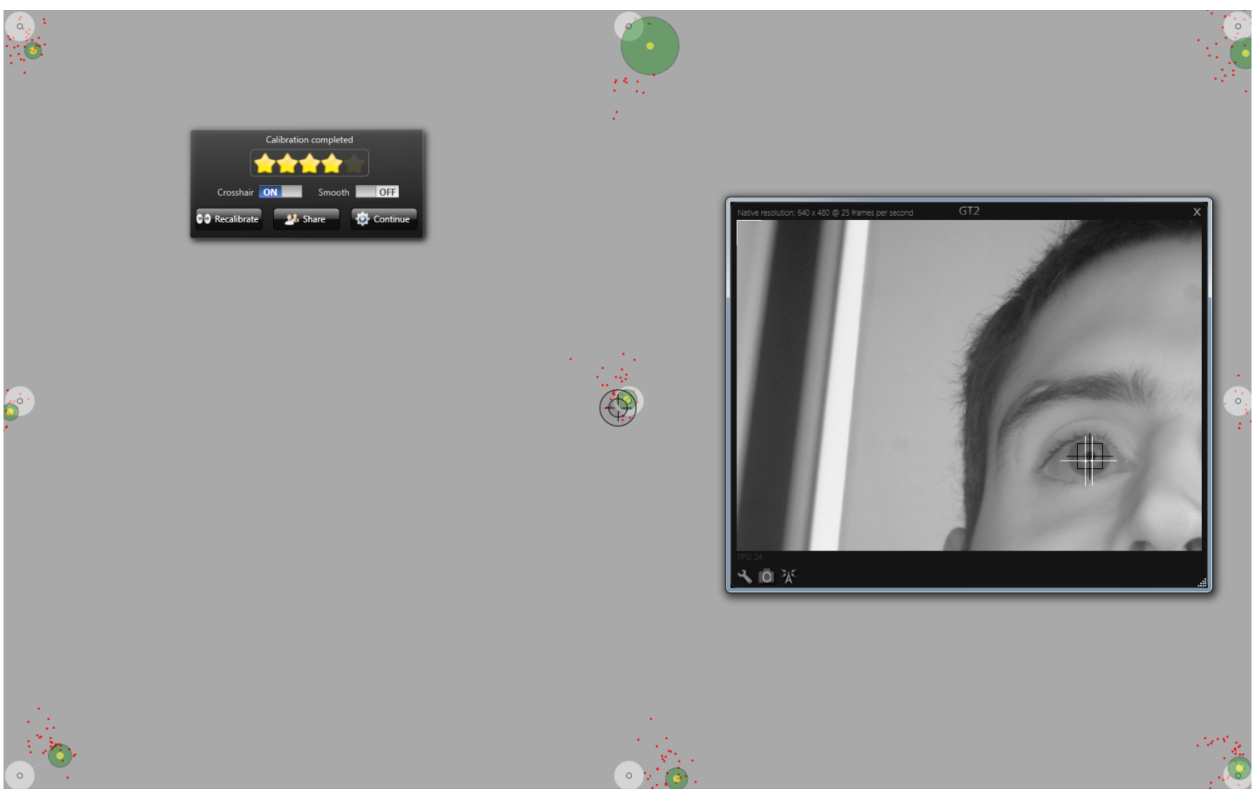

Figure 3f: ITU Gazetracker 2.0 Calibration (Verification)

If the scores are as desired the user can turn the mouse control on and move the mouse with his or her eyes. The mouse is unable to click but able to move to any desired location.

Figure 3g: ITU Gazetracker 2.0 (Ready)

GUI

The GUI was designed in XAML and implemented with C# for use on a Windows computer. The UI was optimized for eye control in which the mouse's movements are controlled by the ITU Gaze-Tracker software. Selection is controlled by mouse location and hover time.

The largest problem with eye controlled applications is the lack of precision when estimating gaze point as well as the inability to click to select. To solve the precision issue the layout had to be optimized for minimal calibration time without loss of reliability. The layout was designed in a 3 x 2 Grid across the full screen. This leaves large boxes that are 1/6 the size of the monitor giving the User plenty of leniency when selecting a box. The top three boxes (1x1) are used for primary selections and the Bottom rectangle (3x1) is used to rest eyes or move the arm to the Rest position. All boxes are a different color to easily distinguish between selection. There is also a label on each box to indicate the intended action when selecting it. This layout allows for only four possible selections at one time so we decided what four or less options were necessary to the user after any given decision and changed the menu for them while the previously selected action was being completed. This resulted in three different menus which we referred to as Bowl Selection, Eating, and Drinking.

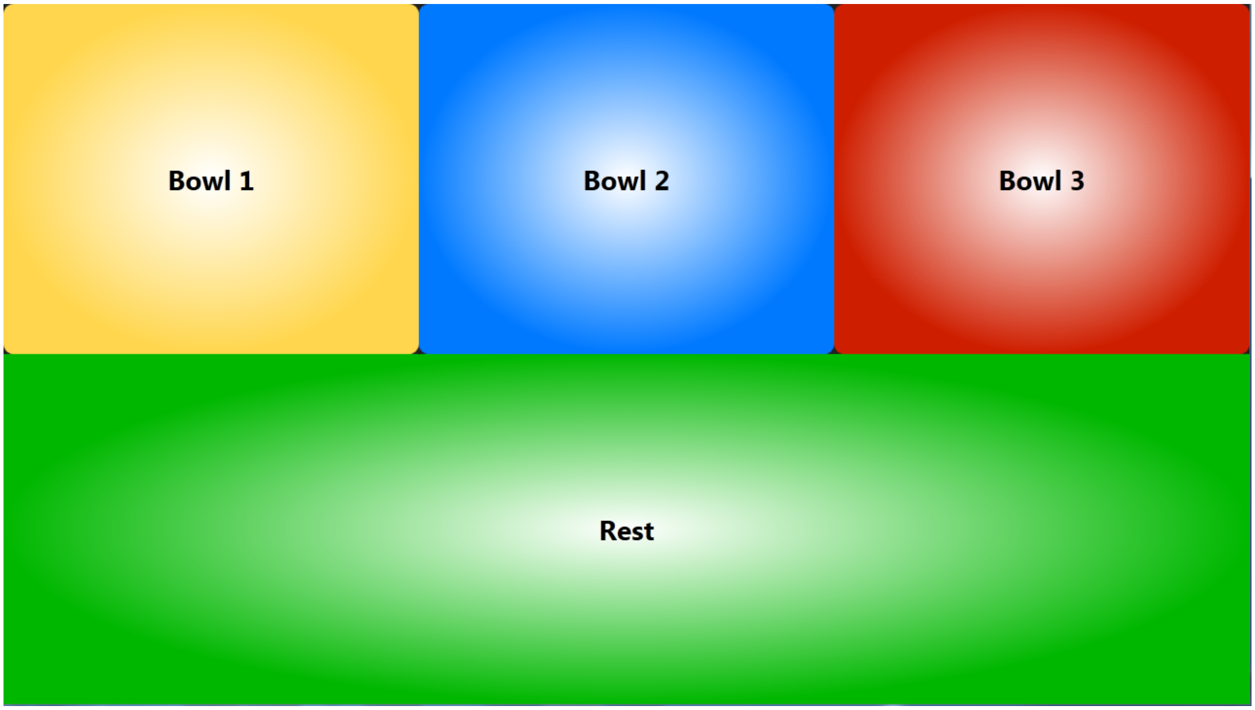

Figure 4a: Bowl Selection

On the Bowl Selection Menu the options allow the user to move the arm to the rest position or select which bowl to eat from. Once the Bowl is selected the arm will begin running its preprogrammed motion for scooping then rotate to the feeding position. After it reaches feeding position the Eating menu will appear. If Rest is selected the arm will move to the rest position and the current menu will remain the same.

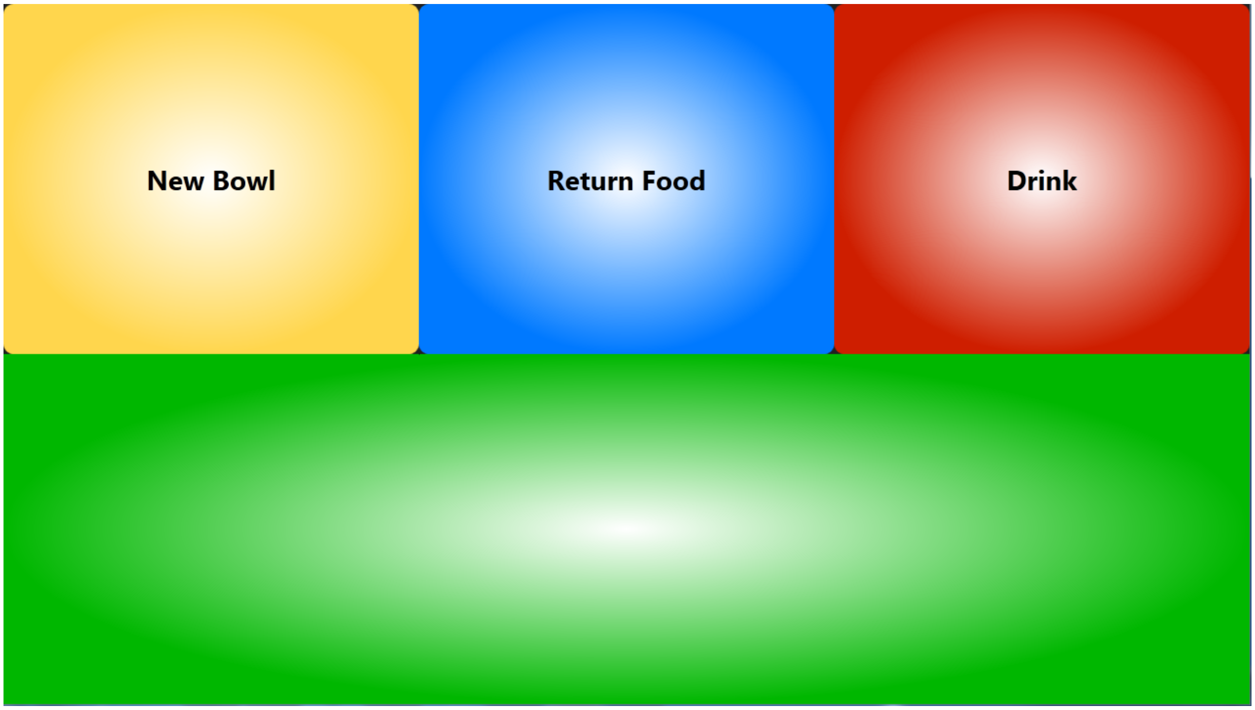

Figure 4b: Eating

On the Eating Menu the options allow the user to choose a new bowl to eat from, return excess food to its bowl, or move the arm to the drinking position. Once the action is selected the arm will begin running its preprogrammed motion while the menu changes the item to the appropriate selections. If New Bowl or Return Food is selected the Bowl Selection menu loads. If Drink is selected then the Drinking menu loads.

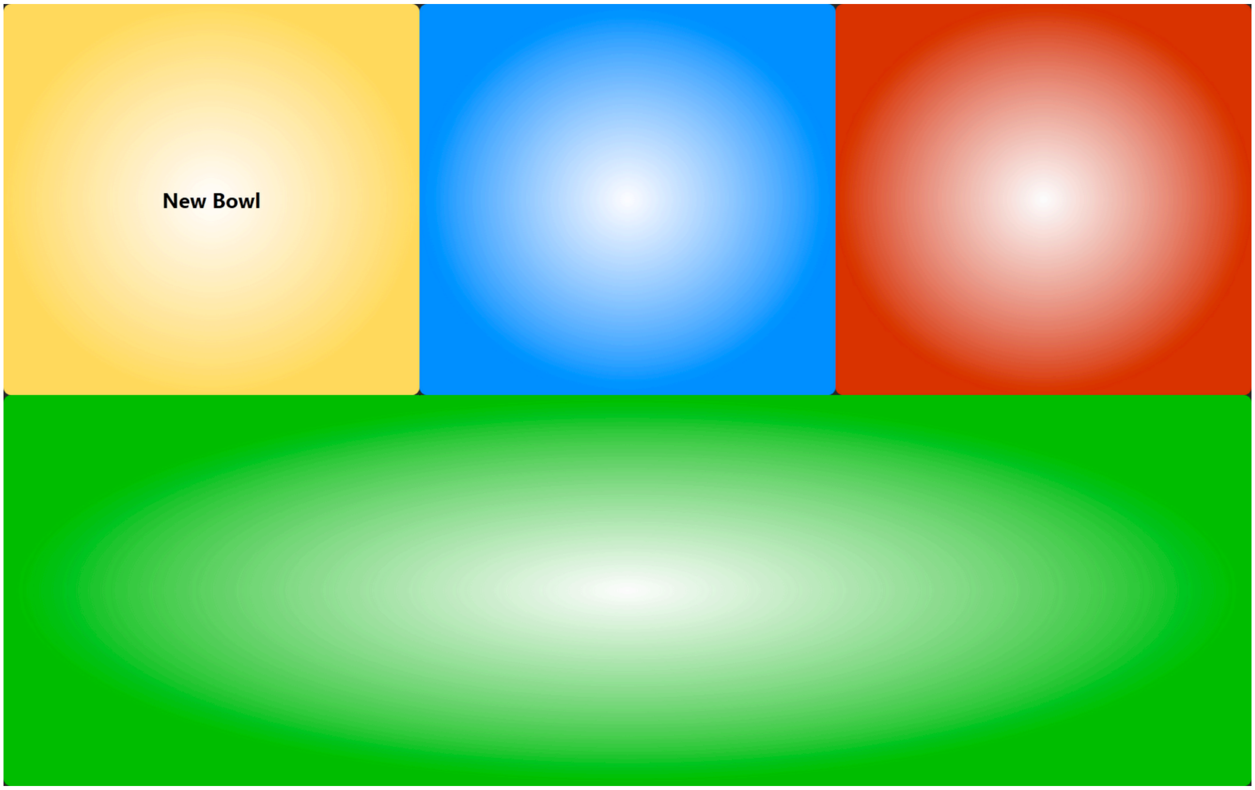

Figure 4c: Drinking

On the Drinking Menu the only option is New Bowl. Once selected it will move the arm to rest position and load Bowl Selection menu. This is in the Yellow Square, because it is easiest for the user to see while drinking. Every other area will do nothing

Using eye control to move the position of the mouse made clicking to select these boxes impossible. Our solution to this problem was a timed selection indicated by a gradient fill. When the user wishes to select a box all they have to do is stare at it for two-seconds. During this time a gradient will begin to fill the box from outside of the screen in. The user can look at another box to change their selection at anytime as long as the box has not been fully filled yet. If this is done the gradient will quickly disappear and the new box will begin to fill to indicate that it is now being selected. After a box is completely filled the gradient extends to fill the whole screen instantly preventing the user from making any more selections while the robotic arm is carrying out their desired action. Also to indicate this all text on the menu disappears behind the semi transparent gradient.

Figure 5a: GUI immediately after Selecting Bowl 1

During the carryout phase of selection when the gradient fills the screen there is a flashing status in the center of the screen. This provides the user with some feedback on the actions the arm is carrying out so they still feel in control of the arm even though they cannot currently choose anything. After the robotic arm has completed its action the text for the new menu will fade in and the gradient will disappear the screen to allow the user to begin selection again. To avoid accidental selection at this time the user can continue to look at whatever box their eyes were on without selecting it. Once they look at another box the selection process will begin again.

One of our overall goals for the GUI and iCRAFT in general was ease of use. This would make the experience more pleasant for both the user and their assistant. As far as GUI development is concerned this meant making the Setup processes for the assistant easy as well. We did this by creating a simple three step process that would allow the assistant to follow on screen instructions to get the User started.

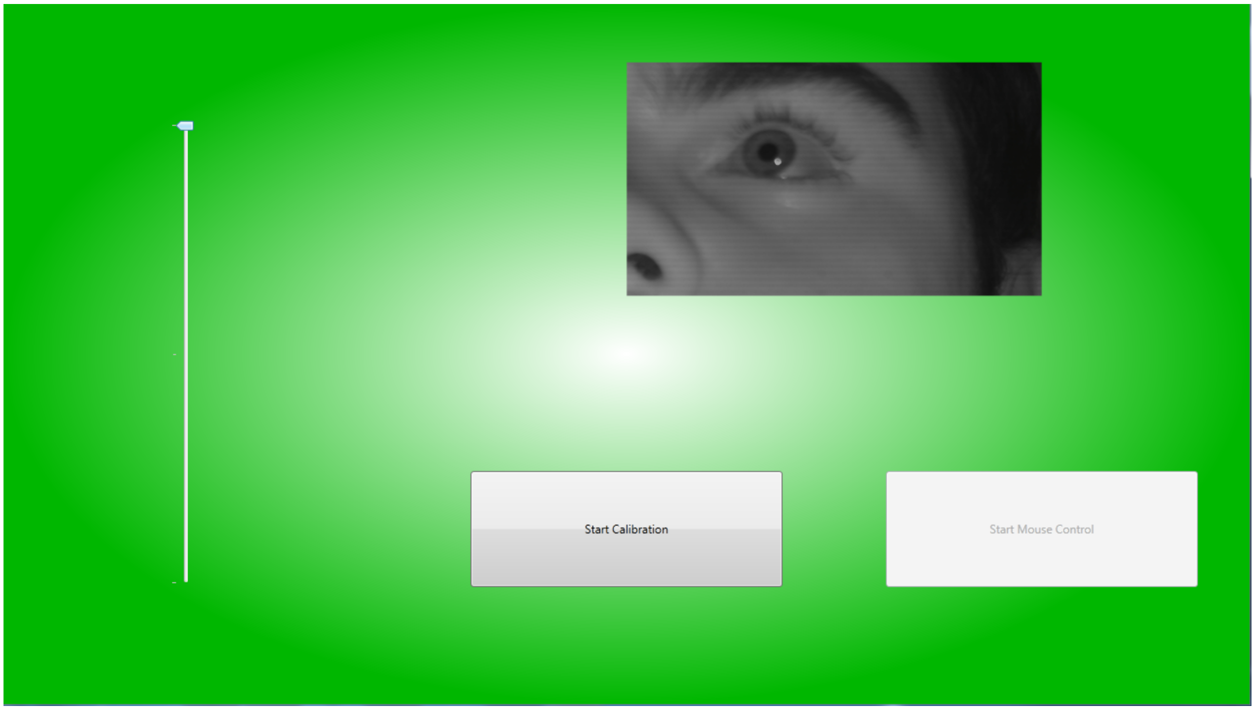

Figure 5b: Setup Menu

On the Setup Menu there is a stream of the webcam that will be used for eye tracking a Height Selection slider, a Start Calibration button, and a Start Mouse Control Button. We found that torso height did not vary much between people of greatly differing sizes so we decided upon a height selection slider to choose the feeding position of the arm. The options are simply Low, Medium, and High. Any height variance in between should not matter because the user must be able to move their neck. After Height is selected the assistant will begin the eye tracking calibration process. The video stream allows the user and the assistant to see if the user's eye is within the view of the webcam before starting the calibration. Once the Start Calibration button is selected the GUI will fade a red box in over the video stream and release the webcam. The ITU Gazetracker software will open up then the calibration will begin automatically based on the settings we found to be the most effective for use with the GUI. Afterward the Start Mouse Control Button will be enabled and the assistant can click it to begin eye control of the mouse based on the calibration settings. Once this is done all elements on the Setup Menu fade out and the Green background shrinks down to half the screen revealing the Bowl Selection menu and Becoming the rest box.

The GUI is very simple, easy to use, and aesthetically pleasing. This will make the process more user friendly for both the user and the assistant alike. The whole eating process was designed by us to meet these specific goals and completely eliminate any extra hassles that could arise when only using mouse position for control. Overall the GUI makes eating with iCRAFT a very comfortable and enjoyable experience for everyone involved.

Bowl and Drinking

We are using a scooper bowl that is available through the Internet. The bowl has a high rim and reverse curve on one side aid in scooping food onto a utensil without spilling. This is important to us because it would also help push the food onto the spoon, which makes the scooping transition smoother. Also, we are going to add a string on the top of each bowl to help remove excess food in each scoop. Since the robot arm is going to be designed to reach to {x,y,z} coordinates for each bowl, the bowls have to be in the same place every time the unit is used. The way we are going to solve this problem is by permanently mounting the suction cups (which come pre-assembled to each bowl) onto a tray that comes with the unit. This way it is easier for the user to place the bowls on top of the suction cups without needing to worry about placing each bowl in the right distance/coordinates. Due to the rounded shape of the bowl, the food is going to get stuck on the sides of the bowl. The way we are going to solve this problem is by including an algorithm that makes the arm push the food on the sides to the bottom of the bowl after three consecutive scoops from the same bowl.

We are planning on using a regular tablespoon to make it easier and affordable to the user. The spoon will be mounted on the tip of the robot arm. A clamp will be added to the robot arm tip to clamp the spoon in for easy removal of the spoon in the cases of washing or replacing.

Almost every time we eat we want to have a drink of something with our food. Because of that we decided to add a drinking feature to our project. We picked the Hydrant system for the job because it features the followings:

- 44” long tube to help us with placing the bottle almost anywhere without worrying about it getting in the way of the arm or the user.
- The tube has a leakage avoidance system represented in a mouthpiece that requires the user to bite to drink.
- It is plastic so it is durable.
- It has metric markings on the outside, which allows for easy and accurate monitoring of fluid intake.
- If the bottle is placed a foot higher than the table, it works as a siphon. So user only needs to bite to drink. That feature would help make it easy to drink for people with problem sucking on the tube. (Could be implemented in future design)

Figure 6: Scooper Bowl

Figure 7: The Hydrant Bottle

Scooping

Since there is no feedback system in our robot to detect food in the bowls we had to rely on the visual feedback from the user. However, since the user can't manipulate the arm any more than is allowed through the GUI there is no way to instruct the spoon where in the bowl to scoop from. These problems lead us to create a scooping algorithm that would automatically move the spoon to an appropriate location in the bowl. This worked by simply keeping track of how many times each bowl was scooped from, and slightly adjusting the positioning of our scooping sequence to the right and left of the middle. The first time each bowl is scooped from the positioning would be right in the middle of the bowl. Then, any subsequent scooping from that bowl would result in scooping from the left then right of this position. This allowed the food to move back to the middle of the bowl after 3 scoops. The fourth scoop would then start again from the middle and successfully scoop the remainder of the food out of the bowl.

Design Evaluation

iCRAFT was a success, we won the design competition. We ran through some bumps in the road, but we were able complete our goals. We also were able to implement the addition of the water bottle. We tested iCRAFT through multiple trials of solid and liquid foods. We had to figure out the scooping motion and placement of the bowls so it would be easier to implement. We implement one scooping motion for each bowl, but the location of the base is changed to the corresponding bowl location. Using this it made it easier to trouble shoot if we had any problems.

If we had more time and money to improve our project, we would try to implement a sensor that can detect your mouth always. This would help with height situations where now iCRAFT can accept any user with no problem. Also this would help with any safety concerns, where the robot will feed the user and not hit s/he in the face.

Parts

Part	Project Cost	Production Cost
RobotShop M100RAK Robot Arm	$598.85	$563.46
Lynxmotion SSC-32 Servo Controller	$41.94	$36.05
Scooper Bowl & Drinking Bottle	$55.23	$35.75
Custom Camera & IR	$114.74	$89.51
Power Supply	$16.30	$8.99
Misc./Aesthetics	$120.14	$73.94
Total Cost	$947.20	$807.70

Figure 8: Parts list w/pricing

References

Reeve Foundation. "Paralysis Facts and Figures." Internet: http://www.christopherreeve.org/site/c.mtKZKgMWKwG/b.5184189/k.5587/Paralysis_Facts__Figures.htm, [Jun. 8, 2011]

"Meal Assistance Robot- My Spoon." Internet: http://www.secom.co.jp/english/myspoon/, [Jun. 7, 2011]

"Meal Buddy- Robot Assistive Feeder." Internet: http://www.richardsonproducts.com/mealbuddy.html, [Jun. 7, 2011]