



KANGWON NATIONAL UNIVERSITY

컴퓨터비전 실습

실습12 | RANSAC

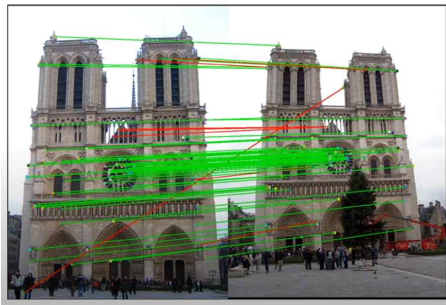
실습과제 이루리 내 제출

CVMIP LAB @ KNU

실습 12

- RANSAC 매칭 알고리즘

: 이상치(outliers)가 있는 데이터셋에서도 강인성(robustness)을 확보할 수 있는
매칭 알고리즘



실습 12

- RANSAC 매칭 알고리즘
(Random Sample Consensus)

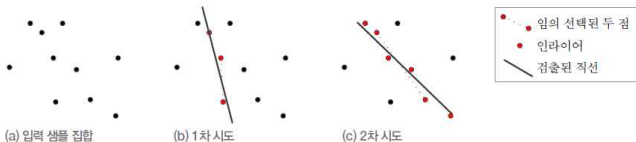


그림 3-31 RANSAC의 원리

실습 12-1 | 최소제곱법

문제

최소제곱법 알고리즘을 구현합니다.

- 오래 전부터 수학과 통계 분야에서 사용된 기법
- 예) $X=\{x_1, x_2, x_3, x_4\}$ 를 가장 잘 대표하는 직선을 찾아라. ← 회귀 문제

요구 결과

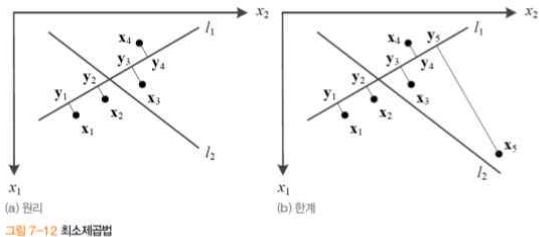
우측의 최소제곱법 알고리즘을 구현하여
이를 이용하여 무작위로 생성된 점들을
가장 잘 표현할 수 있는 직선을 구합니다.

과제 제출

실습 10-1, 10-2를 수행한 뒤

cpp 파일과 ransac.bmp, leastsquare.bmp 이미지들을 같이 압축하여

(최상위 디렉토리 없이) 이루리 시스템에 제출합니다.



실습 12-1 | 최소제곱법

설명자료

$$E(a, b) = \sum_{i=1}^n (y_i - (ax_i + b))^2 \rightarrow$$

최소제곱법

$$a = \frac{\sum_{i=1}^n (y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$
$$b = \bar{y} - a\bar{x}$$

a : 기울기

b : 절편

\bar{y} : y 평균

\bar{x} : x 평균

참고자료:

<https://m.blog.naver.com/PostView.nhn?blogId=moojigai07&logNo=120186757908&proxyReferer=https%3A%2F%2Fwww.google.com%2F>

실습 12-1 | 최소제곱법

설명자료

```
EquationElement algorithms::LeastSquare::calculate_least_square()
{
    double sumX = 0.0;
    double sumY = 0.0;

    double avgX = 0.0;
    double avgY = 0.0;

    // 기울기와 절편을 구하세요.
    for (auto& point : point_vector_list)
    {
        // ** 지금부터 코드를 작성하세요. 이 줄은 지우시면 안 됩니다 **
        // sumX +=
        // sumY +=
        // ** 여기까지 코드를 작성하세요. 이 줄은 지우시면 안 됩니다 **
    }

    avgX = sumX / point_vector_list.size();
    avgY = sumY / point_vector_list.size();

    double sumNumerator = 0.0;
    double sumDenominator = 0.0;

    for (auto& point : point_vector_list)
    {
        // Numerator와 Denominator는 각각 분자와 분모를 의미함
        // ** 지금부터 코드를 작성하세요. 이 줄은 지우시면 안 됩니다 **
        // sumNumerator +=
        // sumDenominator +=
        // ** 여기까지 코드를 작성하세요. 이 줄은 지우시면 안 됩니다 **
    }

    EquationElement elem;
    // elem.m =
    // elem.b =

    return elem;
}
```

$$a = \frac{\sum_{i=1}^n (y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$b = \bar{y} - a\bar{x}$$

a : 기울기

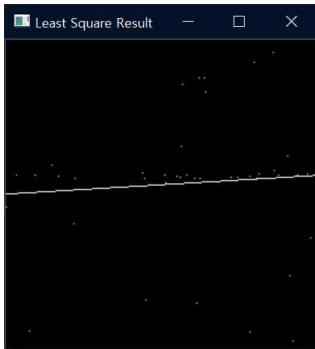
b : 절편

\bar{y} : y 평균

\bar{x} : x 평균

실습 12-1 | 최소제곱법

결과영상



실습 12-2 | RANSAC

문제

RANSAC 알고리즘을 구현합니다.

요구 결과

RANSAC 알고리즘을 이용하여 최적의 직선을 검출합니다.

검출 시 임의의 선택을 2,000번 수행하여 그 중 최적의 직선을 검출하여 화면에 표시합니다.

실습 12-2 | RANSAC

설명자료

알고리즘 3-8 직선 검출을 위한 RANSAC

입력: 예지 영상 $e(i, j)$, $0 \leq i \leq M-1$, $0 \leq j \leq N-1$ // 예지는 1, 배경지는 0인 이진 영상

반복 횟수 n , 인라이어 판단 t , 인라이어 집합의 크기 d , 직선 적합 오차 ϵ

출력: 하나의 직선(기울기 a 와 절편 b)

```
1 line = ∅;
2 for(loop = 1 to n) {
3     예지 화소 두 개를 임의로 선택한다.
4     이 두 점으로 직선의 방정식  $l$ 을 계산한다.
5     이 두 점으로 집합  $inlier$ 를 초기화한다.
6     for(이 두 점을 제외한 모든 예지 화소  $p$ 에 대해)
7         if( $p$ 가 직선  $l$ 에 허용 오차  $t$  이내로 적합)  $p$ 를  $inlier$ 에 넣는다.
8     if( $|inlier| \geq d$ ) { // 집합  $inlier$ 가  $d$ 개 이상의 샘플을 가지면
9          $inlier$ 에 있는 모든 샘플을 가지고 직선의 방정식  $l$ 을 새로 계산한다.
10        if( $l$ 의 적합 오차  $< \epsilon$ )  $l$ 을 집합  $line$ 에 넣는다.
11    }
12 }
13  $line$ 에 있는 직선 중 가장 좋은 것을 취한다.
```

현재 알고리즘에서 제외

실습 12-2 | RANSAC

설명자료

직선의 방정식을 이용하여 기울기와 절편을 계산하세요.

```
void CRANSAC::convert_two_point_to_line()
{
    // Point pt1과 Point pt2를 이용해서 기울기와 절편을 계산하세요
    // ** 지금부터 코드를 작성하세요. 이 줄은 지우시면 안 됩니다 **
    // lineEquation.m =
    // lineEquation.b =
    // ** 여기까지 코드를 작성하세요. 이 줄은 지우시면 안 됩니다 **
}
```


$y = mx + b$ 일때, $m = \frac{y_2 - y_1}{x_2 - x_1}$, $b = y - (m * x)$ 로 계산됨

실습 12-2 | RANSAC

설명자료

직선과 점 사이의 거리를 계산하세요.

```
void CRANSAC::calculate_inlier()
{
    std::vector<cv::Point> inlier;
    for (int i = 0; i < point_vector_list.size(); i++)
    {
        if (i != pt1Idx && i != pt2Idx)
        {
            // distance = |(-(x * m) - b + y) / root(m^2 + 1)|
            // point_vector_list[i]를 이용해서 거리를 계산하세요
            // ** 지금부터 코드를 작성하세요. 이 줄은 지우시면 안 됩니다 **
            // double distance =
            // ** 여기까지 코드를 작성하세요. 이 줄은 지우시면 안 됩니다 **
            if (distance <= 5)
                inlier.push_back(point_vector_list[i]);
        }
    }
    if (maxInlier < inlier.size())
    {
        maxInlier = inlier.size();
        optimalEquation.m = lineEquation.m;
        optimalEquation.b = lineEquation.b;
    }
}
```


$$distance = \left| \frac{-(x * m) - b + y}{\sqrt{m^2 + 1}} \right|$$

실습 12-2 | RANSAC

결과화면

