CPSC 490 Project Proposal
Thomas Weng
Prof. Scassellati

Short Description
I want to build a system capable of visually detecting deictic (pointing) gestures and identifying the targets of these gestures. The system will be able to interface with robotics software to act as a perceptual component in social robots.

Motivation
The recognition of gestures is an important factor in understanding human social interactions. Humans use gestures intuitively to demonstrate their attention to points of interest, intention to perform future actions, and experience of emotions, often expressing information through these movements that is not provided through other means, such as speech. Social robots must have the capability to pick up on these cues to interpret and potentially mimic human behavior successfully.

Reliable gesture recognition, however, is a hard computer vision problem. Some of the technical challenges include classifying and tracking humans and objects in a scene, extracting three dimensional information from two dimensional images to get vectors (e.g. determining head orientation), and accounting for human imprecisions or abbreviations in movement (e.g. ambiguous pointing). Many of these challenges are not mathematically well-defined, so human intuition and machine learning techniques are often necessary to design and implement gesture classifiers.

Nevertheless, the difficulty of gesture recognition does not detract from its importance, and recent advances in hardware and software solutions for computer vision have simplified some of the stated challenges. The Microsoft Kinect revolutionized computer vision and robotics research by providing researchers with a low-cost sensor bar pre-packaged with many useful features such as skeletal tracking. Software packages such as Robot Operating System and OpenCV were both released within the last 15 years and have similarly made a significant impact on research in this area. These new technologies leave us poised to develop powerful and robust gesture recognition systems for social robots and other applications.

Project Description
My project will fulfill the goal of developing a pointing gesture recognition system that can easily interface with social robots as a perceptual component. I am focusing on recognizing pointing in particular as it is especially significant for social robots. Pointing can be used to teach and guide a robot by indicating objects and locations, opening up the ability to complete a range of tasks when coupled with vocal input and other cues such as head orientation.

The final product of my project will be a system that aggregates pointing gestures, head orientation, and simple vocal input to rank known objects in terms of how likely a human is pointing to a particular object. The system will use multiple pointing vectors to ensure that it captures the full range of human deictic gestures, e.g. head to hand-tip, elbow to hand-tip, etc.
The nature of vocal input for the system has yet to be determined, but will most likely take the form of simple prosody analysis or vocal commands—the goal here is to demonstrate the potential of the system to incorporate this data, not to build out a full vocal analytical engine. The algorithm for aggregating the different sources of input has yet to be finalized as well, but will most likely use techniques such as Euclidean distance of object from pointing vectors to assign confidence values for each object. As a test of the system, a robot should be able to use the gesture recognition system to point to the same object as a human.

I will build the system using the Microsoft Kinect v.2, ROS, and OpenCV. The Kinect is only compatible with Windows whereas ROS is only compatible with Linux, so an auxiliary goal of the project is to bridge the two technologies and make the data from the Kinect available in ROS using rostopics. Rostopics are the standard way for robotic systems using ROS to publish and subscribe to data. Note that while the Kinect v.1 has Linux-compatible software and drivers provided by the open-source community, the second version was released fairly recently and does not have the same support. A collaboration between Yale and a number of other universities through an NSF Expeditions in Computing Award in Socially Assistive Robotics has led to a software-based networking solution to bridging these two technologies, which I intend to use and extend in my project.

A secondary goal for the project is to take extensibility into consideration during development so that the system can be modified for use in related applications. For example, future researchers and users of the software may want to classify gestures other than pointing, or use a different object segmentation method to identify objects of interest. Documentation and source code will be key deliverables to promote this goal of extensibility.

**Implementation Details**

**Overview of system architecture**

The Kinect is connected to a Windows machine by USB. A C# program using the Windows SDK collects the data streams coming from the Kinect, such as Body Tracking, sends them over Wi-Fi to a ROS instance. Any objects of interest in the frames are also identified using OpenCV and their three-dimensional coordinates are also sent to ROS. A Python script takes the incoming data and publishes them to independent rostopics, to which other Python scripts can selectively subscribe.

A majority of the data evaluation will take place within the Python scripts that subscribe to the rostopics. To determine the direction of a point, for example, a script would subscribe to the rostopics providing skeletal coordinates and calculate a vector from the head point to the hand tip. These scripts will parse and aggregate the data from the Kinect, calculating direction vectors, registering vocal cues, etc. to output confidence metrics on how likely it is that a person is pointing to a specific object or location.

**Current state of the project and slated improvements**

I have implemented a minimum working pointing recognition system using the framework provided in the `kinect2_sensing` repository from the NSF Expeditions collaboration. The system outputs a message to the console when it detects that a point made by the right hand intersects a color-segmented object. The focus of the project will be to improve upon this system and make it fully featured and reliable. To name a few examples:
• Multiple vectors from both hands (e.g. elbow to hand tip, head to hand tip, hand to hand tip) should be used to determine the direction of a point, and confidence scores should be calculated for each vector and potential object.
• A vector indicating head or face orientation should be provided as another input to the system.
• The coordinate systems of the Kinect and social robot should be combined into a universal coordinate system. This merging could be accomplished by translating the Kinect coordinate system into the social robot's, or vice versa.
• Relatedly, a social robot using the universal coordinate system should be able to point to objects that humans point to.
• Vocal commands should be incorporated into the calculation of confidence scores.

System evaluation
I plan to evaluate the system by recording trials where humans point to specific objects and the system evaluates their behavior. We can then compare the system performance to human evaluation and calculate an accuracy percentage.

Potential stretch goals
• The current implementation allows the Kinect to communicate with ROS, but not vice versa. One enhancement to the system would be to allow messages to be sent from ROS to the C# program so that the social robot could request changes in the data it receives, e.g. requesting the depth associated with a particular surface for pick and place tasks.
• Another enhancement to the current specification would allow the system to recognize deictic gestures of multiple humans simultaneously. The Kinect v2 can recognize up to six users at once, and the current system only processes one.

Deadlines and Deliverables
By Feb. 1
• Complete literature review
• Research the capabilities of Kinect face and audio APIs
• Clean up and organize existing codebase
By Mar. 1
• Submit final proposal
• Enhance skeletal pointing with hand shape data and new vectors (e.g. elbow to hand-tip)
• Provide face and audio data to ROS through Kinect APIs
• Design a method to aggregate skeletal, face, and audio data into pointing salience
By Apr. 1
• Develop pointing salience system
• Run system evaluations
• Leave flexibility for slipped deadlines and stretch goals
By May 1
• Complete documentation and project write-up
• Publish source code, example executable(s), and video demo