6.111 Final Project Proposal

Overview

The overall system will be comprised of a video camera and a video monitor. The video camera will be positioned such that the hands of the user are in its frame of view. On the monitor, the user will see the real-time video of the input camera. Therefore, when the user moves his hands, this movement will be displayed on the monitor in real-time.

Superimposed on top of the real video will be a picture of a ball that is added by the system. This ball will be moving on the monitor and the user will be able to interact with it by moving his hands. The system will treat all dark pixels on the monitor as being solid. The background color behind the user's hands will be white therefore the hands will appear dark on the video monitor. When the ball on the monitor comes into contact with the picture of the user's hand, it will bounce off of it as if it were a solid object. In this manor, the user can interact with the picture of the ball on the monitor by moving his hands and "touching" it.

Given this basic framework, the system can be easily made into a game such as air hockey or pong. The user's hands will act as the paddles that the virtual ball bounces off of. In such a case, the system will have two users (players) that are facing each other. The camera will be placed such that both users' hands are in the frame of view and on opposite sides of the screen. The ball will bounce between the two sides and if it passes the edge of the screen on either side, the opposite side is awarded a point. This is the same concept as pong or air hockey. Namely, this implies that one of the players failed to block the ball with his hands.

High Level Implementation

Since the system will recognize all dark pixels from the camera as solid objects, the actual playing field of the game can be created by making physical markers on the background. If these markers are dark, the system will interpret them as being solid. Therefore, to make the outer walls of the playing field, black lines can be placed on the background. In fact, physical objects can be placed in the frame of view of the camera and the ball will bounce off of the object's image. This results in a highly scalable and general system that can be changed and modified easily.

The heart of the system is in the edge-detection and physics algorithms. To make this process simpler, the initial design will have a very basic algorithm. If necessary, this can be changed in the future. Specifically, the system will assume that the ball moves at a constant speed in one of 8 set directions in the x-y plane. The ball is divided into four quadrants. The system will perform overlap detection on each of these quadrants with the image from the camera. If it is determined that one of these four quadrants is overlapping a dark pixel (solid object), it will determine the new direction that the ball should be moving in by performing a table lookup. Namely, since there are only 8 directions and 4 quadrants, all possible outcomes can be hard coded into a lookup table. Given a quadrant and an incoming direction, the table value will represent the outgoing direction. Though this algorithm is simplistic and not very

elegant, we hope that it will be sufficient for the initial design. If necessary, this can be changed without affecting the rest of the system.

Using such an algorithm makes implementation much simpler. Namely, a very general system (one that allowed the ball to move in all directions and at different velocities for example) would require a large amount of computation and fairly complex algorithms. By making the system less general, the algorithm becomes much simpler. The algorithm discussed above is very easy to implement and we hope that it will produce a sufficiently convincing output.