Project Presentation

Image Processing & Autonomous Driving

2022.06

[MEE1005] C Programming Basic

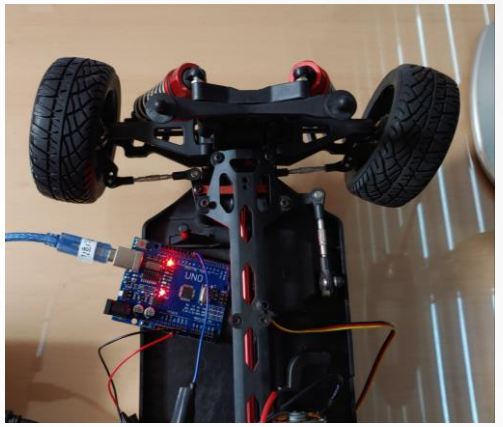
20171842 김희성

서강대학교 SOGANG UNIVERSITY

Project Introduction :

Background

- Computer Vision기술을 이용한 자율주행과 같이 차량 소프트웨어 개발이 이루어지고 있음.
- C++로 PC와 Arduino(Embedded System Architecture) 사이 Serial 통신 방법에 대한 호기심이 생김.



Objective

- 웹캠으로부터 받아오는 영상정보에서 얻은 라인(차선)의 기울기에 따라 모형자동차의 서보모터 조작을 통해 적절한 steering angle을 만들어낸다.

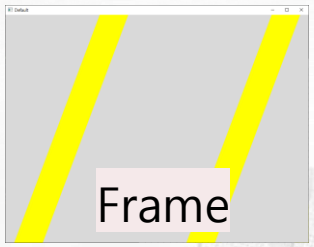
서강대학교 SOGANG UNIVERSITY

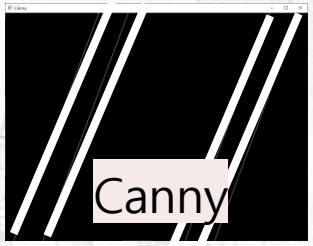
Implementation Method :

- Image Processing using OpenCV Library
- Configuration of Serial Communication
- Transmit Data

<Line Detecting Process>

OpenCV





main.cpp

- Grab Frame from Camera
- inRange Masking
- Canny Edge Detection
- Hough Transform
- Return Slope of Line

<Serial Communication between PC & Arduino>

SerialClass.h

- Serial Port Class Construction

Serial.cpp

- Define Port
- Open Port
- Initialize Communication
- Connect Port
- Transmit Data

- Receive Data
- Operate Servo

steer.ino

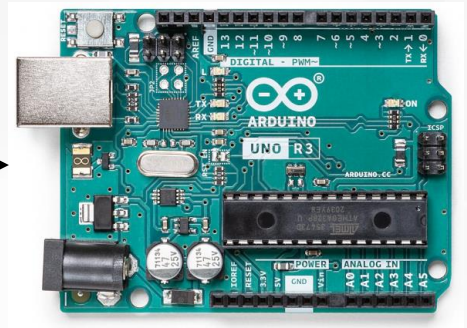


Image Processing Sequence : using namespace cv

- Grab frame from camera

```
Mat img_color;
cap.read(img_color);
```

- Gaussian Blur - reduce noise of the image- Convolution operation

```
Mat img_blur, img_hsv;
GaussianBlur(img_color2hough, img_blur, Size(3, 3), 0);
```

$$\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

7	2	3	3	8
4	5	3	8	4
3	3	2	8	4
2	8	7	2	7
5	4	4	5	4

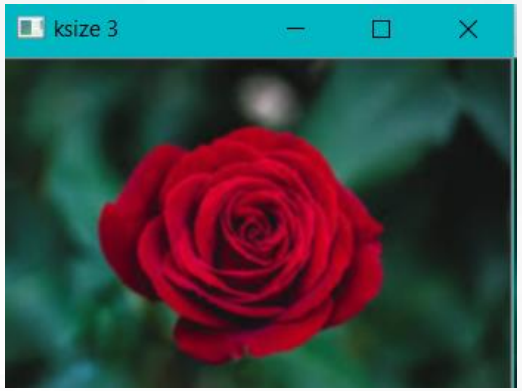
*

1	0	-1
1	0	-1
1	0	-1

7x1+4x1+3x1+
2x0+5x0+3x0+
3x-1+3x-1+2x-1
= 6

=

6		



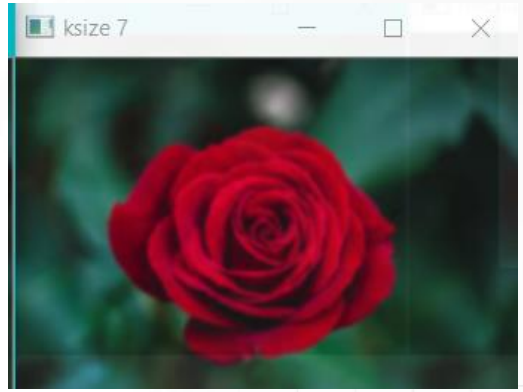
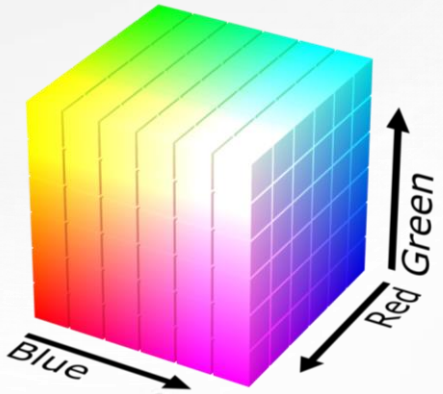
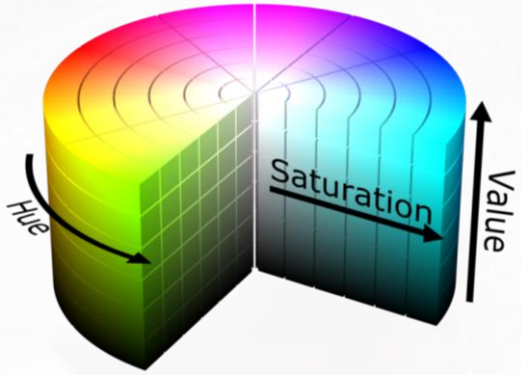


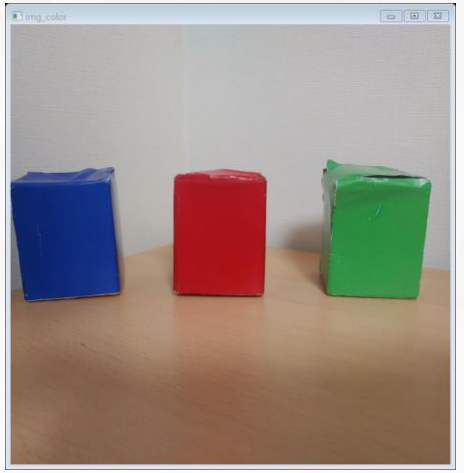
Image Processing Sequence : using namespace cv

- Convert color(BGR to HSV) – Similarity from human sight
  - Blue(0~255), Green(0~255), Red(0~255)
  - Hue(0~180), Saturation(0~255), Value(0~255)

```
cvtColor(img_blur, img_hsv, COLOR_BGR2HSV);
```

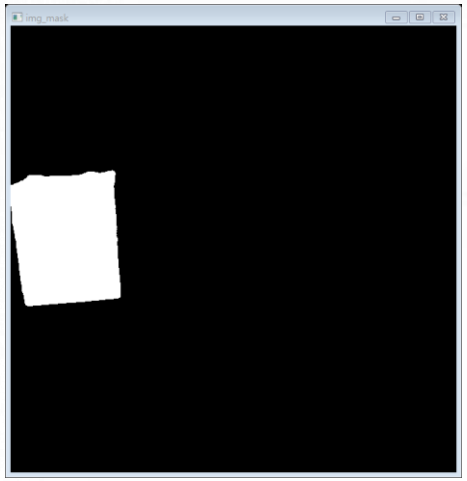






- inRange Masking

```
lower_blue1 = Vec3b(hue, threshold1, threshold1);
upper_blue1 = Vec3b(hue + 10, 255, 255);
lower_blue2 = Vec3b(hue - 10, threshold1, threshold1);
upper_blue2 = Vec3b(hue, 255, 255);
lower_blue3 = Vec3b(hue - 10, threshold1, threshold1);
upper_blue3 = Vec3b(hue, 255, 255);
```



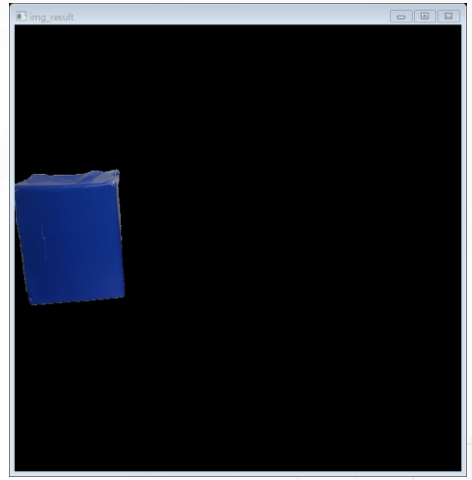
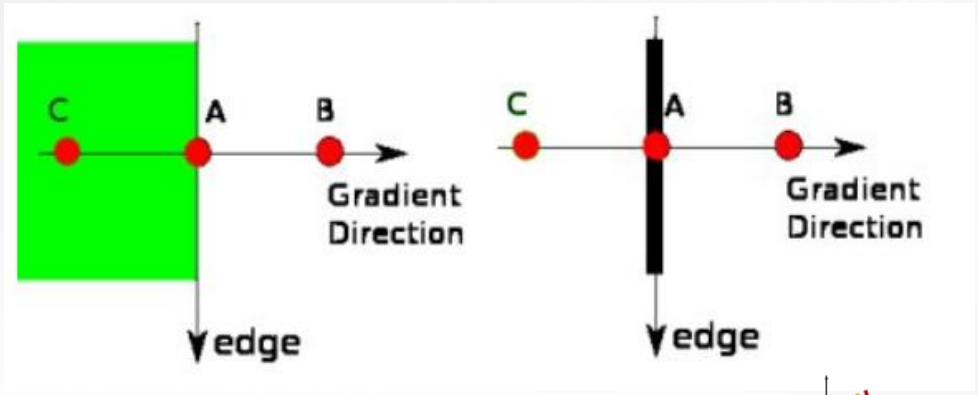
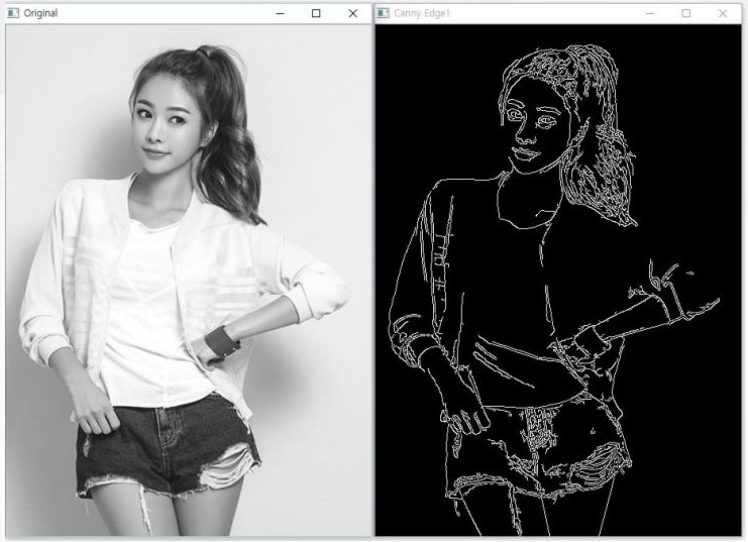


Image Processing Sequence : using namespace cv

- Canny Edge Detection

```
Mat img_canny;
Canny(img_result, img_canny, 50, 200);
```

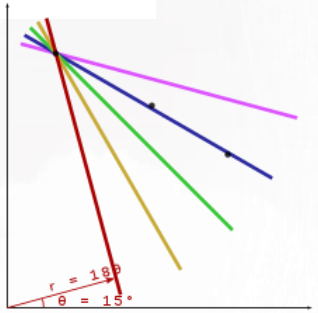




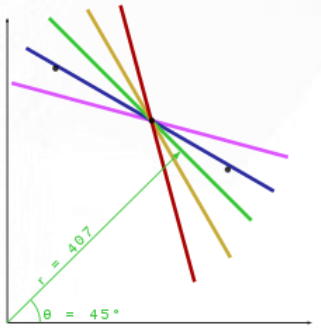
- Hough Transformation
  - Transform y=mx+b to

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r}{\sin\theta}\right)$$

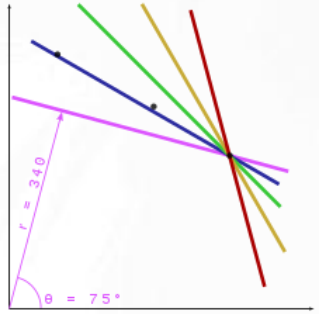
$$r = x\cos\theta + y\sin\theta$$



θ	r
15	189.0
30	282.0
45	355.7
60	407.3
75	429.4



θ	r
15	318.5
30	376.8
45	407.3
60	409.8
75	385.3



θ	r
15	419.0
30	443.6
45	438.4
60	402.9
75	340.1

Return Values ,Serial Communication, and Operation :

- Return Values
  - 'L', 'l', 'N', 'r', 'R' for returned angle range

```
double angle = drawHoughLines(lines);
char tilt;
if (angle < -30) { tilt = 'L'; }
else if (angle < -10 && angle >= -30) { tilt = 'l'; }
else if (angle > 30) { tilt = 'R'; }
else if (angle > 10 && angle <= 30) { tilt = 'r'; }
else { tilt = 'N'; }
int steer = TX_DATA(ser, tilt, angle);
```

- Transmit Data

```
ser->WriteData("L", 1);
```

- Receive Data

```
if(Serial.available()){
  state=Serial.read(); // receive c
  if (state == 'L'){value=55;}
  else if (state =='l'){value=66;}
  else if (state =='N'){value=80;}
  else if (state =='r'){value=94;}
  else if (state=='R'){value=105;}
```

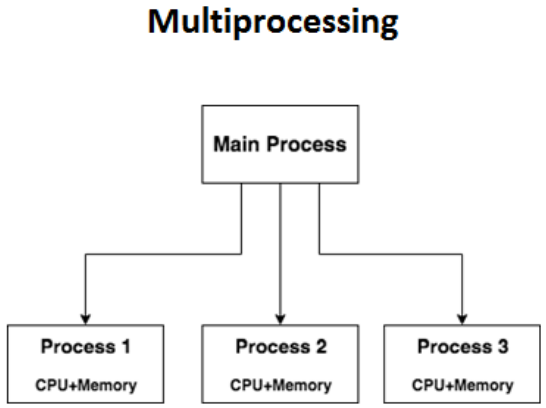
- Operate Servo

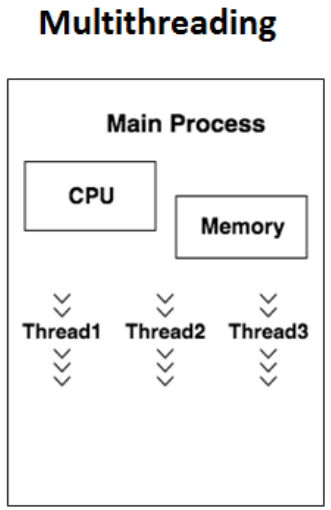
```
if (value0 < value)
{
  for (int i = value0 ; i < value; i++)
  {
    front.write(i);
    delay(15);
  }
value0=value;
}
else
{
  for (int i = value0; i > value ; i--)
  {
    front.write(i);
    delay(15);
  }
value0 = value;
}
```

Future Work :

Multithreading and Multiprocessing

- 현재는 HoughLine의 정보를 받아오면 Serial를 통해 정보를 전달하는 절차형 code로 이루어져 있다.
- 만약 메모리가 작은 Embedded architecture에서 프로그램을 실행할 경우, 명령 처리 속도가 느리고 실시간으로 반환값을 구하기가 어려워진다. 따라서 image processing, Serial transmitting이 동시에 진행할 수 있는 방안이 필요하다.





서강대학교 SOGANG UNIVERSITY

Demo

서강대학교
SOGANG UNIVERSITY