NGENUITY 3D Visualization Automation - Alpro

Team Name:Alpro
Group Members:

Abby Wysopal (Team Lead) - abbywysopal@ucsb.edu
Brent Luker (Scribe) - brentluker@ucsb.edu
Kiet Nguyen - knguyen104@ucsb.edu
Michael Hau - mdhau@ucsb.edu
Ryan Mitchell - ryan_mitchell@ucsb.edu

Introduction

Motivation and Background

Streamlining eye surgery through automation means that doctors will be able to bring improved vision to more patients and restore one of the most important human senses. With the prediction that “the world’s population will increase by about one-third” [1] and along with it, the demand for cataract surgery, optometrists will struggle to keep up with the demand. At the relatively low rate at which optometrists are trained, the demand will outpace the current solution. Automation of the NGENUITY 3D Visualization system will drastically cut down the amount of time medical professionals will need to spend on preparing the patient for surgery and in turn allow optometrists to help more patients.

Existing Solutions

The increase in cataract surgery demand is already causing great concern in the medical community and there are solutions currently being used to alleviate the problem. Beyond changes in administrative workflow, using lasers to make incisions increases consistency and accuracy while decreasing the amount of time to make the incisions, however even with advances in the actual surgical operation technology, systems using lasers will still benefit from decreasing the actual setup time through automation.

Core Components

Facial Detection

The facial detection will be used to both uniquely identify the patient and determine where precisely they are in the room so that the robot can prepare for surgery. The facial identification will be used to make sure that the patient on the table corresponds to the medical data the surgeon is currently using. The facial location will be used to position the robot accordingly.

Eye Detection

The eye detection algorithm will detect the eye from the face detected from the facial detection algorithm and make use of the existing focus and zoom functions on the camera to clarify the feed of the eye.

Simulation Environment

The simulation environment will act as an abstraction of the actual environment that the software will be used in. The simul8ation will include basic controls including: pan left and right to mimic changing x/y-coordinates and a zoom to mimic z-coordinates. The controls will be used on the video feed from the camera and be displayed in a very pretty front end.

Goals

The primary goal of the project is to decrease the amount of time the surgeon needs to spend situating and configuring the camera onto the patient before surgery; this end goal can be broken down into several key steps:

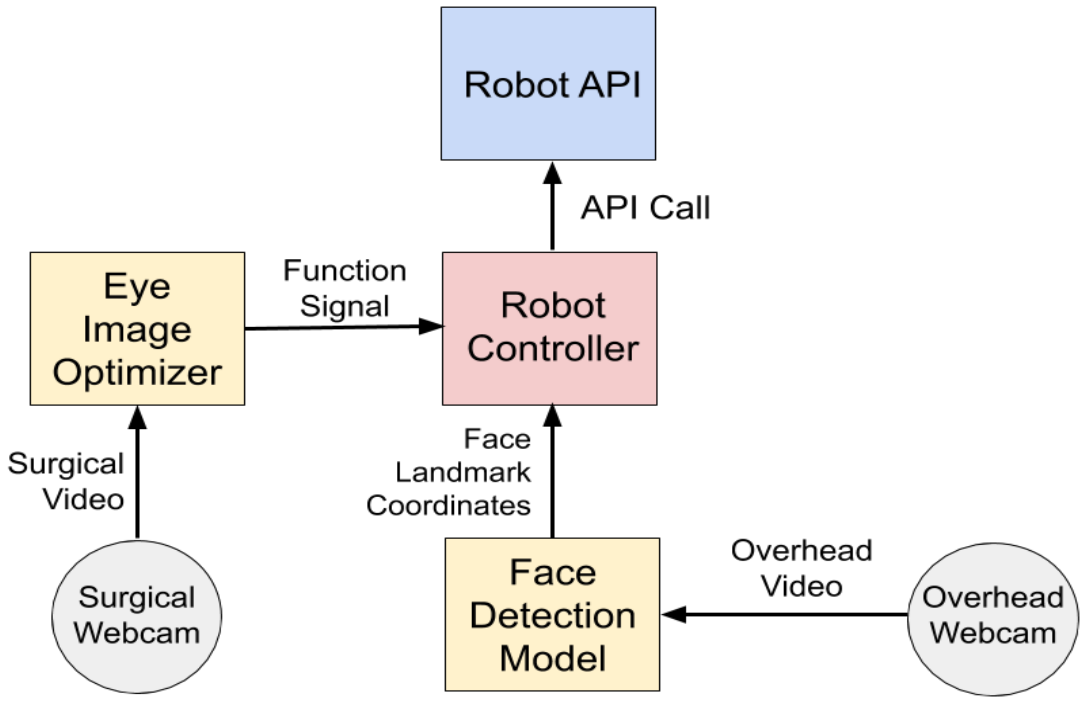
1. The robotic arm will automatically situate itself over the patient so that the camera on the arm will be able to detect the patient while ignoring other subjects in the room.
2. The camera on the robotic arm will zoom in and focus on the correct eye requiring surgery.
3. The overall experience for the operating team will be streamlined and user friendly while requiring almost no direct control.

Assumptions

Eye surgeries are expected to take place in a controlled environment with some variables but many invariants. The key invariants that the automation software will require include:

1. Only the patient will be facing upwards for the majority of the surgery, so that the bird's eye view camera will know which subject is the patient.
2. The lighting in the room will be adequate (adequate level of lighting is to be determined) for the camera to detect objects in the room.
3. Nothing will obstruct the patient's face during the facial detection phase.

Architecture Overview:



Requirements

1. **As a programmer, I can use a face detection model to accuractly identify the patient's face from a webcam video feed.**

   Github links (issues): (https://github.com/mycoal99/AIconCapstone/issues/1)

   a. Scenario 1: Patient is ready
      Given the patient is in the surgical chair, their face is accurately detected as the patient's face by the facial detection software.

   b. Scenario 2: No Patient in webcam
      Given there is no patient in the web camera, the facial detection software will determine which face has the highest probability of being the patient's face.

   Acceptance Criteria: Inputting a live video feed of the surgery room, the face detection model outputs which face is the patient's, and where the face is in the room.

2. **As a programmer, I can find the eye coordinates from a face found in an image**

   Github links (issues): (https://github.com/mycoal99/AIconCapstone/issues/5)

   a. Scenario 1: One eye in image
      Given a face has been found in the image and there is only one eye in the image, then I will return the coordinates for the eye in the image

   b. Scenario 2: Two eyes in image
      Given a face has been found in the image and there is two eyes in the image, then I will return the coordinates for the two eyes in the image

   Acceptance Criteria: Inputting an image of a person's face, the eye detection model outputs the coordinates of the eyes.

3. **As an Assistant Optometrist, I can press begin setup to initiate the setup process.**

   Github links (issues): (https://github.com/mycoal99/AIconCapstone/issues/8)

   a. Scenario 1: Ready for set up
      Given the patient is in the surgical chair and ready for surgery and the Assistant Optometrist begins the setup process, then the set up process will initiate and a successful set up will occur.

   b. Scenario 2: Not ready for set up
      Given the patient is not ready for surgery and the Assistant Optometrist begins

the setup process, then the set up process will initiate and an error message will occur.

Acceptance Criteria: Initiating the start sequence, the app controller will throw an error message or continue setup according to the situation of the room.

4. **As an engineer, I can load my webcam feed into the simulation so that I can test my Machine Learning model**

   Github links (issues): (https://github.com/mycoal99/AlconCapstone/issues/10)

   a. Scenario 1: Face Detection
      Given a video that a programmer wishes to test a face detection model on, the simulation can load and display the webcam video feed, as well as the results of the face detection model.

   b. Scenario 2: Eye Detection
      Given a video that a programmer wishes to test a face detection model on, the simulation can load and display the webcam video feed, and adjust the feed according to the signals of the eye detection optimizer.

   Acceptance Criteria: Inputting a webcam feed into the simulation, the machine learning models can run on the video feed and edit the video feed according to their respective goals.

5. **As an engineer, I can use image processing techniques to create a stabilized image**

   Github links (issues): (https://github.com/mycoal99/AlconCapstone/issues/9)

   a. Scenario 1: Stable frame with moving objects
      Given a video or a live video recording with stable frames in which objects such as hands or the eyes are moving, the simulation will accurately determine when to stabilize the frames.
   b. Scenario 2: Shaky frame with moving objects
      Given a video or a live video recording with shaky frames, the simulation will attempt to stabilize the frames so that the images will look better for human vision without rendering illusions.

   Acceptance Criteria: Inputting a live video feed of the patient's eye, the camera will remain stabilized on the eye as it moves around.

6. **As an engineer, I want to create a dataset to classify eye images**

   Github links (issues): (https://github.com/mycoal99/AIconCapstone/issues/7)

   a. Scenario 1: Ideal image
      Given an image is the ideal target image, it will be classified in the dataset as a good image

   b. Scenario 2: Bad image
      Given an image is not the ideal target image, it will be classified in the dataset as a bad image

   Acceptance Criteria: Inputting a set of eye images, the dataset will be correctly stored and labeled in the database.

7. **As an engineer, I can integrate eye detection software with the robot arm to move the surgical camera**

   Github links (issues): (https://github.com/mycoal99/AIconCapstone/issues/5)

   a. Scenario 1: Move Closer to Eye
      Given the signal from the eye detection software, the corresponding robot api function is called and the robot moves closer to the eye.

   b. Scenario 2: Move Further from Eye
      Given the signal from the eye detection software, the corresponding robot api function is called and the robot moves further from the eye.

   c. Scenario 3: Move West of Eye
      Given the signal from the eye detection software, the corresponding robot api function is called and the robot moves west of the eye.

   d. Scenario 4: Move East of Eye
      Given the signal from the eye detection software, the corresponding robot api function is called and the robot moves east of the eye.

   e. Scenario 5: Move North of Eye
      Given the signal from the eye detection software, the corresponding robot api function is called and the robot moves north of the eye.

   f. Scenario 6: Move South of Eye
      Given the signal from the eye detection software, the corresponding robot api function is called and the robot moves south of the eye.

Acceptance Criteria: Inputting the eye detection output into the robot api, the robot moves in the correct direction depending on the output of the eye detection model.

8. **As a software engineer, I can access live webcam stream to create a simulation**

   Github links (issues): (**https://github.com/mycoal99/AIconCapstone/issues/4**)

   a. Scenario 1: Webcam on
      Given the web camera is on, I can access the live webcam stream in my program

   b. Scenario 2: Webcam off
      Given the web camera is off, I will generate an error message

   Acceptance Criteria: The livestream will operate on a wifi network and be accessible through a very nice GUI.

9. **As a software engineer, I can zoom in on live webcam stream to simulate surgical camera z-axis movement**

   Github links (issues): (**https://github.com/mycoal99/AIconCapstone/issues/3**)

   a. Scenario 1: Closer to the Eye
      Given the live webcam stream in the simulation, I can call a function to zoom into the video and simulate robotic Z-axis movement towards the eye.

   b. Scenario 2: Further from the Eye
      Given the live webcam stream in the simulation, I can call a function to zoom out of the video and simulate robotic Z-axis movement away from the eye.

   Acceptance Criteria: With a live webcam streaming in the simulation, the video stream is zoomed in or out depending on what function is called.

10. **As a software engineer, I can move around on live webcam stream x axis to simulate surgical camera moving east and west**

    Github links (issues): (**https://github.com/mycoal99/AIconCapstone/issues/2**)

    a. Scenario 1: Move stream east
       If there is a signal to move the stream to the east, the camera's view should be shifted accordingly.
    b. Scenario 2: Move stream west
       If there is a signal to move the stream to the east, the camera's view should be shifted accordingly.

Acceptance Criteria: Inputting a live webcam stream, the programmer can move the live stream east and west.

Appendix

Tech Stack: Python, PyTorch, NGENUITY 3D Robot API, OpenCV, Javascript, Electron JS, Google Colaboratory, Google Cloud