

로직

Bird eye view 사용

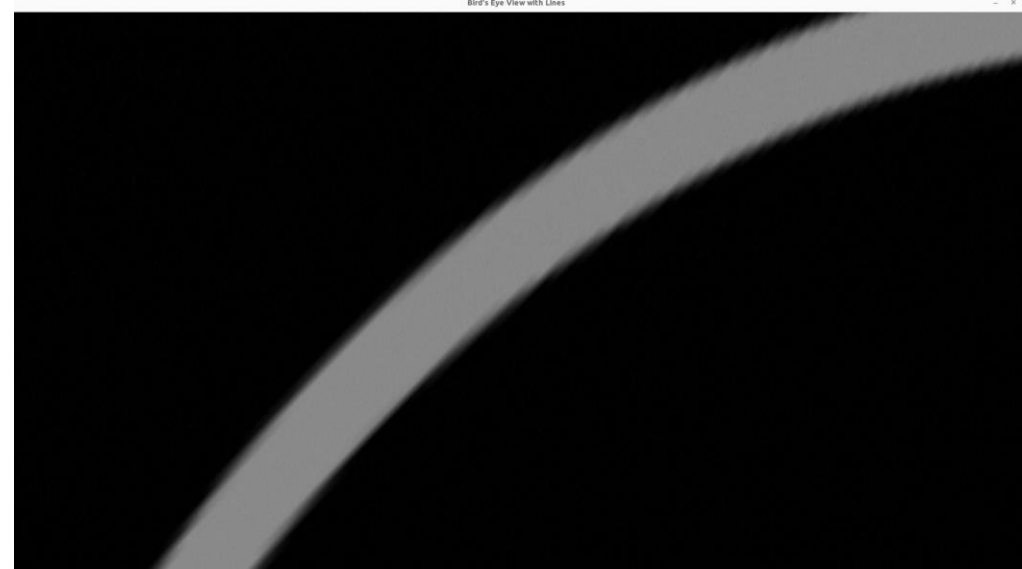
흰선, 노란선 둘다 존재할 경우 : 각속도 0.0

흰선만(노란선만) 감지 된 경우

```
std::vector<cv::Vec4i> lines_yellow, lines_white;
cv::HoughLinesP (edges_yellow, lines_yellow, 1, CV_PI / 180, 50, 50, 10);
cv::HoughLinesP (edges_white, lines_white, 1, CV_PI / 180, 50, 50, 10);
if(!yellow_detected && white_detected) {
if (l[0] < img_width * 4 / 5 || l[2] < img_width * 4 / 5) {
    angular_velocity_ = +0.7; // 더 강하게 회전
    break;
}
else if(l[0] < img_width * 3 / 5 || l[2] < img_width * 3 / 5) {
}
}
```

둘 다 감지 안된 경우 교차선으로 판단

토픽 받아오는것에 따라 회전 방향 달라짐



```
void transform(Point2f* src_vertices, Point2f* dst_vertices, Mat& src, Mat &dst){
    Mat M = getPerspectiveTransform(src_vertices, dst_vertices);
    warpPerspective(src, dst, M, dst.size(), INTER_LINEAR, BORDER_CONSTANT);
}

int main(){
    Mat src = imread("test.png");

    Point2f src_vertices[4];
    src_vertices[0] = Point(270,120);
    src_vertices[1] = Point(395, 120);
    src_vertices[2] = Point(560, 480);
    src_vertices[3] = Point(110, 480);

    Point2f dst_vertices[4];
    dst_vertices[0] = Point(0, 0);
    dst_vertices[1] = Point(640, 0);
    dst_vertices[2] = Point(640, 480);
    dst_vertices[3] = Point(0, 480);

    Mat M = getPerspectiveTransform(src_vertices, dst_vertices);
    Mat dst(480, 640, CV_8UC3);
    warpPerspective(src, dst, M, dst.size(), INTER_LINEAR, BORDER_CONSTANT);
}
```