

motiondetection

//Tracking of a green color across the footage resulting in Y-coordinate value

```
import processing.video.*;
```

```
// Variable for capture device
```

```
Movie movie;
```

```
//FrameRate
```

```
int fps = 25;
```

```
float dur;
```

```
float at;
```

```
float prevTime = 0;
```

```
boolean done = false;
```

```
boolean saved = false;
```

```
boolean first = true;
```

```
//Output
```

```
ArrayList<String> list = new ArrayList<String>();
```

```
String n = "22";
```

```
String fileName = n+".txt";
```

```
String movFile = n+"-belt.mp4";
```

```
// How different must a pixel be to be a "motion" pixel
```

```
float threshold = 0;
```

```
// Previous Frame
```

```
PImage prevFrame;
```

```
PVector centerOfMotionPrev = new PVector(0,0);
```

```
void setup() {
```

```
size(1440, 1080);
```

```
movie = new Movie(this, movFile);
```

```
//movie.speed(0.5);
```

```
movie.play();
```

```
dur = movie.duration();
```

```
}
```

```
void movieEvent(Movie movie) {
```

```
movie.read();
```

```
}
```

```
void draw() {  
background(0);
```

```
at = movie.time();  
if(at >= dur){  
  done = true;  
}
```

```
// You don't need to display it to analyze it!
```

```
image(movie, 0, 0);
```

```
PVector centerOfMotion = GetCenterOfMotion(movie);
```

```
//Save to file
```

```
if(done && !saved){  
  list.add(Float.toString(movie.time())+", "+Float.toString(movie.height - centerOfMotion.y));  
  saveStrings(fileName, list.toArray(new String[list.size()]));  
  saved = true;  
}else if(!first){  
  if(movie.time() != prevTime)  
    list.add(Float.toString(movie.time())+", "+Float.toString(movie.height - centerOfMotion.y));  
  prevTime = movie.time();  
}  
else{  
  list.add("Time,Motion");  
  first = false;  
}
```

```
//Draw a circle based on average motion
```

```
stroke(204, 102, 0);  
//fill(204, 102, 0);  
//float r = 100;  
//ellipse(centerOfMotion.x, centerOfMotion.y, r, r);  
line(0, centerOfMotion.y, movie.width, centerOfMotion.y);
```

```
textSize(100);
```

```
fill(0, 102, 153);
```

```
//text(avgMotion, 10, 100);
```

```
text(centerOfMotion.y, 10, 100);
```

```

text("Done: "+String.valueOf(done), 20, 200);
text(movie.time(), 10, 300);
text(movie.duration(), 10, 400);
}

```

Operations

```

PVector GetCenterOfMotion(PImage a){
    if(a!=null){
        PVector centerOfMotion = new PVector(0, 0);
        int motionPixels = 0;

        //println(a.pixels.length);
        a.loadPixels();

        for (int i = 0; i < a.pixels.length; i ++ ) {
            color current = a.pixels[i];

            float r1 = red(current);
            float g1 = green(current);
            float b1 = blue(current);

            //White
            /*if((r1>190 && r1<220) && (g1>190 && g1<220) && (b1>190 && b1<220)){
                centerOfMotion.add((i+1)%a.width, (i+1)/a.width+1);
                motionPixels ++;
            }*/

            //Green
            //General&Light (LED on only restricted)
            if((r1>10 && r1<80) && (g1>70 && g1<180) && (b1>10 && b1<80) && (dist(r1, 0, b1, 0)< 20) &&
(dist(r1, 0, g1, 0)> 50)){
                //Dark (LED off included)
                //if((r1>10 && r1<25) && (g1>40 && g1<70) && (b1>10 && b1<25) && (dist(r1, 0, b1, 0)< 20) &&
(dist(r1, 0, g1, 0)> 25)){
                    centerOfMotion.add((i+1)%a.width, (i+1)/a.width+1);
                    motionPixels ++;
                }
            }
        }
        return (motionPixels > 0) ? centerOfMotion.div(motionPixels) : new PVector(0, 0);
    }
    return new PVector(0, 0);
}

```

