487 Documentation

Sam Hoover and Jordan Soltman

Goal

The ultimate goal of this project was to develop a pong-like game which was controlled by motion tracking the player's hands. We wanted to attempt several different methods of tracking to see which would perform the best. Our goal was to implement a color based version, a motion based version, and ultimately a method that would combine these two ideas and use both color and motion. We wanted all of the processing to be fast enough that the game could be played without too much lagging or delay.

Accomplishments

Color

The color mode operates by searching for the largest blob of color in a given hue, saturation, and value range. The color detector starts in configuration mode. In this phase, the thresholded image is shown and you can adjust the sliders to pick the hue, saturation, and value ranges that best isolate the object that will be used for tracking. Once these are set, game play begins. Each frame of video is first converted to HSV. From there the image is thresholded with the parameters defined in the configuration stage. Then, using OpenCV's moments() function the mass center of the blob is found. This point represents the point where the paddle should be and is transmitted to the pong game for use in game play.

Motion

The motion mode operates by taking sequential frames of video, flipping and converting each to grayscale. The absolute difference in the two grayscale frames is then computed. The difference image will show white smudges which indicate motion between the two frames, often times the smudges are very small. To increase the visibility of these smudges a binary image is created. To do this the difference image is thresholded and blurred to create an intensity image which is thresholded again in order to create the binary image. The binary image, which will show much more pronounced smudges, is then used to detect motion between the frames.

In order to track motion for two players in a single frame each frame was split in half along the y-axis and objects in motion were tracked separately in the left and right halves of the frame. Tracking was done by finding all contours in in each half of the binary image. Each contour represents the size of each object in motion. For simplicity, the largest contour was selected as our object to be tracked. A bounding rectangle was the created around the largest contour, the center of which is used as the point being tracked. This point is used to move a player's paddle up or down. Because the x-position of each of the player's paddles is fixed, only the y-coordinate of the tracked object if actually needed for repositioning a player's paddle.

Game Board

The game board operates much like a standard version of pong. There are two paddles and a ball. Each paddle can move vertically and the ball can move horizontally and diagonally in each direction. The game begins with the ball in the center, travelling to the right. If the ball strikes a paddle it will begin travelling the opposite direction, either horizontally or diagonally depending on the position of the impact with the paddle. If the game ball strikes an x-boundary, then the player on the opposite side of the boundary scores a point. After a point is scored, the ball resets to the center travelling right with increased speed. The game concludes when one of the players reaches seven points.

On the background of the game board, the video feed being used for tracking is displayed with red and blue crosshairs indicating the points being tracked for each respective paddle.

Results

Color

Color for the most part works very well as a paddle detector. It does have some problems though. Changing light conditions as well as interference in the background can easily throw off the paddle. For objects that are shaded dramatically different from other objects in the scene (such as a tennis ball) the algorithm can easily keep up, even if the lighting changes. Hands however only work well when there isn't any other skin in the frame. Skin tones also generally require a tighter threshold on the hue, saturation, and value since they are not as distinguishable as other colors. This means that any change in lighting will likely cause the hand to go out of the threshold range and control will be lost.

Motion

The motion mode was less reliable than the color mode for object tracking. This was mainly due to the motion detection in the motion mode using the largest contour as the object to track, instead of a more precise method. This made the motion mode very susceptible to background movements, especially large objects. The best results occur when there is no background motion and the only objects in the frame are the player’s hands.

Color and Motion

We spent a lot of time thinking about how to get a color and motion algorithm to work, but were never able to develop an algorithm that worked at all. We had a couple of proposed solutions to this problem.

Average of Centers

The most obvious answer to us was averaging the centers of the motion detection algorithm and the color detection algorithm. While at first this seemed like it would work, we realized that we were subjecting the more stable color detection to less stable motion detection and the end result was just a less stable color detection algorithm.

Tracking the Motion of a Colored Object

The other solution that we considered was first creating a thresholded image that would isolate objects of a certain color, and then doing motion detection on that. This introduces similar problems. If the object isn’t moving, then no motion is detected, and the paddle will still jump around.

Lessons Learned

In terms of computer vision, we learned two methods for tracking/detecting motion in live video feeds. This required first learning how to capture and process video using openCV.

Overall, the color mode seems to be the better solution for tracking an object when compared to motion mode. Color is far more accurate and precise and is less susceptible to background movements. The motion mode seems to be a good method for detecting moving objects in a frame, but not necessarily for tracking moving objects.