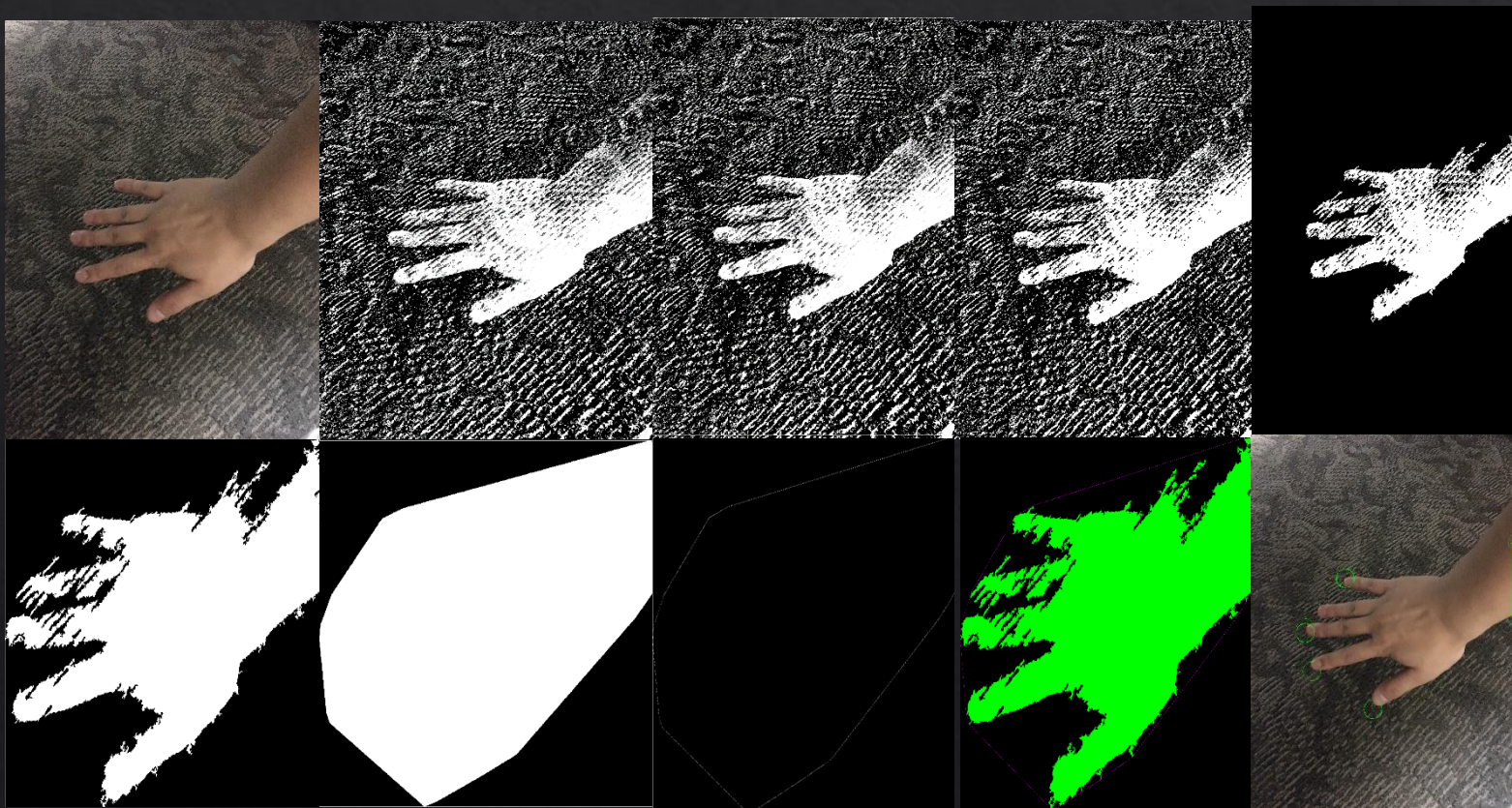


Finger Detection



Sungwon In
Seungmin Baek

Agenda

1. Motivation

2. Background Subtraction

- 1) Subtracting Image

- 2) Filtering

- 3) Converting Image type

- 4) Find connected component

3. Convex Hull

4. Contouring

Agenda

1. Motivation

2. Background Subtraction

- 1) Subtracting Image
- 2) Filtering
- 3) Converting Image type
- 4) Find connected component

3. Convex Hull

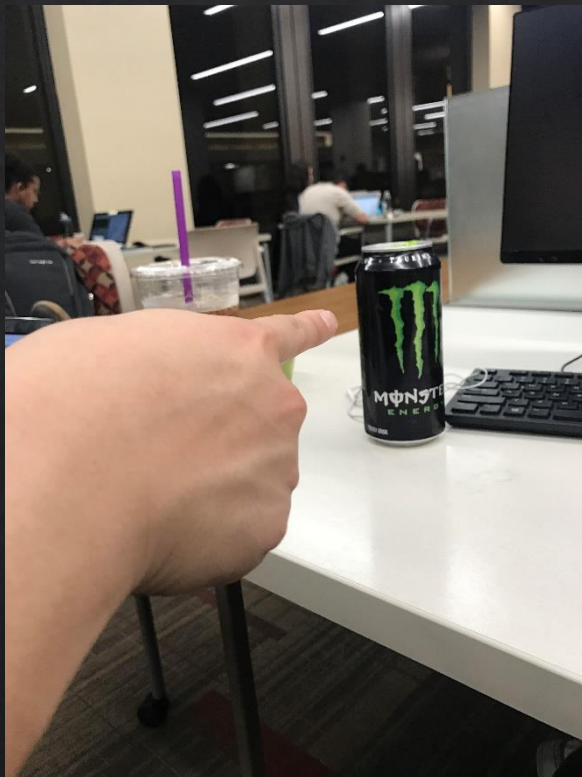
4. Contouring

Motivation

Detecting finger in image



Increase quality of machine learning



Agenda

1. Motivation

2. Background Subtraction

1) Subtracting Image

2) Filtering

3) Converting Image type

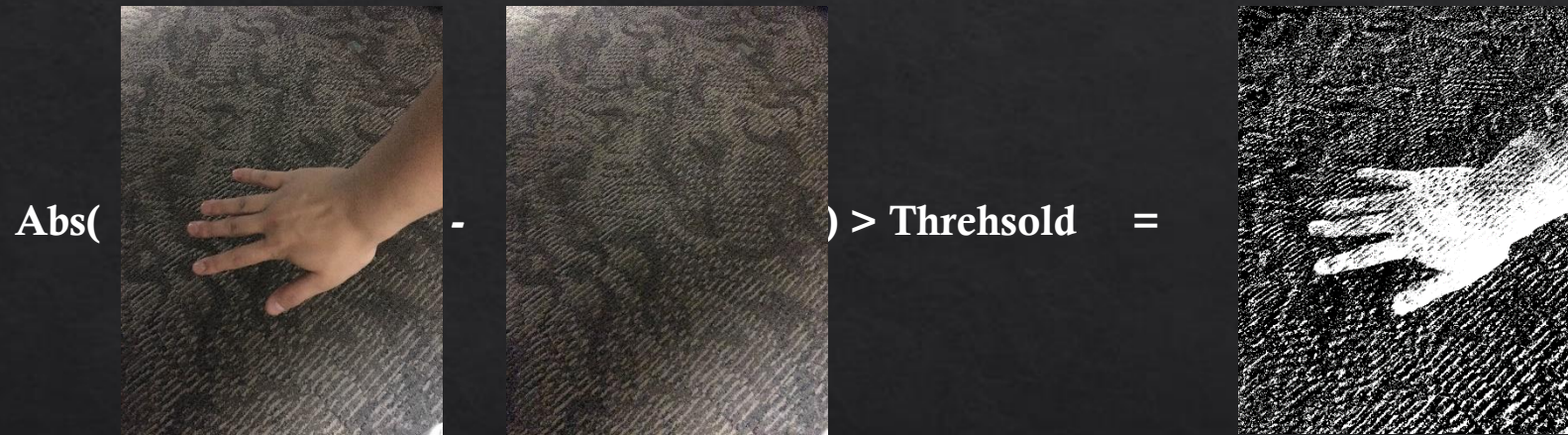
4) Find connected component

3. Convex Hull

4. Contouring

Subtracting Image

Formula: $\text{abs}(\text{current_frame} - \text{previous_frame}) > \text{threshold} = \text{subtracted_image}$



Subtracting Image



20



40



50



100

Agenda

1. Motivation

2. Background Subtraction

1) Subtracting Image

2) Filtering

3) Converting Image type

4) Find connected component

3. Convex Hull

4. Contouring

Subtracting Image

Main function: `medfilt2(Image)`



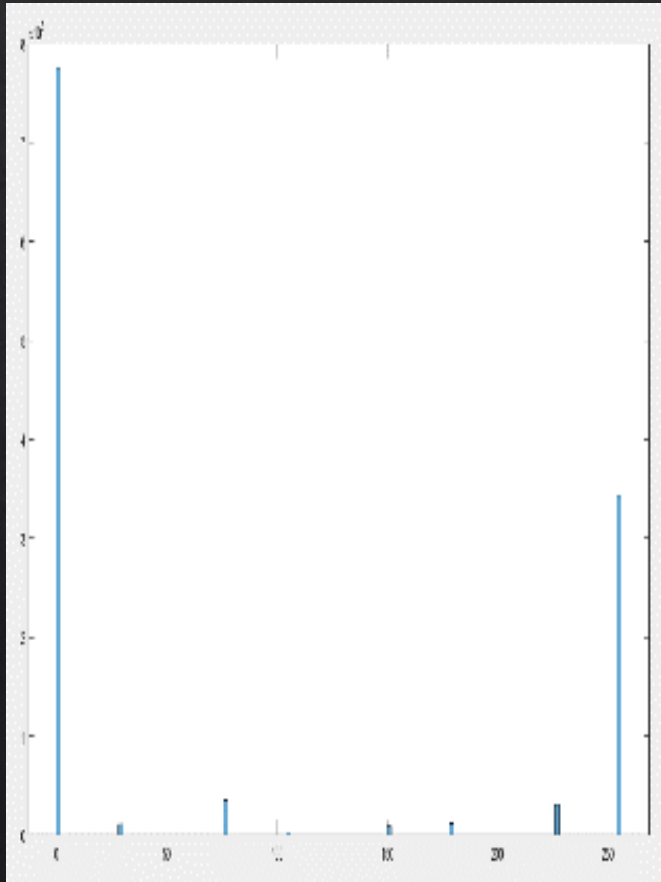
Subtracting Image

Main function: `imgaussfilt(Image)`

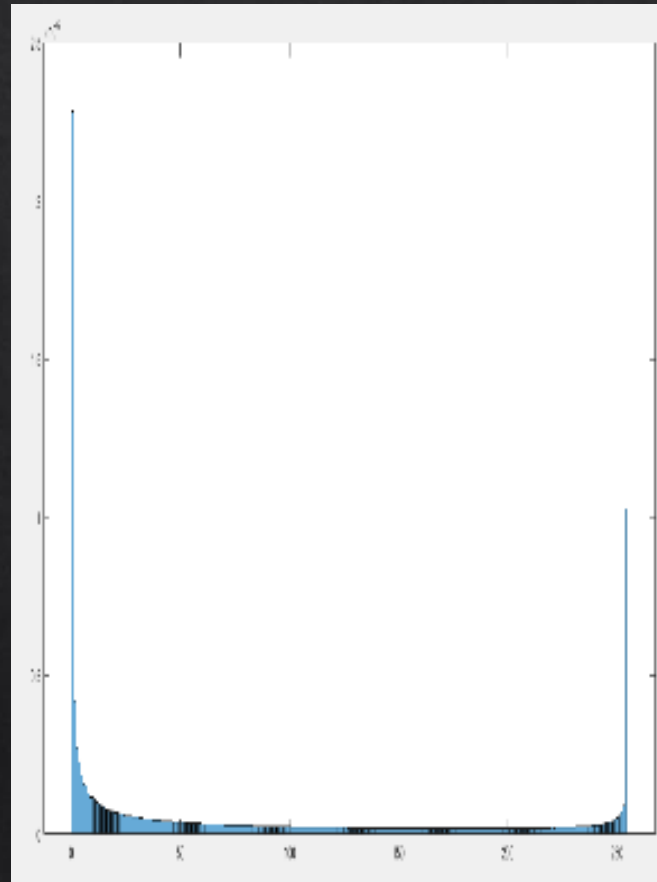


Decide which filtering algorithm

Median Filtered Histogram



Gaussian Filtered Histogram



Agenda

1. Motivation

2. Background Subtraction

1) Subtracting Image

2) Filtering

3) Converting Image type

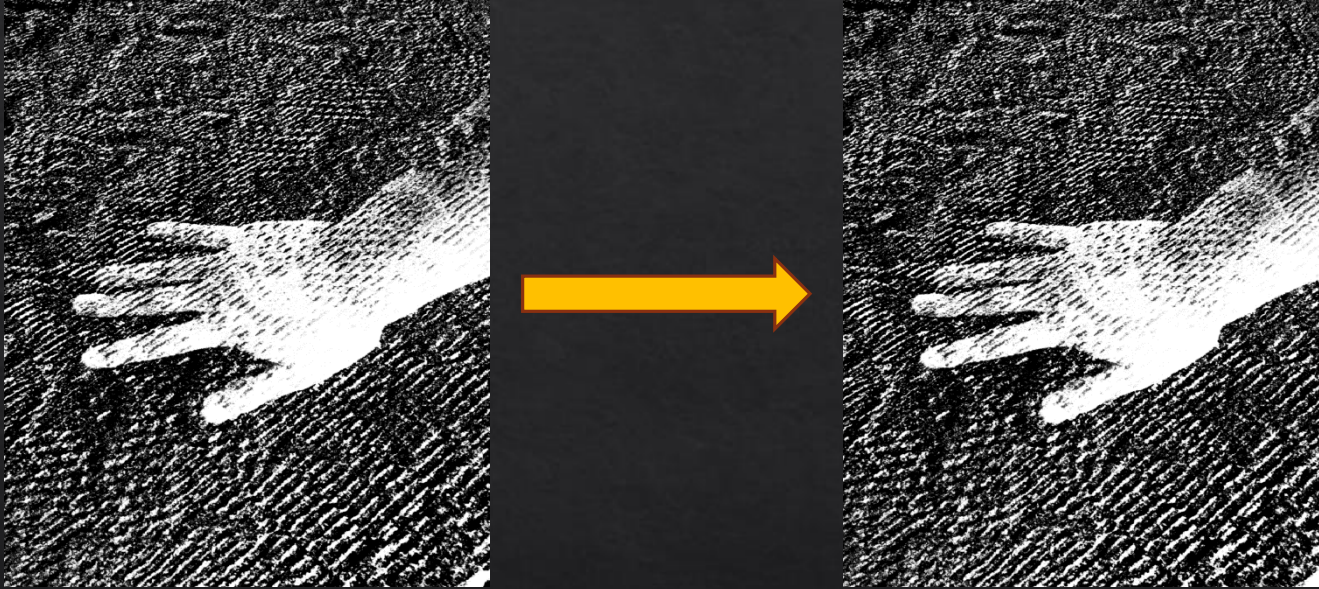
4) Find connected component

3. Convex Hull

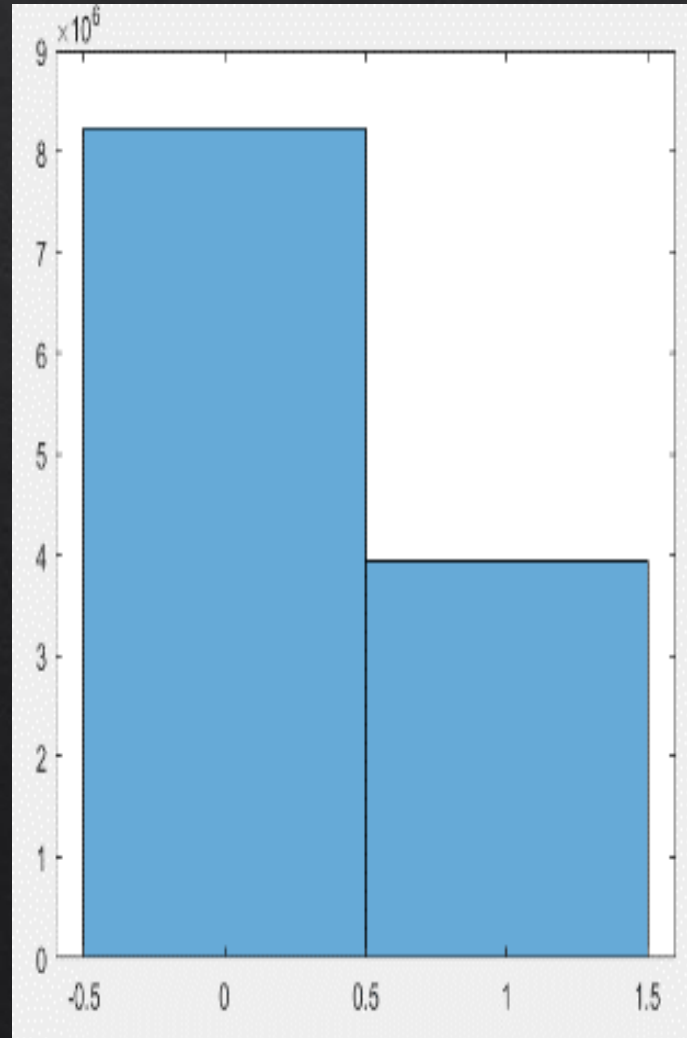
4. Contouring

Converting Image Type

Main function: `im2bw(Image)`



Converting Image Type



Converting Image Type

3024x4032 uint8												
	1	2	3	4	5	6	7	8	9	10	11	12
1	10	6	4	5	12	28	53	80	101	108	97	72
2	24	16	9	6	9	20	41	65	87	99	95	76
3	46	30	17	9	8	14	31	53	75	91	93	79
4	71	47	26	13	8	12	26	46	67	85	90	80
5	91	60	33	17	10	13	26	44	63	79	85	77
6	99	66	36	19	12	16	29	44	59	71	74	66
7	91	60	33	18	13	17	29	42	52	59	59	52
8	71	47	26	14	11	15	25	34	40	43	42	36
9	46	30	17	9	8	11	18	24	27	28	27	25
10	24	16	9	5	4	6	10	14	16	17	18	22
11	10	6	4	2	2	3	4	6	8	10	17	27
12	2	2	1	1	1	1	2	2	5	11	23	40
13	1	2	2	3	2	2	1	2	5	15	33	57
14	3	6	8	9	8	6	3	3	7	20	44	74
15	8	13	18	20	18	13	7	5	9	24	51	84
16	15	25	34	37	33	24	14	7	10	24	51	83
17	23	39	52	58	52	38	23	11	10	21	44	73
18	31	52	71	79	72	54	33	16	10	16	33	56
19	37	65	89	100	93	71	45	23	11	11	21	39
20	43	74	104	118	112	89	59	31	15	9	13	24
21	45	79	112	130	125	102	69	38	18	10	9	14
22	43	78	110	130	127	104	73	42	22	12	8	9
23	39	69	99	117	115	95	68	41	24	14	9	7



3024x4032 logical												
	169	170	171	172	173	174	175	176	177	178	179	180
49	0	0	0	0	0	0	0	0	0	1	0	0
50	0	0	0	0	0	0	0	0	1	1	0	0
51	0	0	0	0	0	0	0	0	1	1	1	0
52	0	0	0	0	0	0	0	0	0	1	0	0
53	0	0	0	0	0	0	0	0	0	0	0	0
54	0	0	0	0	0	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0	0	0	0
56	0	0	0	0	0	0	0	0	0	0	0	0
57	0	0	0	0	0	0	0	0	0	0	0	0
58	0	0	0	0	0	0	0	0	0	0	0	0
59	0	0	0	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0	0	0	0
61	0	0	0	0	0	0	0	0	0	0	0	0
62	1	0	0	0	0	0	0	0	0	0	0	0
63	1	1	0	0	0	0	0	0	0	0	0	0
64	1	1	1	1	0	0	0	0	0	0	0	0
65	0	1	1	1	1	1	1	1	1	1	1	1
66	0	0	1	1	1	1	1	1	1	1	1	1
67	0	0	0	1	1	1	1	1	1	1	1	1
68	0	0	0	1	1	1	1	1	1	1	1	1
69	0	0	0	0	1	1	1	1	1	1	1	1
70	0	0	0	0	1	1	1	1	1	1	1	1
71	0	0	0	0	0	0	1	1	1	1	1	1

Agenda

1. Motivation

2. Background Subtraction

1) Subtracting Image

2) Filtering

3) Converting Image type

4) Find connected component

3. Convex Hull

4. Contouring

Find connected component

Main function: `bwareaopen(connected_component, threshold)`



* Find most connected component

Agenda

1. Motivation

2. Background Subtraction

1) Subtracting Image

2) Filtering

3) Converting Image type

4) Find connected component

3. Convex Hull

4. Contouring

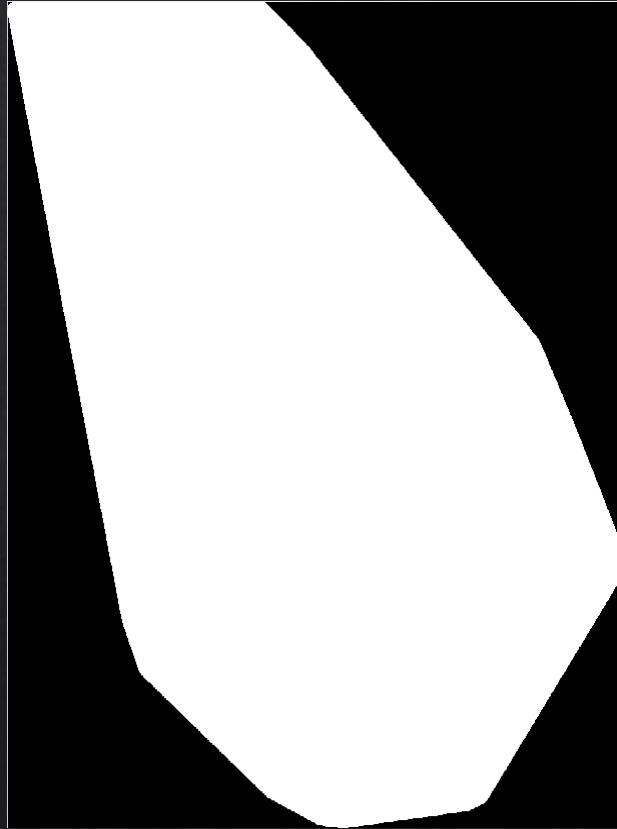
Convex Hull

Algorithm used: Graham's Scan

1. Find the smallest y-coordinate in the image. And let's say this y-coordinate 'pivot'
2. Except pivot, sort all the pixels respect to ascending order of x-coordinate and save it to an array. And let's say that array is called 'P'
3. Push pivot and P(1) into stack. And let's say stack is called 'S'
4. Iterate through index 2 to number of elements in 'P'. And repeat the step 5 and 6.
5. Check if 'S' has more than 2 elements or not. And if 'S' has more than 2 elements, pop the element P(i) until P(i) can be pushed into 'S'
6. Push P(i) into 'S'

Convex Hull

* Main function: `Regionprops()`



Agenda

1. Motivation

2. Background Subtraction

1) Subtracting Image

2) Filtering

3) Converting Image type

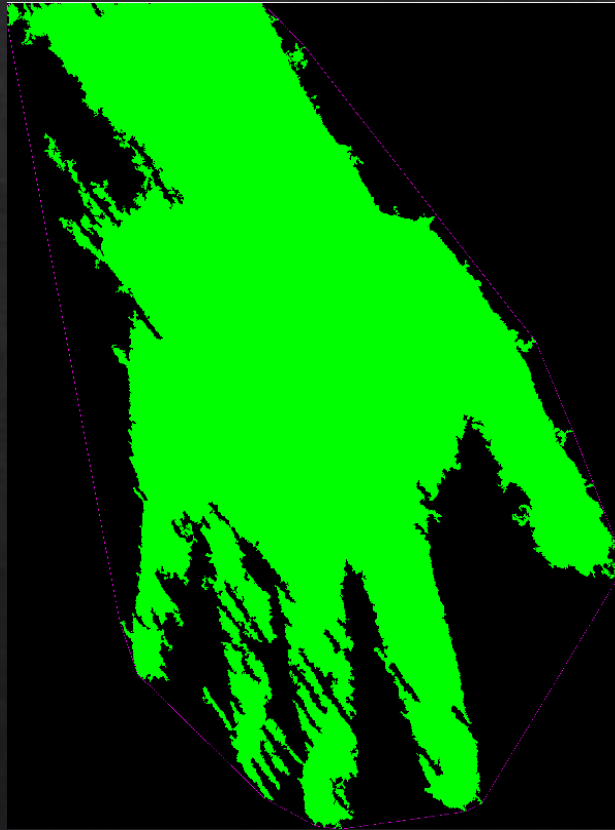
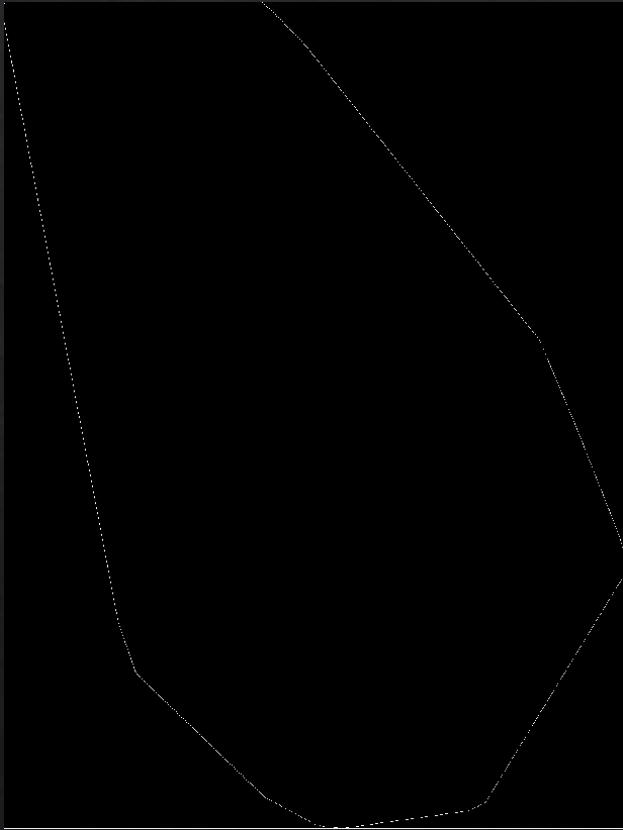
4) Find connected component

3. Convex Hull

4. Contouring

Contouring

- Edge algorithm: Canny
- Main function: `imfuse(Image)`

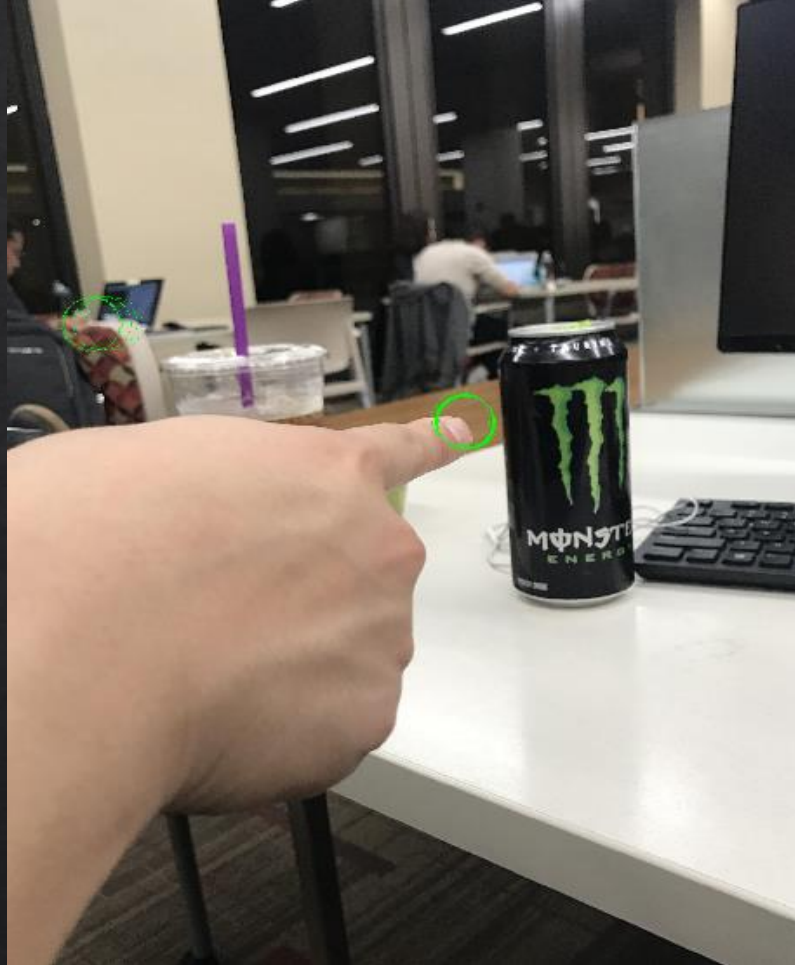
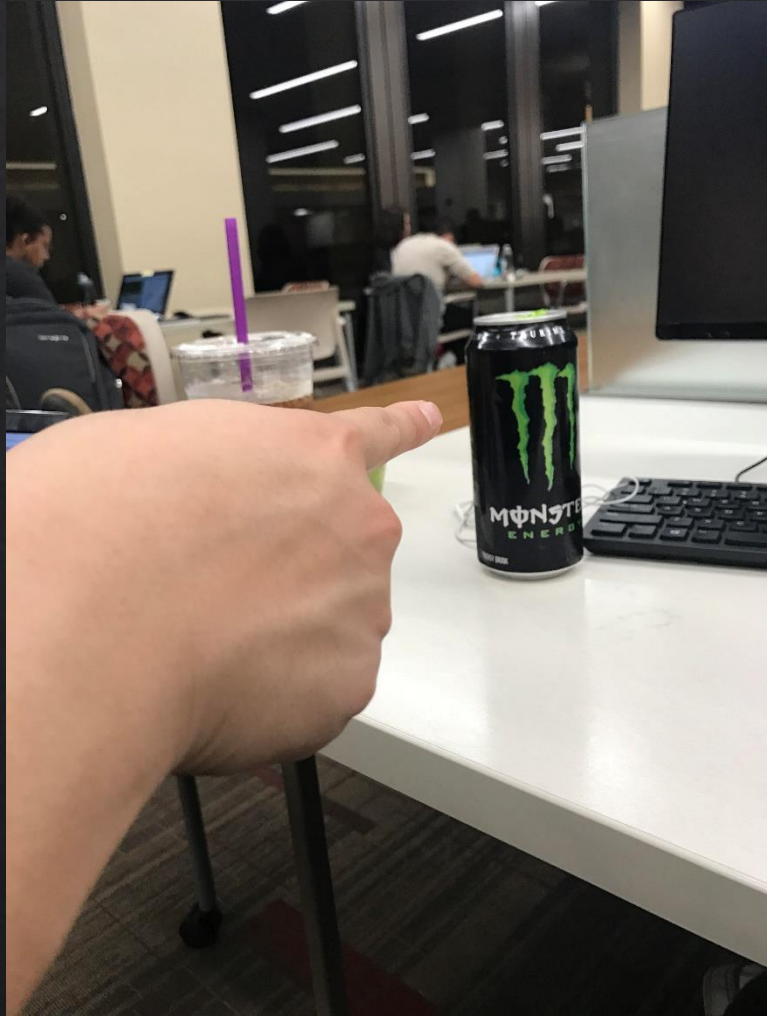


Result #1

Main function: insertMarker()



Result #2



Result #3



Thank you!

Q & A