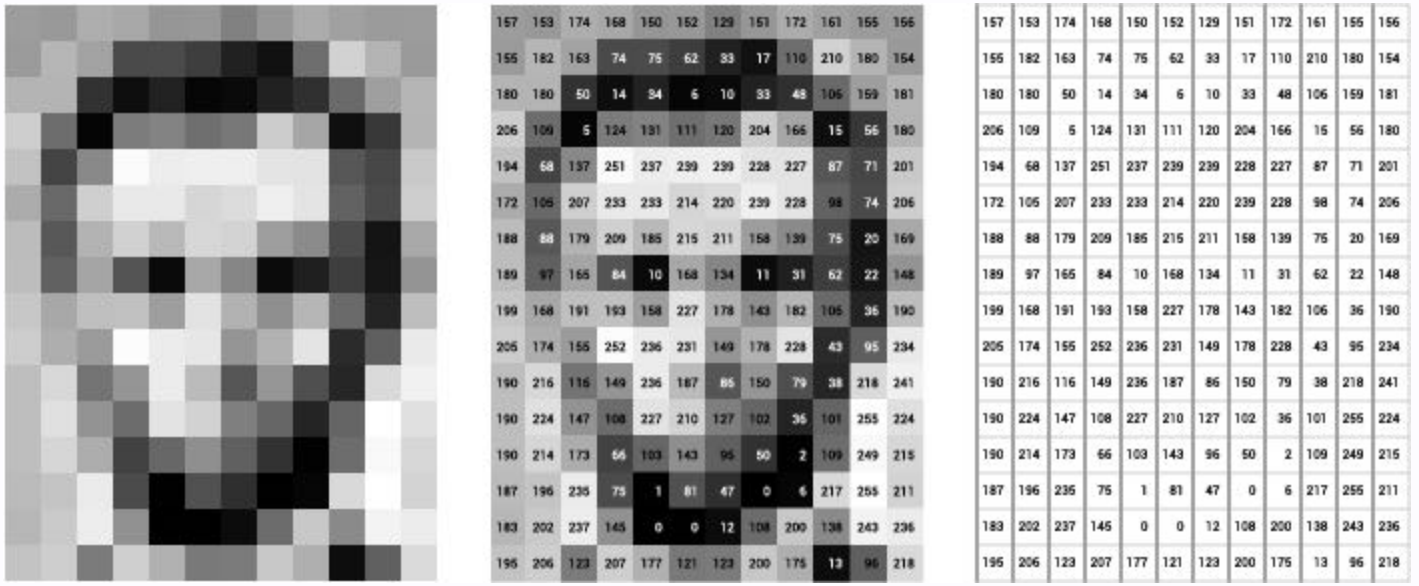
Processing & Media

Content

- PImage (Pixel Manipulation)
- Video / Webcam Input
- Filters

Grayscale Image



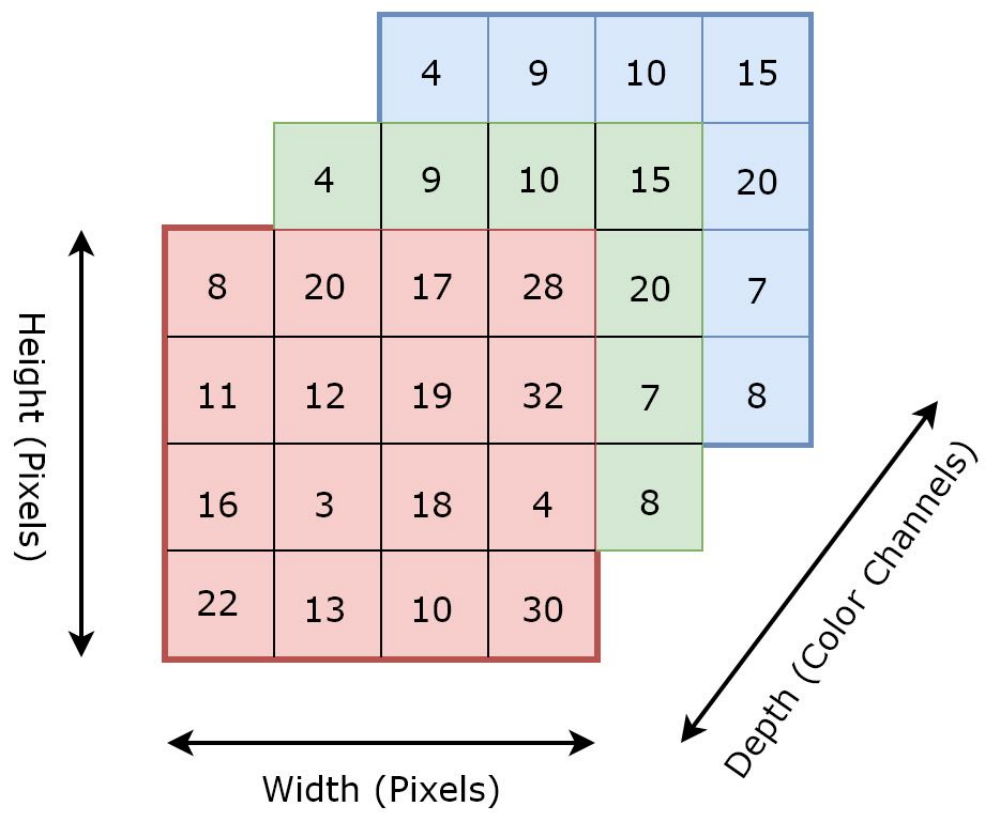
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

- `width * height = pixels`

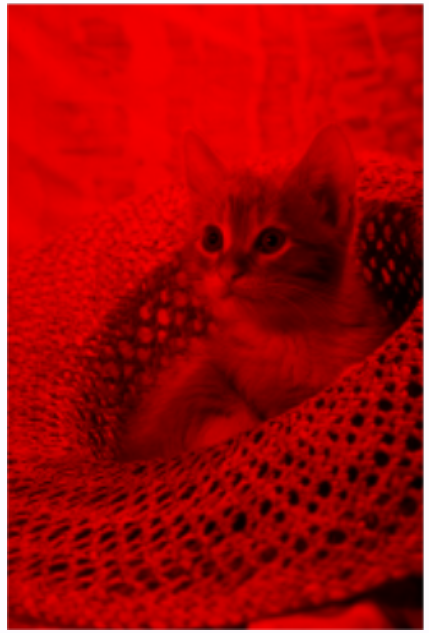
Source: sleebapaul

Multi-Channel Image



Source: OReilly

RGB Image




PImage

```
// create new image
PImage img = new PImage(300, 200, RGB);

// access size
int w = img.width;
int h = img.height;

// access raw pixels
int[] pixels = img.pixels;
```

Load PImage

```
PImage img = loadImage("cat.png");
```

- Loads from the `data` folder ⚡
- URL's are allowed!

Source: University of Wisconsin–Madison

Display PImage

```
// display
image(img, 0, 0);

// display scaled
image(img, 0, 0, 500, 500);
```

Image Mode

`CORNER`, `CORNERS`, `CENTER`

Example:

```
// draw centered image
imageMode(CENTER);
image(img, width / 2, height / 2);
```

Resize PImage

```
// resize
img.resize(800, 600);

// resize and keep ratio
img.resize(0, 50);
```

Accessing Pixels

At location `x`, `y`

```
// read
color pixel = img.get(x, y);

// write
color myColor = color(200, 10, 20)
img.set(x, y, myColor);
```

Color

```
color pixel = color(200, 100, 100);
```

- color
  - red / green / blue
  - hue / saturation / brightness

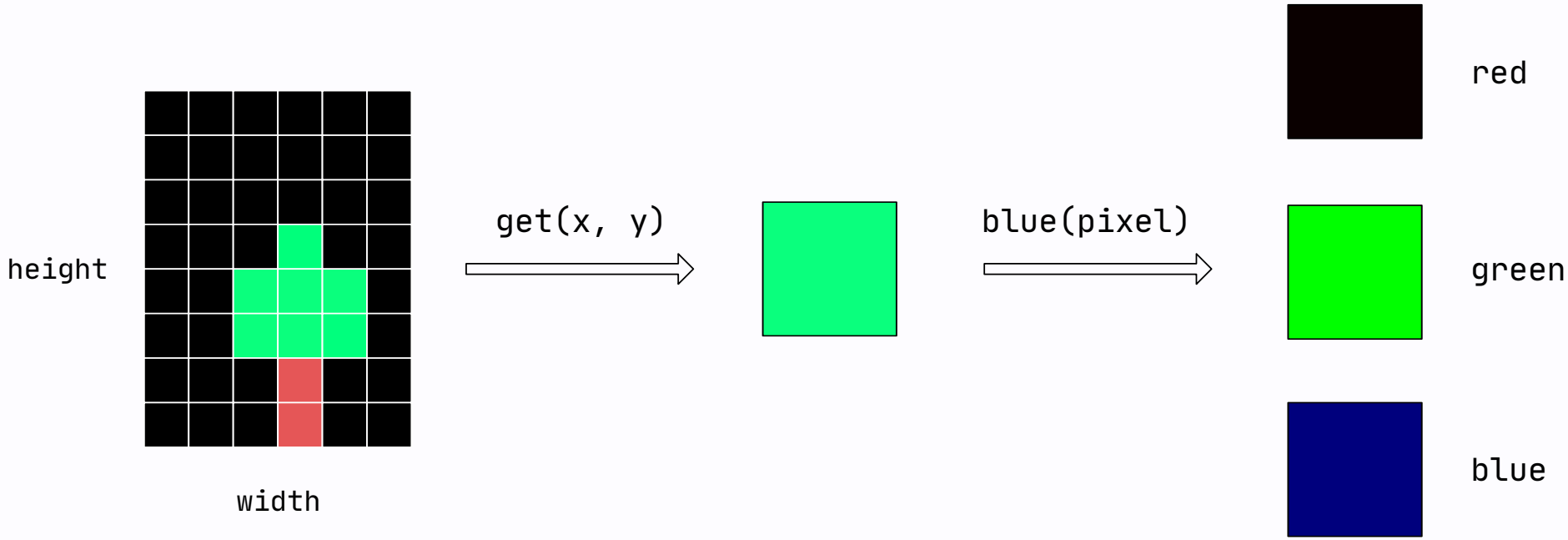
Accessing Color Channels

```
color pixel = img.get(x, y);

int r = red(pixel);
int g = green(pixel);
int b = blue(pixel);

int h = hue(pixel);
int s = saturation(pixel);
int b = brightness(pixel);
```

PImage

Color / Pixel

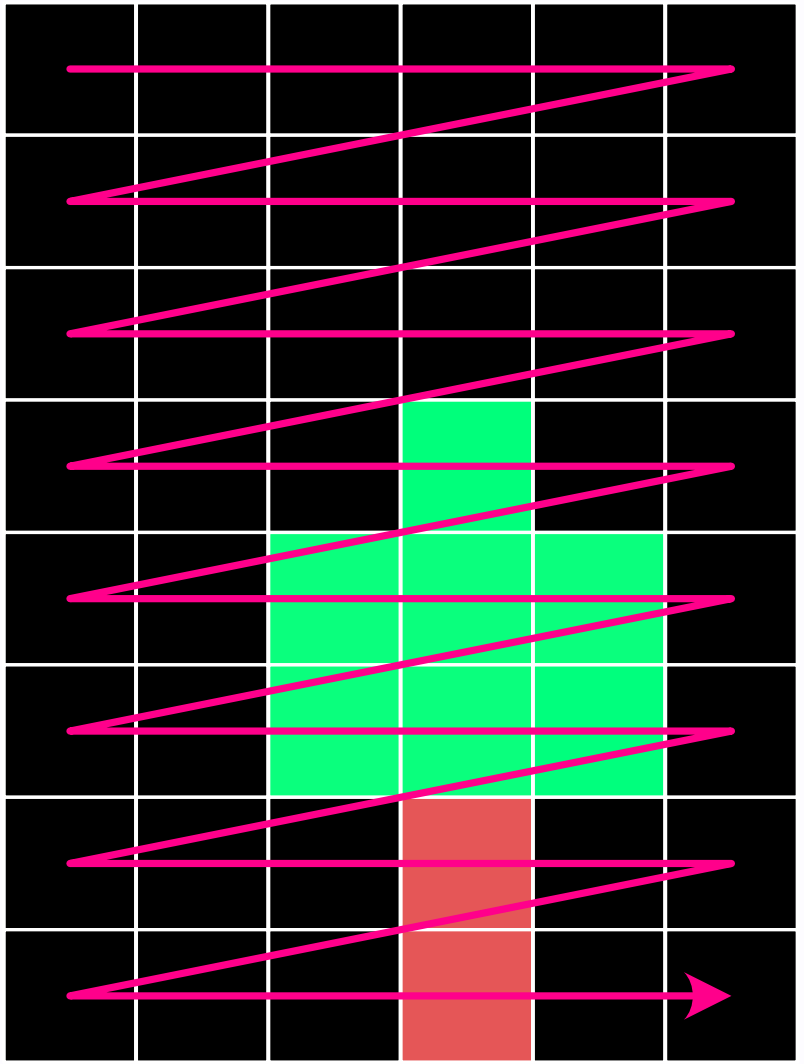
Channels

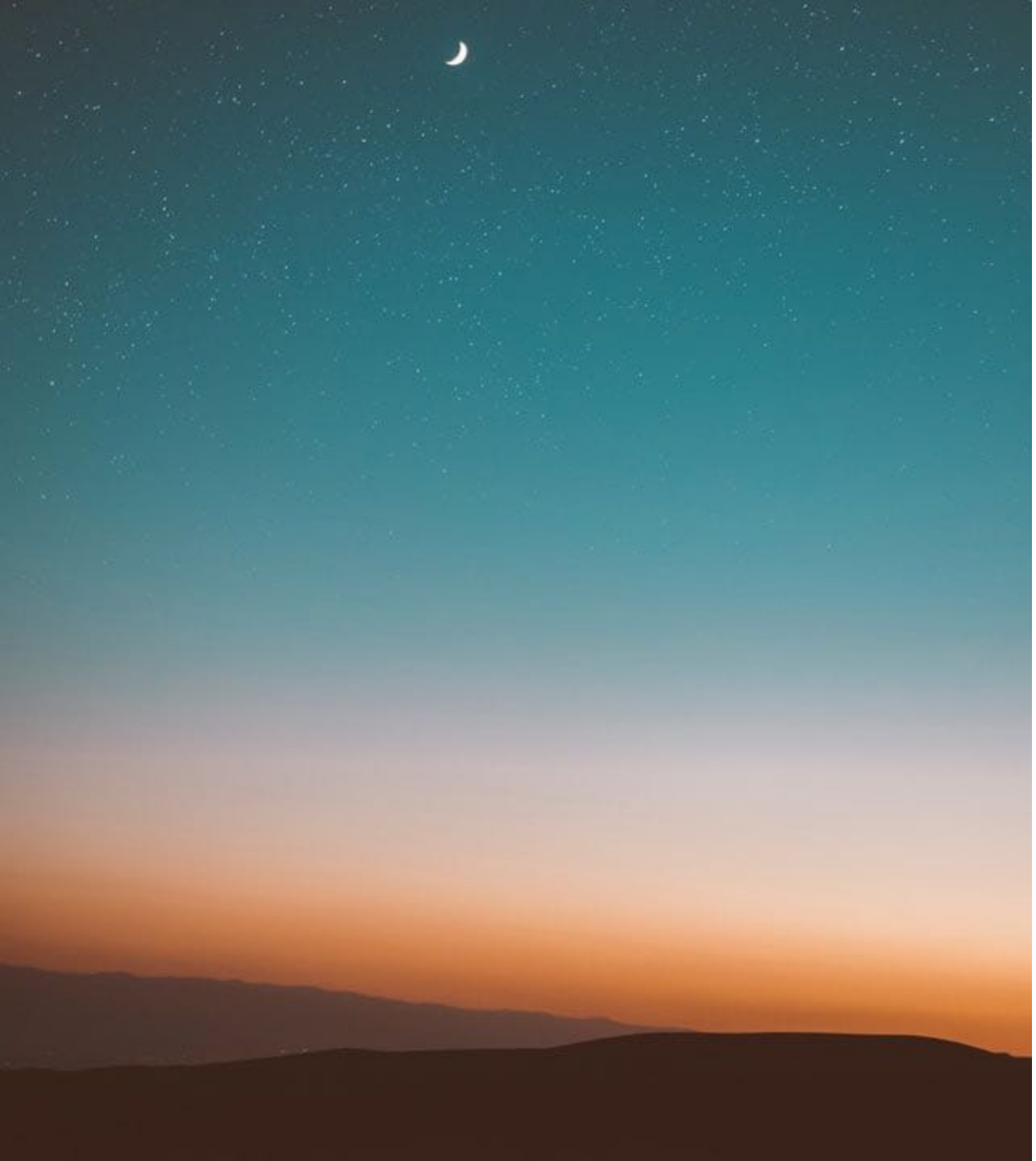


Pixel Iteration

Looping over each pixel value.

```
int w = img.width;
int h = img.height;

for(int y = 0; y < h; y++) {
  for(int x = 0; x < w; x++) {

    // do something
    img.set(x, y, color(0)):
  }
}
```





Task 1 (20min)

Create a new processing sketch which loads this image and displays it.

- How can we find the position of the moon?

Source: Mohammad Alizade

Task 1 - Hint

- Extract the position of the brightest pixel from the image.
- `brightness(...)`

Task 1 - Solution

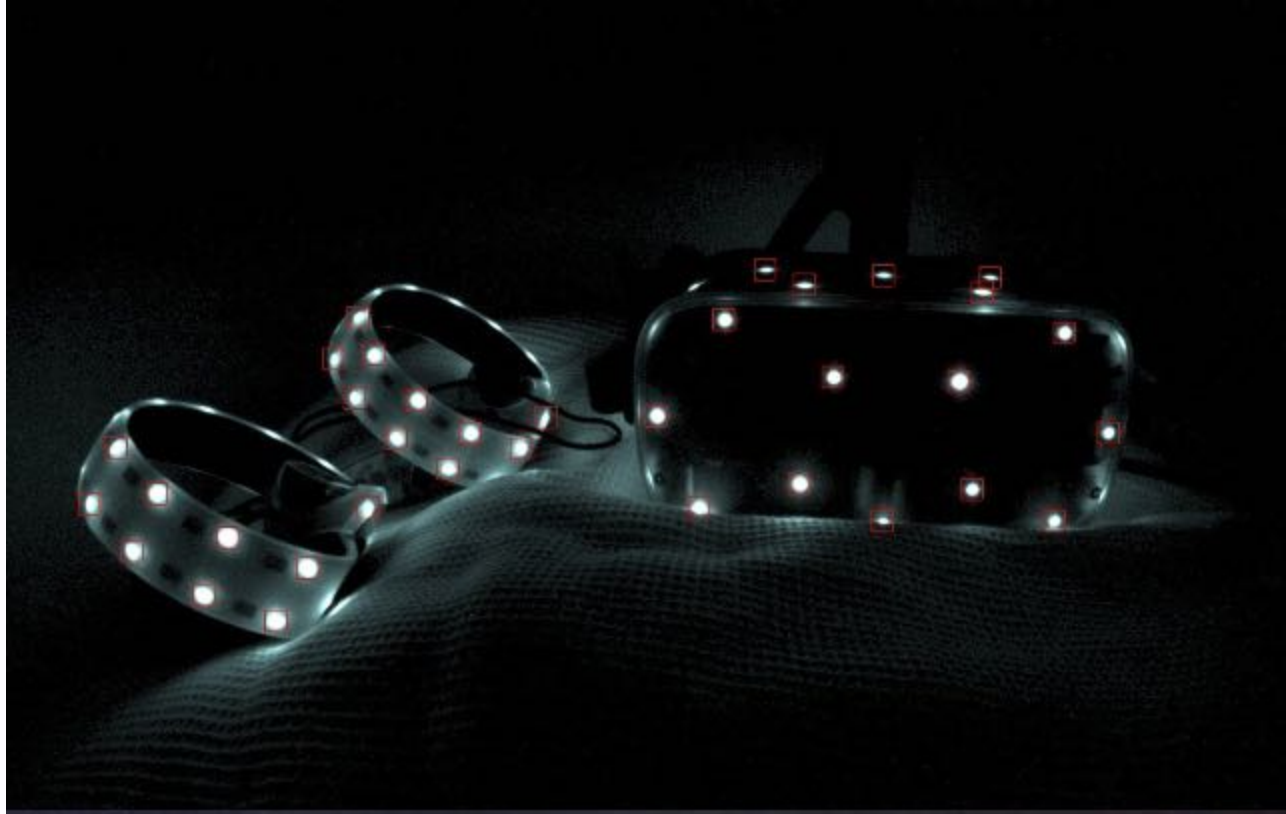
- Init max brightess = `0`
- Init max x & y position
- Loop over each pixel
  - Extract the brightness
  - Check if it is brightest pixel
    - Store position and brightness
- Mark brightest location with circle

```
// find brightest spot
  int maxX = 0;
  int maxY = 0;
  float maxBrightness = 0;

  // loop over all pixels
  for (int y = 0; y < img.height; y++) {
    for (int x = 0; x < img.width; x++) {

      // extract pixel and brightness
      color pixel = img.get(x, y);
      float b = brightness(pixel);

      if (b > maxBrightness) {
        maxBrightness = b;
        maxX = x;
        maxY = y;
      }
    }
  }
```

Why?

- light = attention
- Infrared (IR) Tracking
  - Most contrast
  - No distraction

Source: mechatech.co.uk



Input

Process

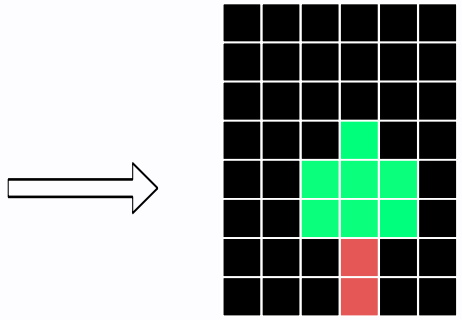
Output

file

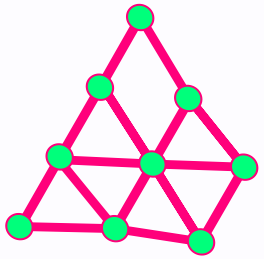
video

camera

special



image

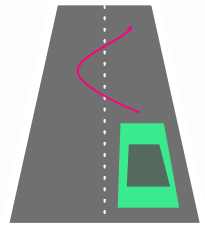


algorithm

image

101010
011001
010101

data



action

Interaction = Live Input

- Processing Video
- Movie and Webcam support
- How to install? Video 2.0

List all cameras

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

```
String[] cameras = Capture.list();
println(cameras);
```

Start a capture session

```
Capture cam;

void setup() {
  cam = new Capture(this, 640, 480, cameras[0]);
  cam.start();
}
```

- Could not work anymore! (MacOS) ⚠️

Start a capture session (reliable)

```
Capture cam;

void setup() {
  cam = new Capture(this, "pipeline:autovideosrc");
  cam.start();
}
```

Read the video and display

```
void draw() {

  // read frame
  if(cam.available()) {
    cam.read();
  }

  // display
  image(cam, 0, 0);
}
```

- `Capture` is a `PImage`

Task 2 (15min)

Install the Video Library 2.0 from the contribution manager and display the webcam image!

```
import processing.video.*;

Capture cam;

void setup() {
  cam = new Capture(this, "pipeline:autovideosrc");
  cam.start();
}

void draw {
  // read frame
  if(cam.available()) {
    cam.read();
  }

  image(cam, 0, 0);
}
```

Magic Wand

Control the brightest spot in the image.

Source: Ethan Hoover)

Task 3 - Tracking & Webcam (30min)

- Combine with brightest point tracking
- Create an own paint application

Task 3 - Hint

How to draw something that stays on the screen?

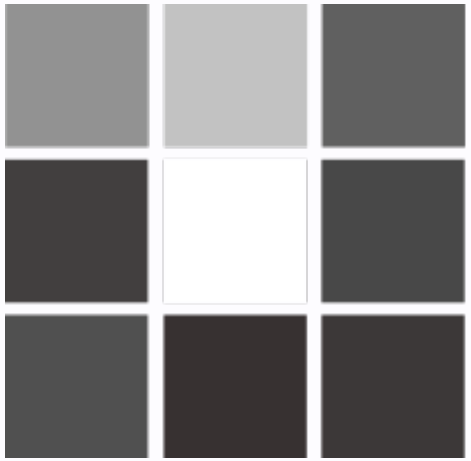
- Do not clear the `background(...)` every frame but just in `setup()`!

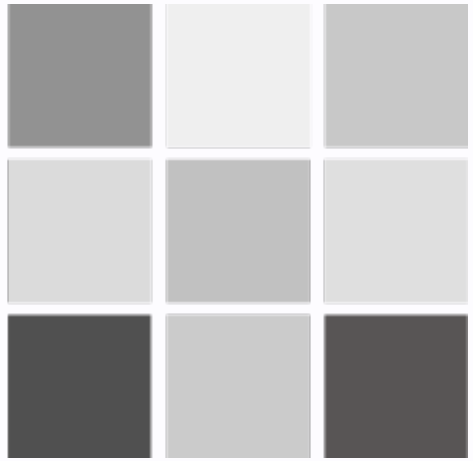
Task 3 - Skeleton

```
void setup() {
  // setup camera

  // clear background once
}

void draw() {
  // read camera image
  // extract brightest point

  // if point is brighter then n
  // draw circle at this point
}
```

Task 3 - Help

Finding a good brightness value is quite difficult.





- Webcams use Auto-Exposure
- Remove noise

Processing Image Filter (Convolution)

- Runs image processing operation
- Alreay implemented
- Quite slow

Builtin Filters

```
img.filter(THRESHOLD);
img.filter(GRAY);
img.filter(INVERT);
img.filter(POSTERIZE, 4);
img.filter(BLUR, 6);
img.filter(ERODE);
img.filter(DILATE);
```

Demo Drawing Tool