Week 6

VIDEO PROCESSING

Digital/Software Mirrors

- A type of applications that provide a digital reflection of a viewer’s image
- Read and manipulate camera pixels

Access Webcam Pixels

- Use either camera.get(x, y) or camera.pixels[x+y*width] to get the colour at pixel coordinates (x, y)
- See Example 1

Converting (x, y) into an index in pixels[] array:

```
pixel_index = x + y * width
```

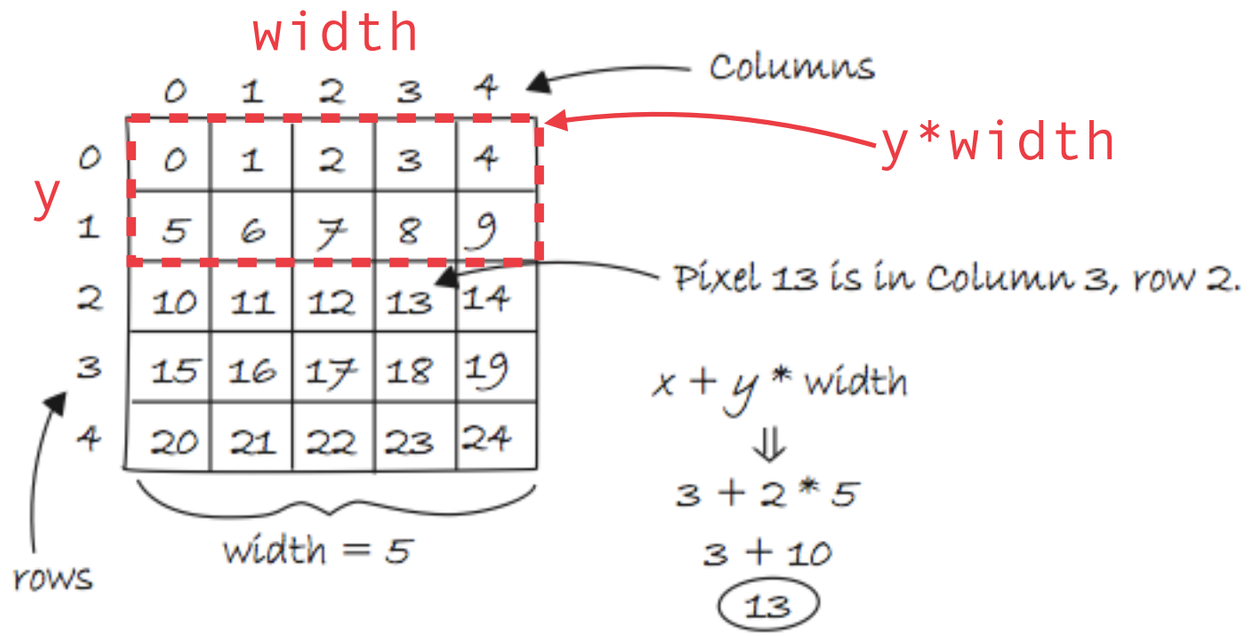


fig. 15.7

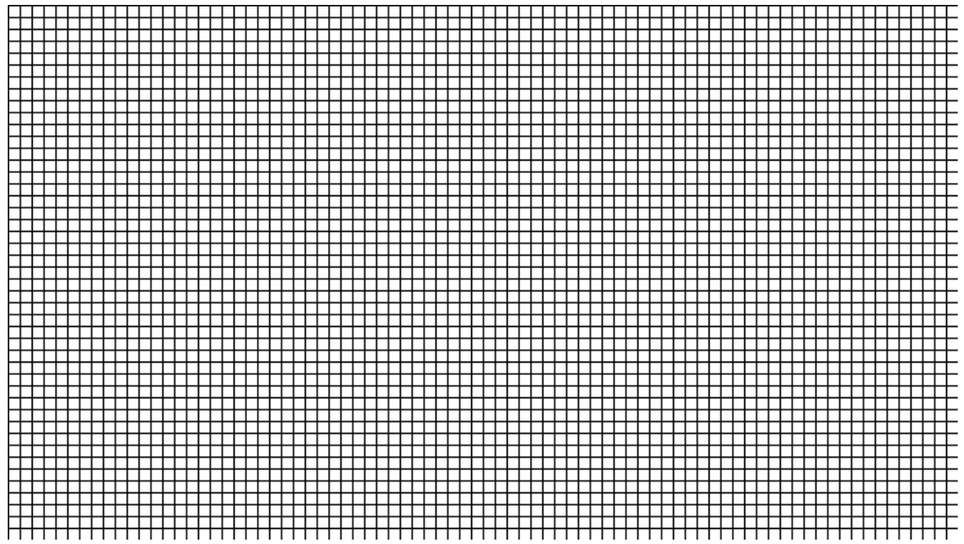
Example 2

```
import processing.video.*;
Capture video;
void setup() {
  size(320, 240);
  video = new Capture(this, width, height, 30);
  video.start();
}
void captureEvent(Capture video) {
  video.read();
}
void draw() {
  // Begin loop to walk through every pixel
  for (int x = 0; x < video.width; x ++ ) {
    for (int y = 0; y < video.height; y ++ ) {
      color current = video.get(x, y); // get its 'packed' color
      // break it into r,g,b
      float r = red(current);
      float g = green(current);
      float b = blue(current);
      // get the distance to the mouse
      float d = dist(x, y, mouseX, mouseY);
      // map it between 1 and 0
      float dx = map(d, 0, 100, 1, 0);
      set(x, y, color(r*dx, g*dx, b*dx)); // scale each color
    }
  }
}
```

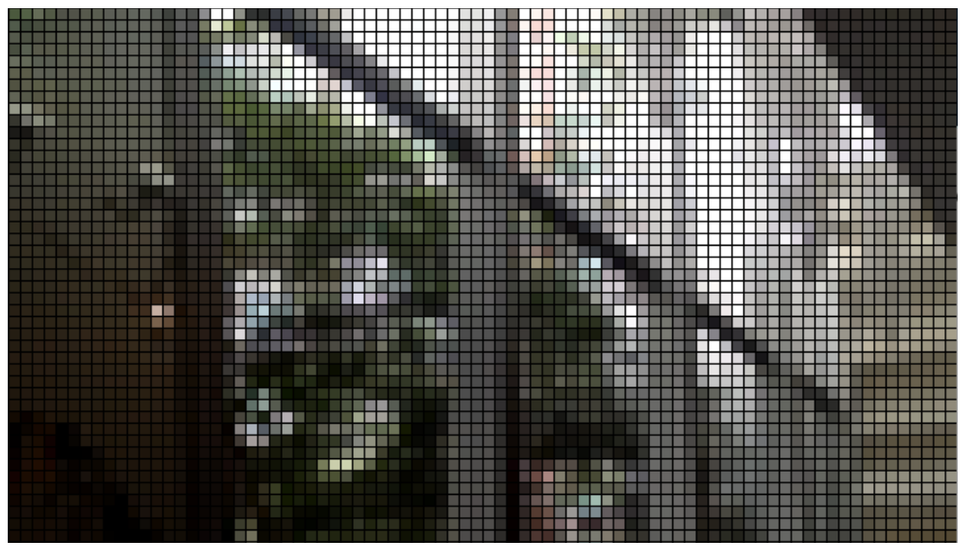
Similar to
Week 4 Example 2 &
Examples/Topics/Image
Processing/Brightness

Example 3 Video Pixelation

- Drawing a grid of 8 x 8 squares on a 640 x 480 window
- 640/8 = 80 columns
- 360/8 = 45 rows
- Create a Capture instant of 80 x 45 pixels (resolution)
- Then, fill each square with the pixel color captured from the camera
- (i, j) are coordinates in the camera's coordinate system
- (x, y) are coordinates in the sketch window's coordinate system



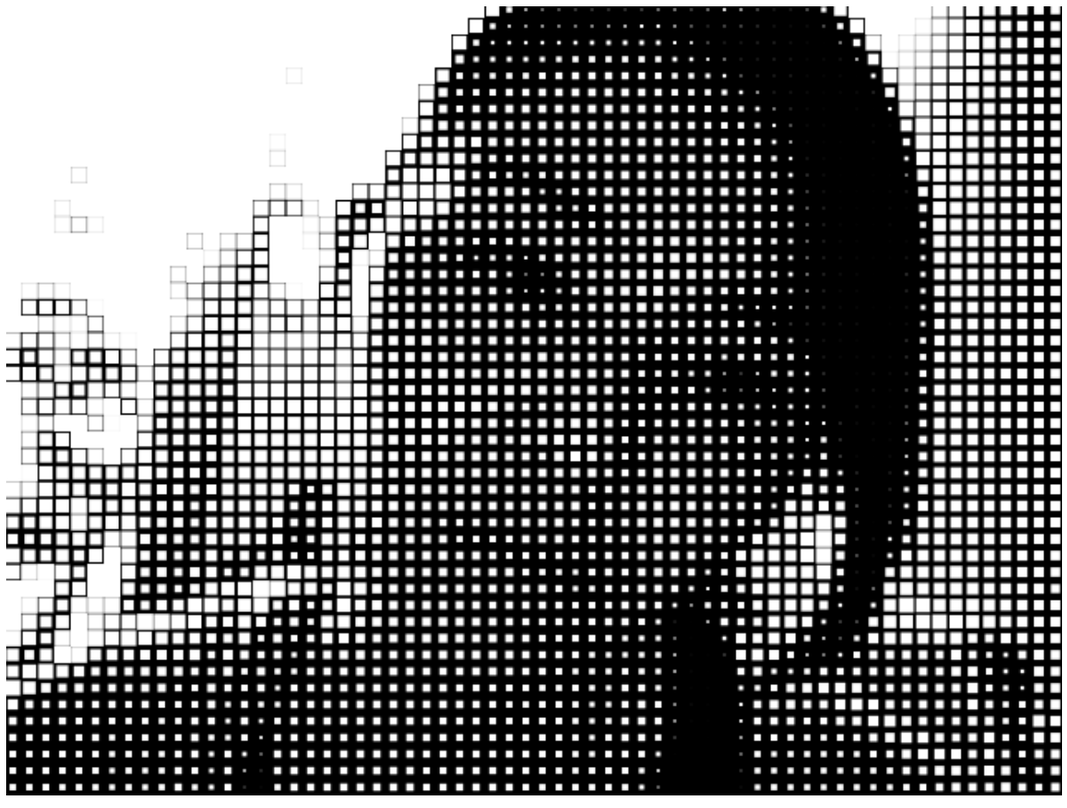
Example 3a



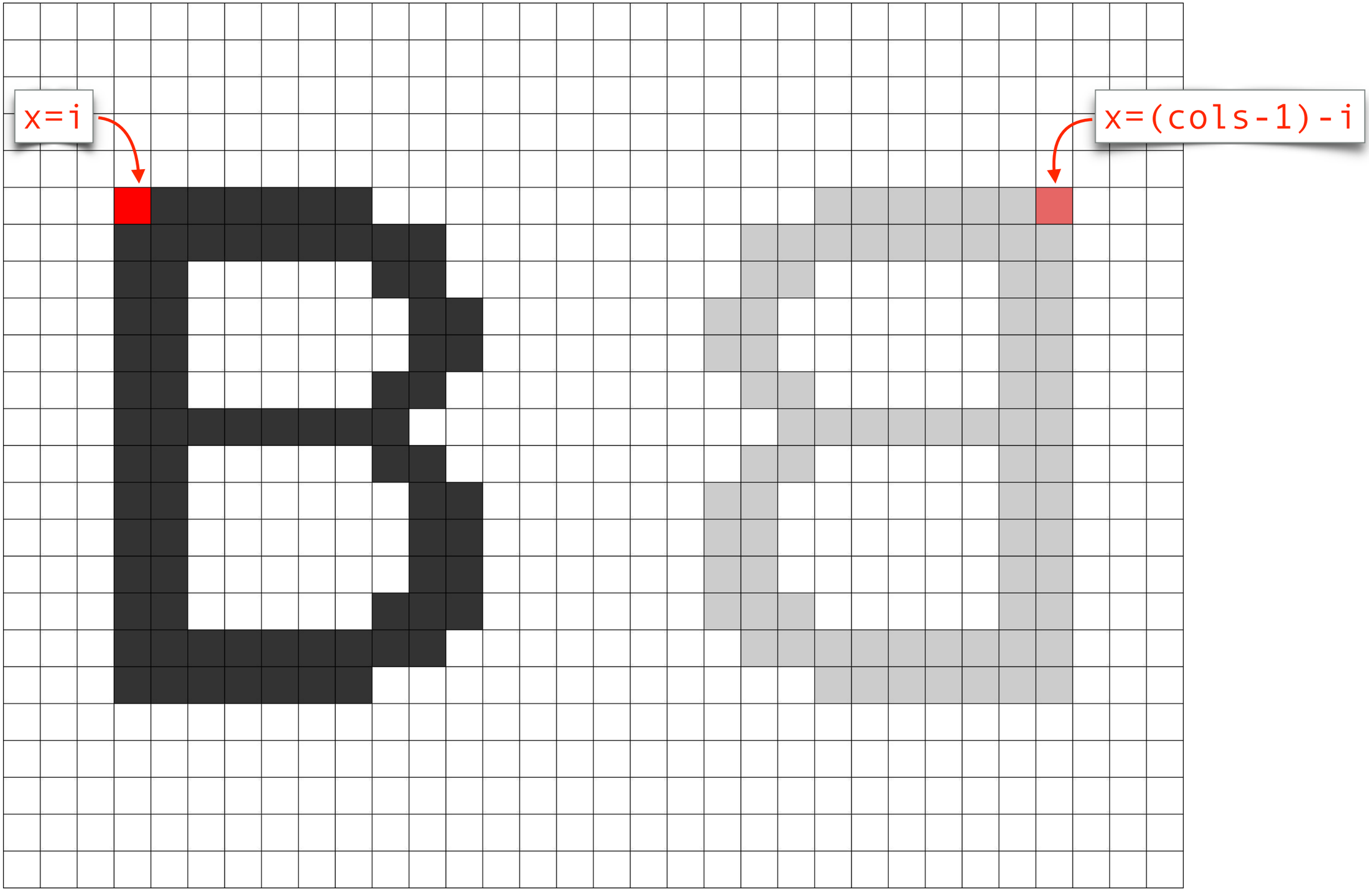
Example 3b

Example 4 Brightness Mirror

- Use only white fill color, but change the size of each square to reflect the brightness at the corresponding pixel
- Flip/mirror the image horizontally: (cols-1)-i

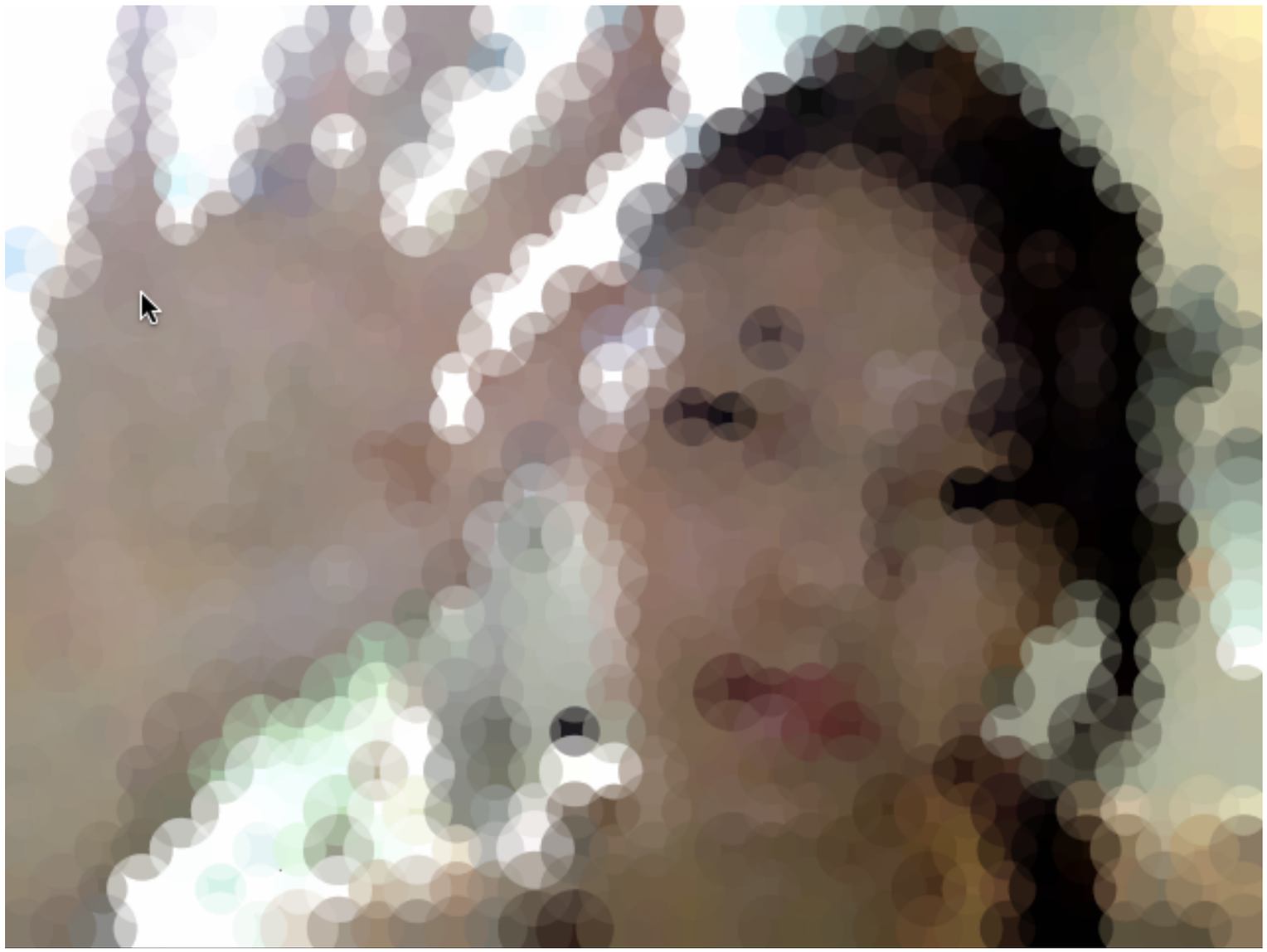


Horizontal Flipping (mirroring)



Example 5 Impressionistic Mirror

- Give each pixel color an alpha component



Example 5a

```
import processing.video.*;
int videoScale = 5;
int cols, rows;
Capture video;
void setup() {
  size(640, 640);
  cols = width / videoScale;
  rows = height / videoScale;
  video = new Capture(this, cols, rows);
  video.start();
  noStroke();
  background(0);
}

void captureEvent(Capture video) {
  video.read();
}

void draw() {
  randomSeed(0);
  for (int i = 0; i < cols; i++) {
    for (int j = 0; j < rows; j++) {
      int x = i*videoScale;
      int y = j*videoScale;
      color c = video.get(cols-i-1, j); // flip the i
      fill(c, 70); // add an alpha component
      float variation = random(5, 20);
      ellipse(x+videoScale/2, y+videoScale/2, videoSc
    }
  }
}
```

Example 5b

```
import processing.video.*;
int videoScale = 5;
int cols, rows;
Capture video;
void setup() {
  size(640, 640);
  cols = width / videoScale;
  rows = height / videoScale;
  video = new Capture(this, width, height); // full size
  video.start();
  noStroke();
  background(0);
}

void captureEvent(Capture video) {
  video.read();
}

void draw() {
  randomSeed(0);
  for (int i = 0; i < cols; i++) {
    for (int j = 0; j < rows; j++) {
      int x = i*videoScale;
      int y = j*videoScale;

      // use x, y instead of i, j to index the pixel coor
      color c = video.get(width-x-1, y); // flip the imag
      fill(c, 70); // add an alpha component
      float variation = random(5, 20);
      ellipse(x+videoScale/2, y+videoScale/2, videoScale+
    }
  }
}
```

Example 5a

Camera:
128 x 72

Example 5b

Camera:
640 x 360

Sketch window:
640 x 360

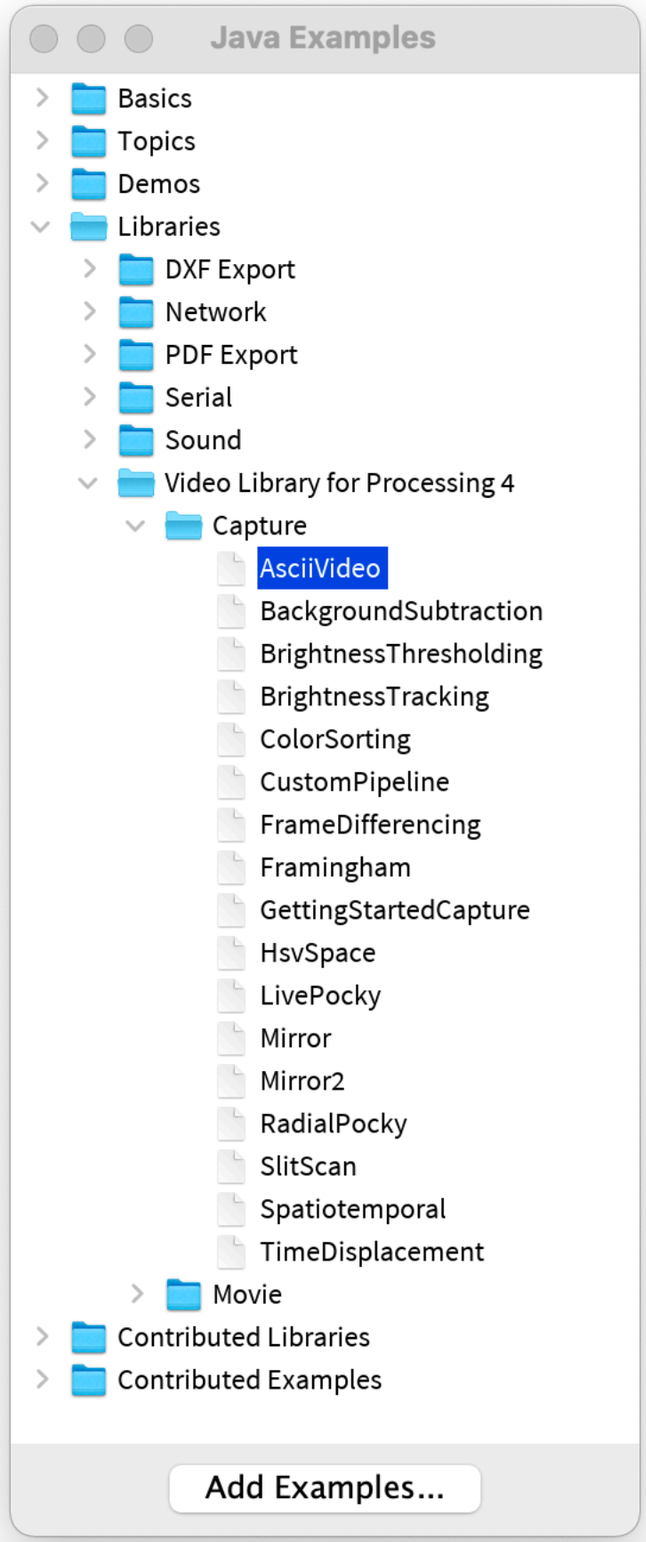
Exercise 1

- Recreate the Pointillism example in Week 1 (Processing official example Basics/Image/Pointillism) to work with live video
- Allow user to press any key to capture a portrait of his/herself

AsciiVideo Example

- Use different letters to represent color brightness

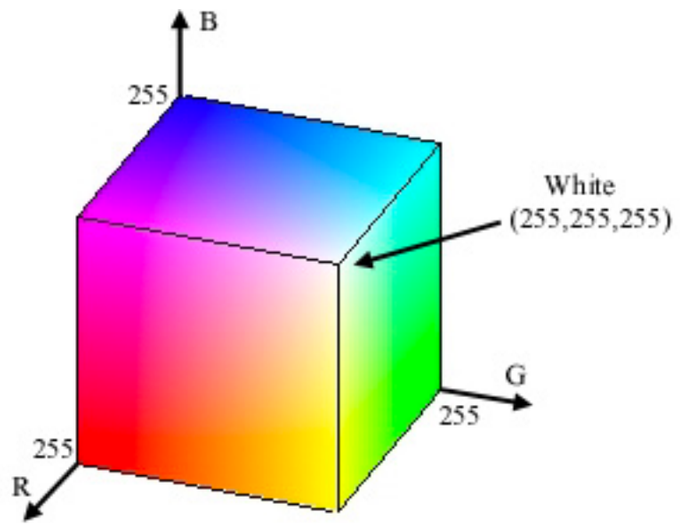
```
String letterOrder =
" .`-_':,;^=+/\"|)\\<>)iv%xclrs{*}I?!][1taeo7zjLu" +
"nT#JCwfy325Fp6mqSghVd4EgXPGZbYkOA&8U$@KHDBWNMR0Q";
```



Compare colors

- measure the “distance” between colors
- treat color as a point in three dimensional space
- if two colors are near each other in the color space, they are similar; if they are far, they are different.

```
float r1 = red(color1);
float g1 = green(color1);
float b1 = blue(color1);

float r2 = red(color2);
float g2 = green(color2);
float b2 = blue(color2);

// Using euclidean distance to compare colors
float d = dist(r1, g1, b1, r2, g2, b2);
```



Example 6a Color tracking

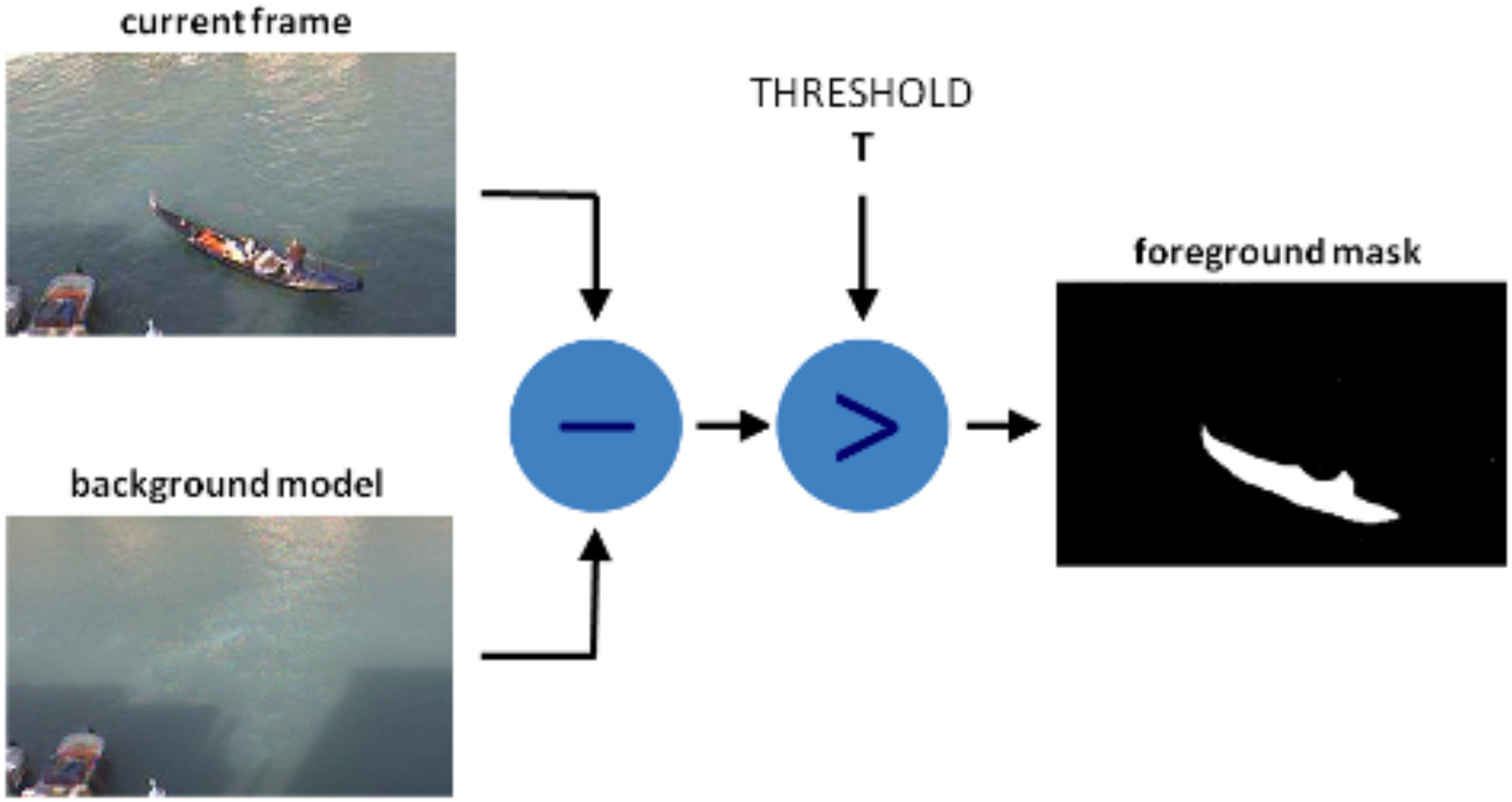
- Specify a color to be tracked
- Check every pixel in the current frame and find the pixel whose color is the most similar to the color being tracked
- i.e. find the pixel with the least color difference from the color being tracked
- Similar to a problem where you need to find the min or max value given an array of values

Example 6b Interpolation with lerp()


```
smoothedx = lerp(smoothedx, closestX, .1);
smoothedy = lerp(smoothedy, closestY, .1);
ellipse(smoothedx, smoothedy, 16, 16);
```

Motion detection

- Detect motion by comparing the current frame with the background frame



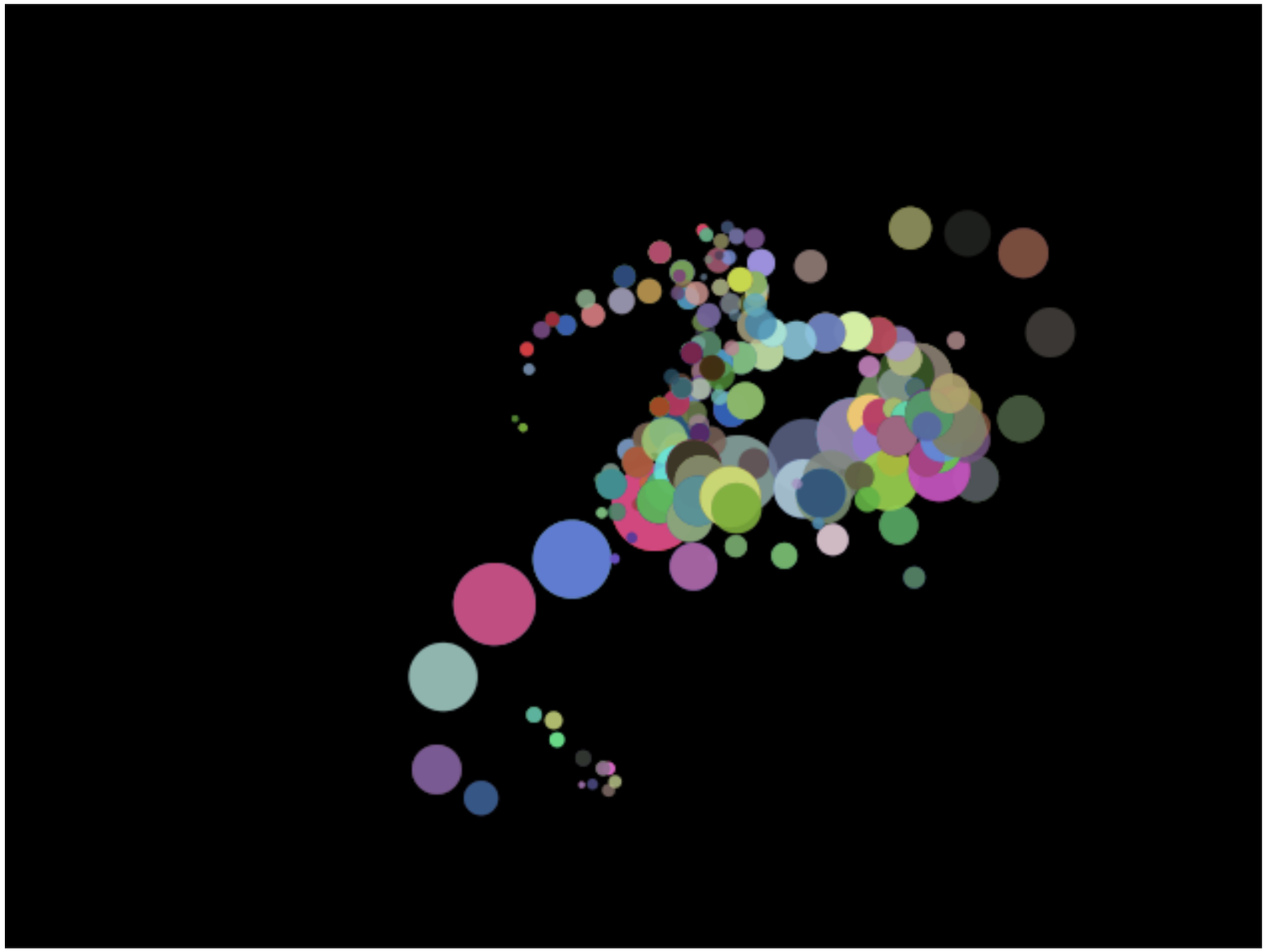
Example 7 Motion detection

Detect motion by comparing the current frame with the previous frame

Example 8a Average motion position

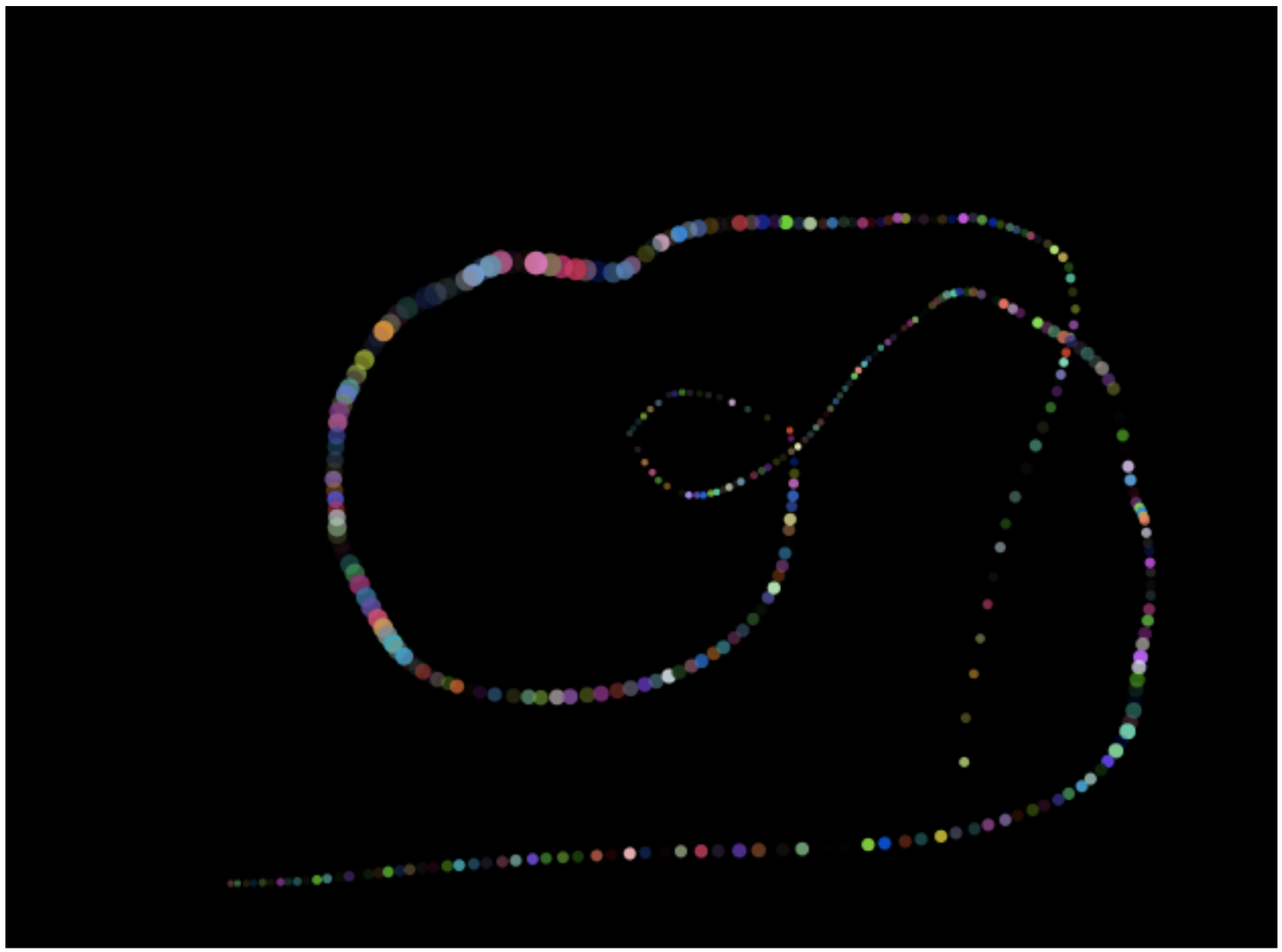
```
for (int x = 0; x < video.width; x++ ) {
  for (int y = 0; y < video.height; y++ ) {
    color current = video.pixels[x+y*video.width];
    color previous = prevFrame.pixels[x+y*video.width];
    // Compare colors (previous vs. current)
    float r1 = red(current);
    float g1 = green(current);
    float b1 = blue(current);
    float r2 = red(previous);
    float g2 = green(previous);
    float b2 = blue(previous);
    // Motion for an individual pixel is the difference
    // between the previous color and current color.
    float diff = dist(r1, g1, b1, r2, g2, b2);
    // If it's a motion pixel add up the x's and the y's
    if (diff > threshold) {
      sumX += x;
      sumY += y;
      motionCount++;
    }
  }
}

// average location is total location divided by the number of motion pixels.
float avgX = sumX / motionCount;
float avgY = sumY / motionCount;
```

Example 8b Motion as paintbrush



Exercise 2

- Smoothen the path of the ellipse in Example 8b
- Flip the video or the tracked point horizontally (mirror)



Exercise 3

- Flip both the live video and tracked point horizontally in Example 6b

Exercise 4

- Enclose the motion area using a rectangle