Lab 5: Color Sensing

Tuesday 2/10 2:15pm

Jp Marhefka, Aveed Gorji, & Gwyn Anawalt

Part 1

white:

Color Temp: 3743 K - Lux: 303 - R: 455 G: 416 B: 281 C: 1174

Color Temp: 3735 K - Lux: 309 - R: 468 G: 426 B: 288 C: 1203

Color Temp: 3728 K - Lux: 315 - R: 476 G: 433 B: 292 C: 1222

Color Temp: 3737 K - Lux: 309 - R: 466 G: 425 B: 287 C: 1201

Color Temp: 3744 K - Lux: 308 - R: 463 G: 423 B: 286 C: 1193

Color Temp: 3728 K - Lux: 312 - R: 471 G: 429 B: 289 C: 1210

Color Temp: 3731 K - Lux: 313 - R: 472 G: 430 B: 290 C: 1215

Color Temp: 3734 K - Lux: 313 - R: 473 G: 431 B: 291 C: 1217

Color Temp: 3731 K - Lux: 313 - R: 472 G: 430 B: 290 C: 1214
Color Temp: 3736 K - Lux: 311 - R: 471 G: 429 B: 290 C: 1212

Average White Paper:

White Paper Noise Levels

R Avg.:

G Avg:

B Avg:

Black:
Color Temp: 3758 K - Lux: 61 - R: 103 G: 90 B: 64 C: 258
Color Temp: 3871 K - Lux: 68 - R: 106 G: 97 B: 69 C: 273
Color Temp: 3897 K - Lux: 75 - R: 114 G: 106 B: 75 C: 296
Color Temp: 3870 K - Lux: 82 - R: 126 G: 116 B: 82 C: 326
Color Temp: 3879 K - Lux: 81 - R: 124 G: 115 B: 81 C: 322
Color Temp: 3858 K - Lux: 80 - R: 122 G: 113 B: 79 C: 316
Color Temp: 3869 K - Lux: 79 - R: 123 G: 113 B: 80 C: 318
Color Temp: 3899 K - Lux: 78 - R: 120 G: 111 B: 79 C: 312
Color Temp: 3878 K - Lux: 79 - R: 121 G: 112 B: 79 C: 314
Color Temp: 3889 K - Lux: 80 - R: 122 G: 113 B: 80 C: 317

Average Black Paper:

C Avg:

R Avg.:

G Avg:

B Avg:

Red, Green, and Blue paper from Test Sample Set A Observations:

Red: R high ~350, B and G low ~100-150

Blue: all in ~100

Green: r & g ~250, B ~150

Measurement time 24ms and measurement gain 16X:

```
public:
 Adafruit_TCS34725(uint8_t = TCS34725_INTEGRATIONTIME_24MS,
                   tcs34725Gain_t = TCS34725_GAIN_16X);
```

Part 2

Blue Data:

0	4
1	6
2	8
3	13
4	16
5	19
6	25
7	29
8	32
9	35
10	40
11	44
12	45
13	48
14	50
15	55
16	57
17	60
18	60
19	62
20	65
21	66
22	68
23	70
24	72
25	73
26	74
27	77
28	78
29	78
30	79
31	80
32	80
33	81
34	81
35	82
36	83
37	84
38	85
39	85
40	85
41	86
42	86
43	87
44	87
45	87
46	87
47	88
48	89
49	88
50	89
51	89
52	89
53	89
54	89
55	90
56	90
57	90
58	90
59	90
60	90
61	91
62	90
63	90
64	91
65	91
66	91
67	91
68	91
69	91
70	91
71	91
72	92
73	91
74	91
75	91
76	91
77	91
78	91
79	91
80	91
81	91
82	92
83	92
84	91
85	92
86	91
87	92
88	92
89	91
90	92
91	91
92	92
93	92
94	92
95	92
96	92
97	92
98	92
99	92

Max value = 92

90% of 92 = 82.8

99% of 92 = 91.08

Blue delay 90%: ((0.9 * (maxvalue)) [i]) *10 = 360

Blue delay 99%: ((0.9 9* (maxvalue)) [i]) *10 = 640

Red Data:

0	4
1	5
2	9
3	12
4	14
5	17
6	22
7	26
8	28
9	31
10	36
11	38
12	41
13	43
14	46
15	48
16	53
17	54
18	56
19	58
20	59
21	60
22	61
23	64
24	65
25	66
26	68
27	69
28	71
29	71
30	72
31	72
32	74
33	74
34	75
35	75
36	76
37	77
38	77
39	78
40	78
41	79

42	79
43	80
44	80
45	80
46	80
47	81
48	81
49	81
50	81
51	82
52	82
53	82
54	82
55	82
56	82
57	82
58	83
59	83
60	83
61	83
62	83
63	83
64	84
65	83
66	83
67	83
68	83
69	83
70	84
71	84
72	84
73	84
74	84
75	84
76	84
77	84
78	84
79	84
80	84
81	84
82	84
83	84

84	84
85	84
86	84
87	84
88	85
89	84
90	84
91	84
92	84
93	84
94	84
95	84
96	84
97	84
98	84
99	84

Max value = 84

90% of 84 = 75.6

99% of 84 = 83.16

Red delay 90%: ((0.9 * (maxvalue)) [i]) *10 = 350

Red delay 99%: ((0.9 9* (maxvalue)) [i]) *10 = 670

Green Data:

0	38
1	47
2	67
3	95
4	113
5	126
6	141
7	153
8	174
9	184
10	195
11	205
12	222
13	228
14	235
15	248
16	259
17	260
18	266
19	269
20	278
21	280
22	286
23	290
24	295
25	298
26	301
27	303
28	310
29	311
30	313
31	315
32	317
33	319
34	320
35	321
36	323
37	324
38	326
39	327
40	328

41	329
42	329
43	330
44	331
45	332
46	332
47	333
48	334
49	334
50	334
51	335
52	335
53	336
54	336
55	336
56	337
57	337
58	337
59	337
60	338
61	337
62	338
63	338
64	338
65	338
66	338
67	339
68	339
69	338
70	339
71	339
72	339
73	339
74	339
75	339
76	339
77	339
78	339
79	339
80	339
81	339

82	339
83	339
84	339
85	339
86	339
87	339
88	339
89	339
90	339
91	339
92	339
93	340
94	339
95	340
96	340
97	339
98	339
99	340

Max value = 340

90% of 340 = 306

99% of 340 = 334

Green delay 90%: ((0.9 * (maxvalue)) [i]) *10 = 275

Green delay 99%: ((0.9 9* (maxvalue)) [i]) *10 = 490

100ms program:

```
#define RED 44    // 90% delay 350 us
#define BLUE 46   // 90% delay 360 us
#define GREEN 42  // 90% delay 275 us
#define PHOTOIN A0


int i;
int r_reading, b_reading, g_reading;


void setup() {

  Serial.begin(9600);
  pinMode(RED, OUTPUT);
  pinMode(BLUE, OUTPUT);
  pinMode(GREEN, OUTPUT);
  digitalWrite(RED, LOW);
  digitalWrite(BLUE, LOW);
  digitalWrite(GREEN, LOW);
  analogReference(DEFAULT);
  pinMode(PHOTOIN, INPUT);

}


void loop() {

  digitalWrite(RED, HIGH);
  delayMicroseconds(350);
  for (i = 0; i < 4; i++) {
    r_reading += analogRead(PHOTOIN);
  }
  digitalWrite(RED, LOW);
  delay(1);


  digitalWrite(BLUE, HIGH);
```

```
  delayMicroseconds(360);
  for (i = 0; i < 4; i++) {
    b_reading += analogRead(PHOTOIN);
  }
  digitalWrite(BLUE, LOW);
  delay(1);


  digitalWrite(GREEN, HIGH);
  delayMicroseconds(275);
  for (i = 0; i < 4; i++) {
    g_reading += analogRead(PHOTOIN);
  }
  digitalWrite(GREEN, LOW);


  Serial.print("r = ");
  Serial.println(r_reading);
  Serial.print("b = ");
  Serial.println(b_reading);
  Serial.print("g = ");
  Serial.println(g_reading);

  delayMicroseconds(97015);

  r_reading = b_reading = g_reading = 0;
}
```

	Red paper	Green paper	Blue paper	White paper	Black paper
averages	R 245 G 692 B 176	R 106.22 G 855.89 B 156.12	R 115 G 699 B 205	R 277.92 G 1119.00 B 324.08	R 83 G 565 B 115
ranges	R 239 to 250 G 687 to 696 B 170 to 179	R 103 to 109 G 854 to 858 B 154 to 159	R 108 to 120 G 695 to 701 B 198 to 209	R 271 to 283 G 120 to 1217 B 320 to 329	R 80 to 86 G 561 to 572 B 110 to 118

Discrimination function:

```
#define RED 44
#define BLUE 46
#define GREEN 42


#define RED_D 49
#define BLUE_D 51
#define GREEN_D 47
#define WHITE_D 45


#define PHOTOIN A0


int i;
long r_reading, g_reading, b_reading;


const float targetR[] = {70.00,  290.00, 115.08, 106.22, 330.00}; // Black Red Blue
Green White
const float targetG[] = {400.00, 720.00, 785.00, 855.89, 1500.00};
const float targetB[] = {100.00, 175.00, 235.00, 156.12, 500.00};
const char* colorNames[] = {"Black", "Red", "Blue", "Green", "White"};


void setup() {
 Serial.begin(9600);
 pinMode(RED_D, OUTPUT);
 pinMode(BLUE_D, OUTPUT);
 pinMode(GREEN_D, OUTPUT);
 pinMode(WHITE_D, OUTPUT);
 pinMode(RED, OUTPUT);
 pinMode(BLUE, OUTPUT);
 pinMode(GREEN, OUTPUT);
 analogReference(DEFAULT);
}


void loop() {
 r_reading = g_reading = b_reading = 0;
```

```
digitalWrite(RED, HIGH);
delayMicroseconds(350);
for (i = 0; i < 4; i++) {
  r_reading += analogRead(PHOTOIN);
}
digitalWrite(RED, LOW);
delay(1);


digitalWrite(BLUE, HIGH);
delayMicroseconds(360);
for (i = 0; i < 4; i++) {
  b_reading += analogRead(PHOTOIN);
}
digitalWrite(BLUE, LOW);
delay(1);


digitalWrite(GREEN, HIGH);
delayMicroseconds(275);
for (i = 0; i < 4; i++) {
  g_reading += analogRead(PHOTOIN);
}
digitalWrite(GREEN, LOW);


float r = r_reading;
float g = g_reading;
float b = b_reading;
 Serial.println(r);
Serial.println(g);
Serial.println(b);


int closestColor = 0;
float minDistance = 999999;


for (int j = 0; j < 5; j++) { //go through each color
```

```
   float distance = sqrt(pow(r - targetR[j], 2) + pow(g - targetG[j], 2) + pow(b -
targetB[j], 2)); // distance formula
   if (distance < minDistance) { //if it beats our previous minimum distance
     minDistance = distance; //update recordholder distance
     closestColor = j; //update recordholder color
   }
 }




 Serial.print("detected: ");
 Serial.println(colorNames[closestColor]);


 digitalWrite(RED_D, LOW);
 digitalWrite(BLUE_D, LOW);
 digitalWrite(GREEN_D, LOW);
 digitalWrite(WHITE_D, LOW);
 if (closestColor == 1) {
   digitalWrite(RED_D, HIGH);
 }
 else if (closestColor == 2) {
   digitalWrite(BLUE_D, HIGH);
 }
 else if (closestColor == 3) {
   digitalWrite(GREEN_D, HIGH);
 }
 else if (closestColor == 4) {
   digitalWrite(WHITE_D, HIGH);
 }


 delay(200);
}
```

Problems We Encountered:

There was a logical error writing the code for step 5 in experiment 2, the 100ms program. In the beginning we thought they would all run at the same time, but then we realized that the red and blue and green operations happen in sequence. This was easily fixed by not implementing everything in one for loop and entering the 90% values we measured previously.

In our determinant function, we faced an implementation issue where we were not using the color space technique. Once we implemented this technique, and shifted some of our color average values around to match a general color space, our color sensing was accurate.