MotionTracking

CODE

```
import processing.video.*;

int numPixels;
int[] previousFrame;
Capture video;

void setup() {
  size(640, 480);

  // This the default video input, see the GettingStartedCapture
  // example if it creates an error
  video = new Capture(this, width, height);

  // Start capturing the images from the camera
  video.start();

  numPixels = video.width * video.height;
  // Create an array to store the previously captured frame
  previousFrame = new int[numPixels];
  loadPixels();
}

void draw() {
  if (video.available()) {

    // When using video to manipulate the screen, use video.available() and
    // video.read() inside the draw() method so that it's safe to draw to the screen

video.read(); // Read the new frame from the camera
    video.loadPixels(); // Make its pixels[] array available

    int movementSum = 0; // Amount of movement in the frame
    for (int i = 0; i < numPixels; i++) { // For each pixel in the video frame...
      color currColor = video.pixels[i];
      color prevColor = previousFrame[i];

      // Extract the red, green, and blue components from current pixel
      float currR = red(currColor); //(currColor >> 16) & 0xFF; // Like
red(), but faster
      float currG = green(currColor); //(currColor >> 8) & 0xFF;
      float currB = blue(currColor); //currColor & 0xFF;

      // Extract red, green, and blue components from previous pixel
      float prevR = red(prevColor); //(prevColor >> 16) & 0xFF;
      float prevG = green(prevColor); //(prevColor >> 8) & 0xFF;
      float prevB = blue(prevColor); //prevColor & 0xFF;

      // Compute the difference of the red, green, and blue values
      float diffR = abs(currR - prevR);
      float diffG = abs(currG - prevG);
      float diffB = abs(currB - prevB);
```

```
/**
 * Frame Differencing
 * by Golan Levin.
 *
 * Quantify the amount of movement
in the video frame using frame-
differencing.
 */
```

```
      // Add these differences to the running tally
      //  movementSum += diffR + diffG + diffB;
      // Render the difference image to the screen

      if (diffR + diffG + diffB>8) {
      pixels[i] = color(0, 0, 0);
      } else {
         pixels[i] = color(255, 0, 0);
      }

      // The following line is much faster, but more confusing to read
      //pixels[i] = 0xff000000 | (diffR << 16) | (diffG << 8) | diffB;
      // Save the current color into the 'previous' buffer
      previousFrame[i] = currColor;
    }

    // To prevent flicker from frames that are all black (no movement),
    // only update the screen if the image has changed.
    // if (movementSum > 0) {
        updatePixels();
    // println(movementSum); // Print the total amount of movement to the console

  }
}
```