

심화전공실습 과제 7 보고서

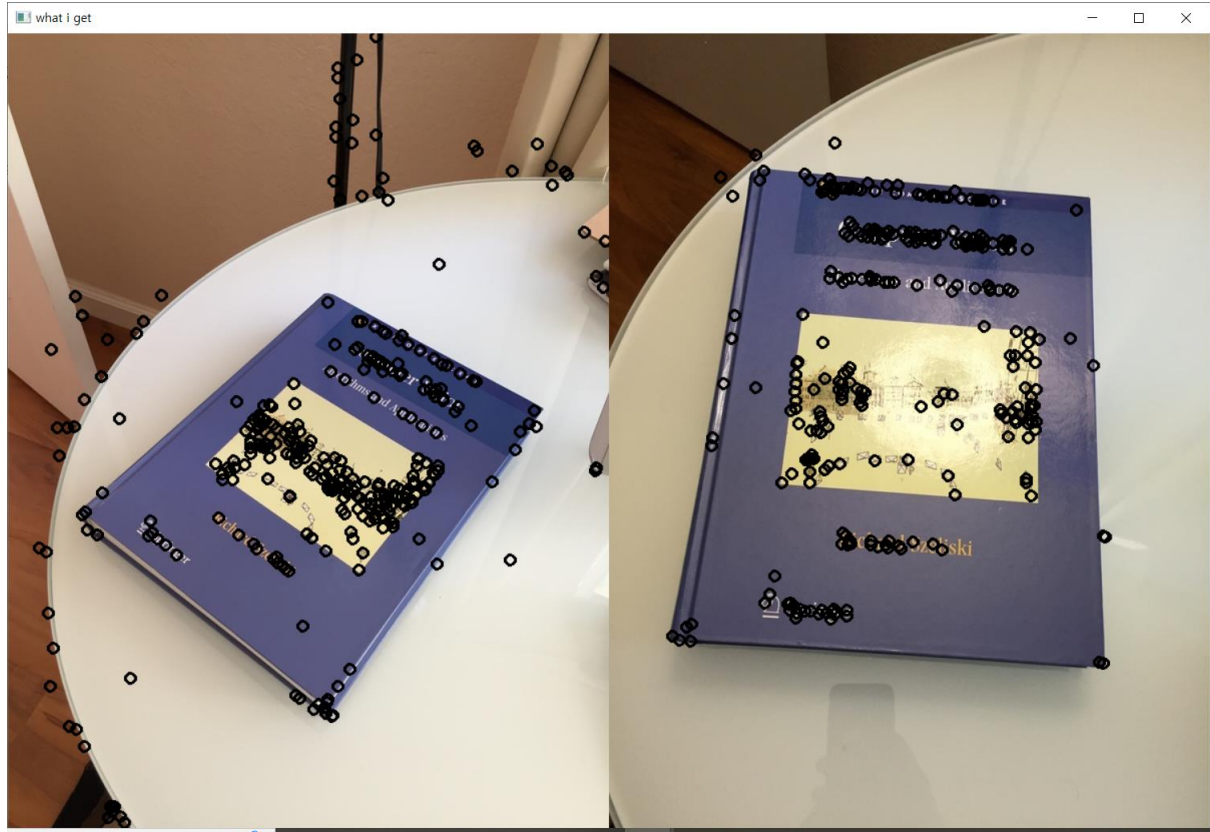
2019203021

소프트웨어학부

이승헌

1-1. SIFT feature detection

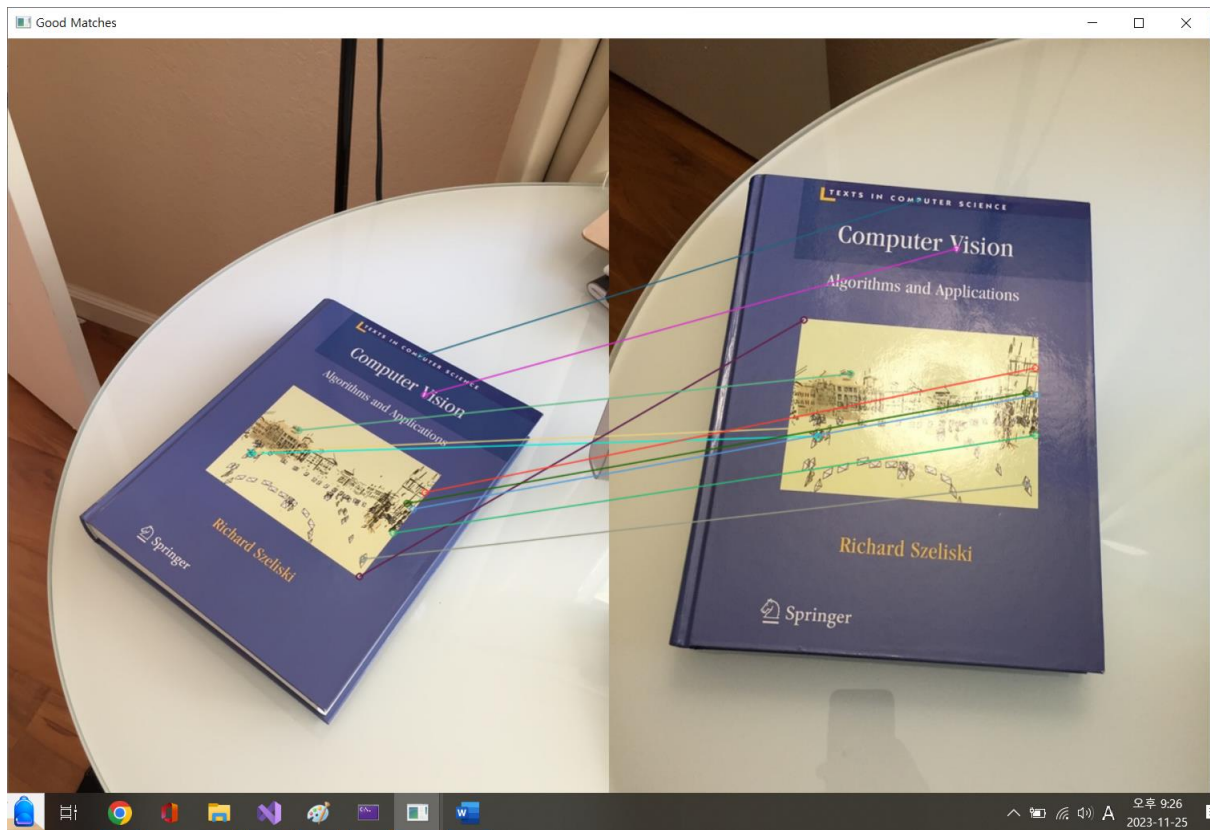
i) 책 이미지



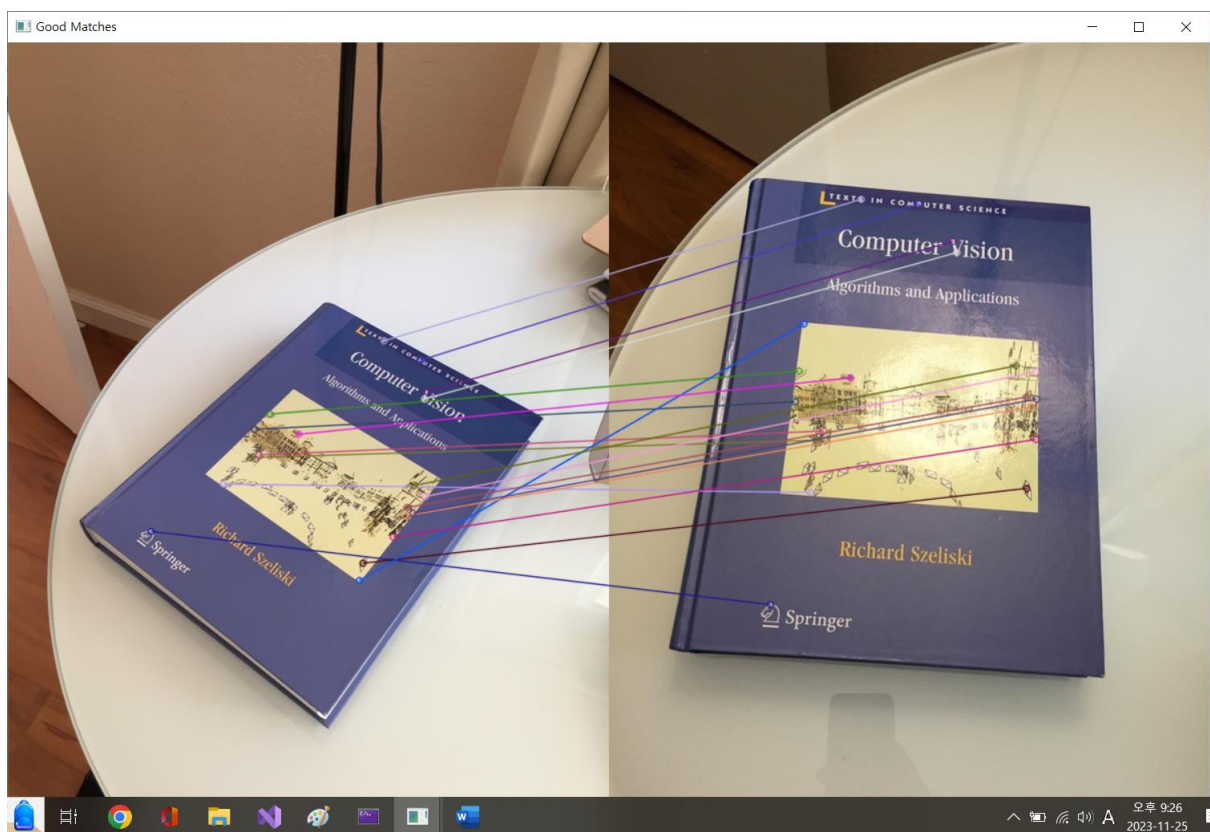
특징점

왼쪽 사진은 비교적 탁자, 문 등의 가구에 대한 특징점이 많이 잡힌 모습을 보여주고, 오른쪽 사진은 거의 책 쪽에만 특징점이 잡혀있다.

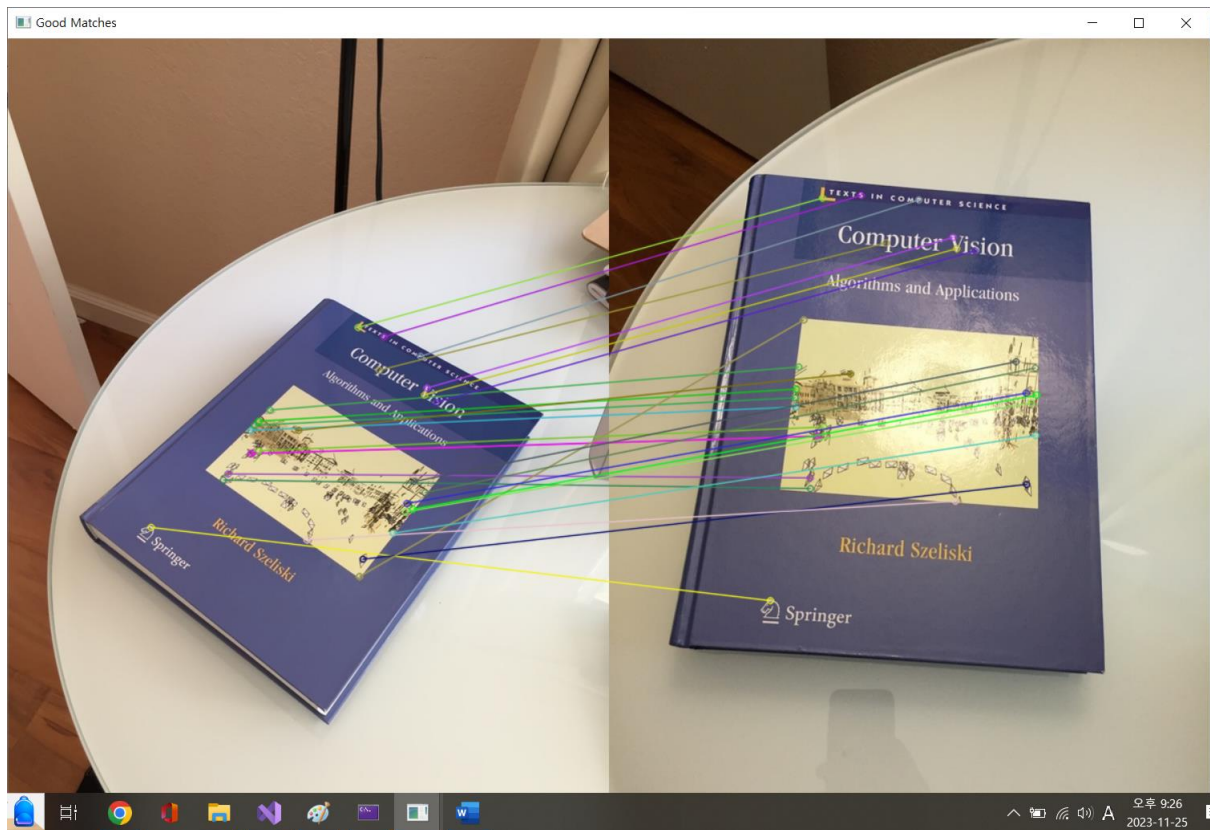
책에 잡힌 특징점의 경우 글자를 표시하는 부분은 어느 정도 유사하게 특징을 잡은 것으로 보인다. 중앙의 그림의 경우는 빛 때문인지 왼쪽과 오른쪽의 차이가 두드러진다.



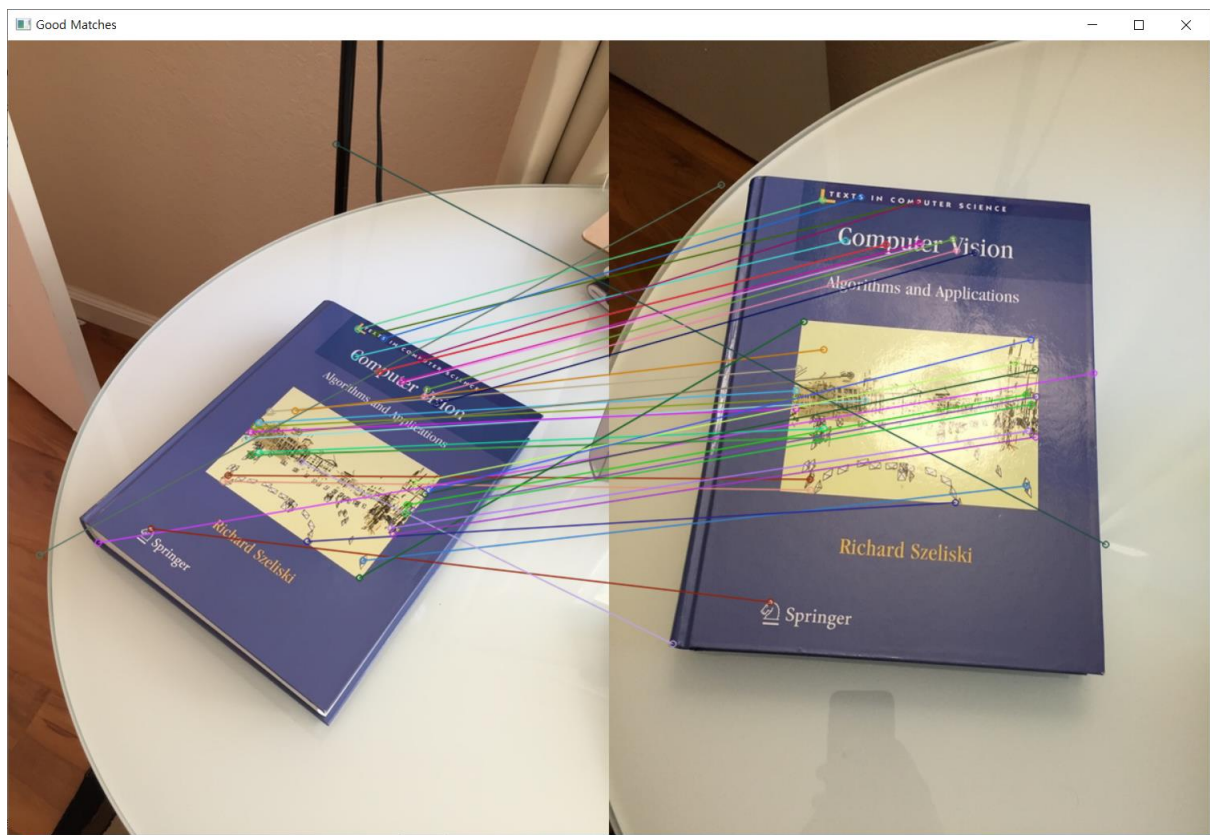
Threshold 0.4



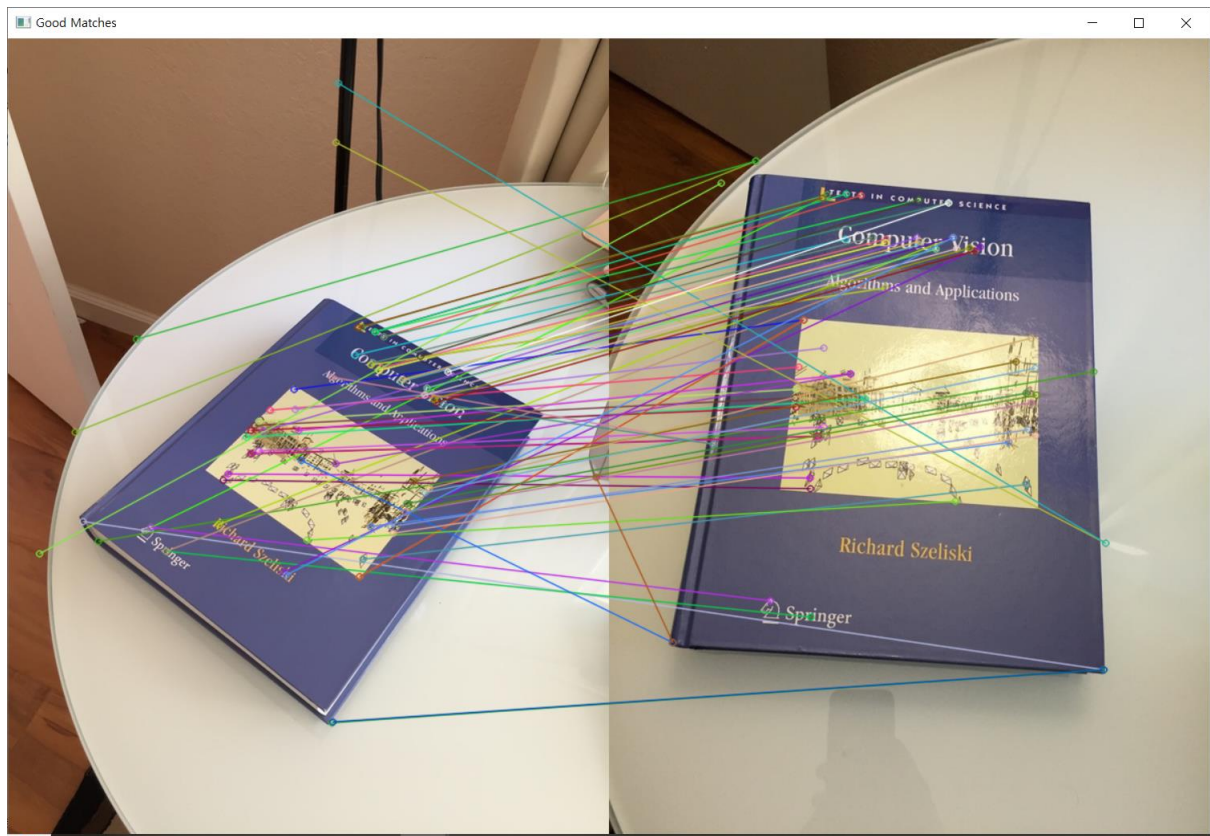
Threshold 0.5



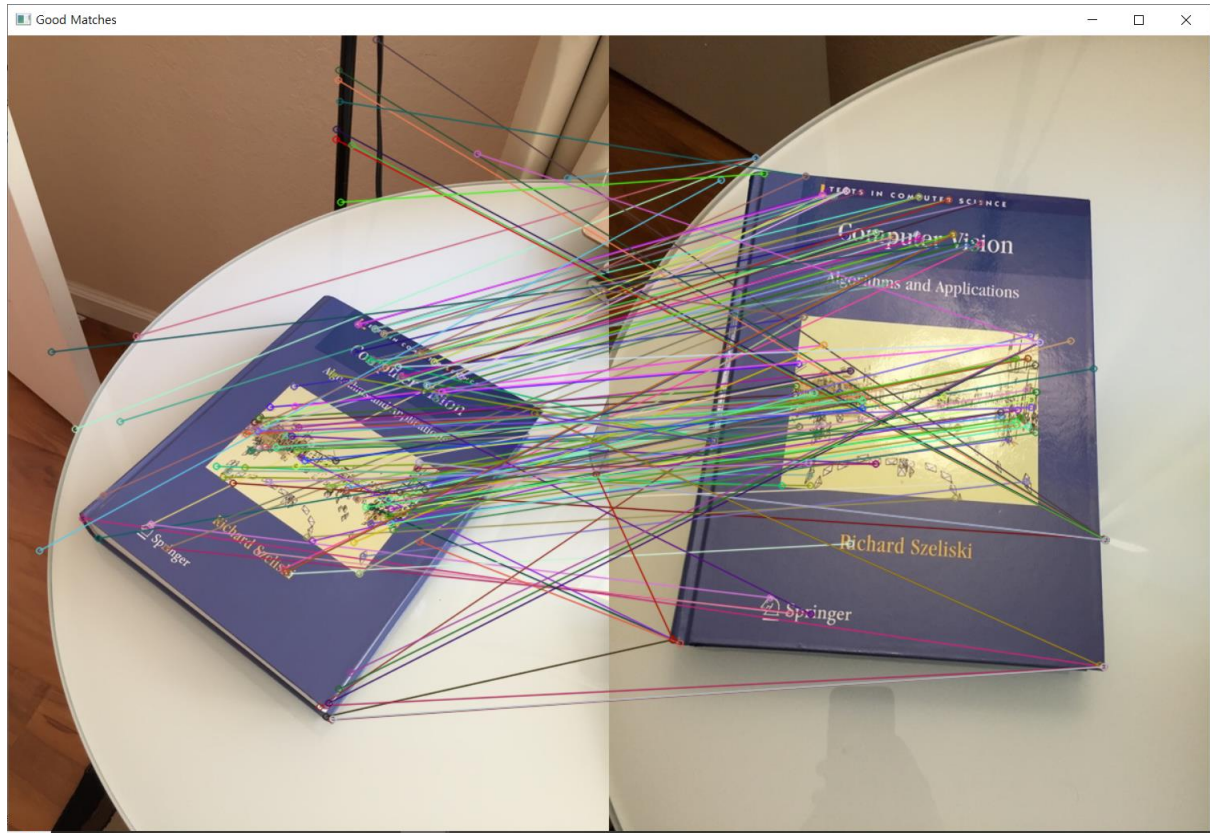
Threshold 0.6



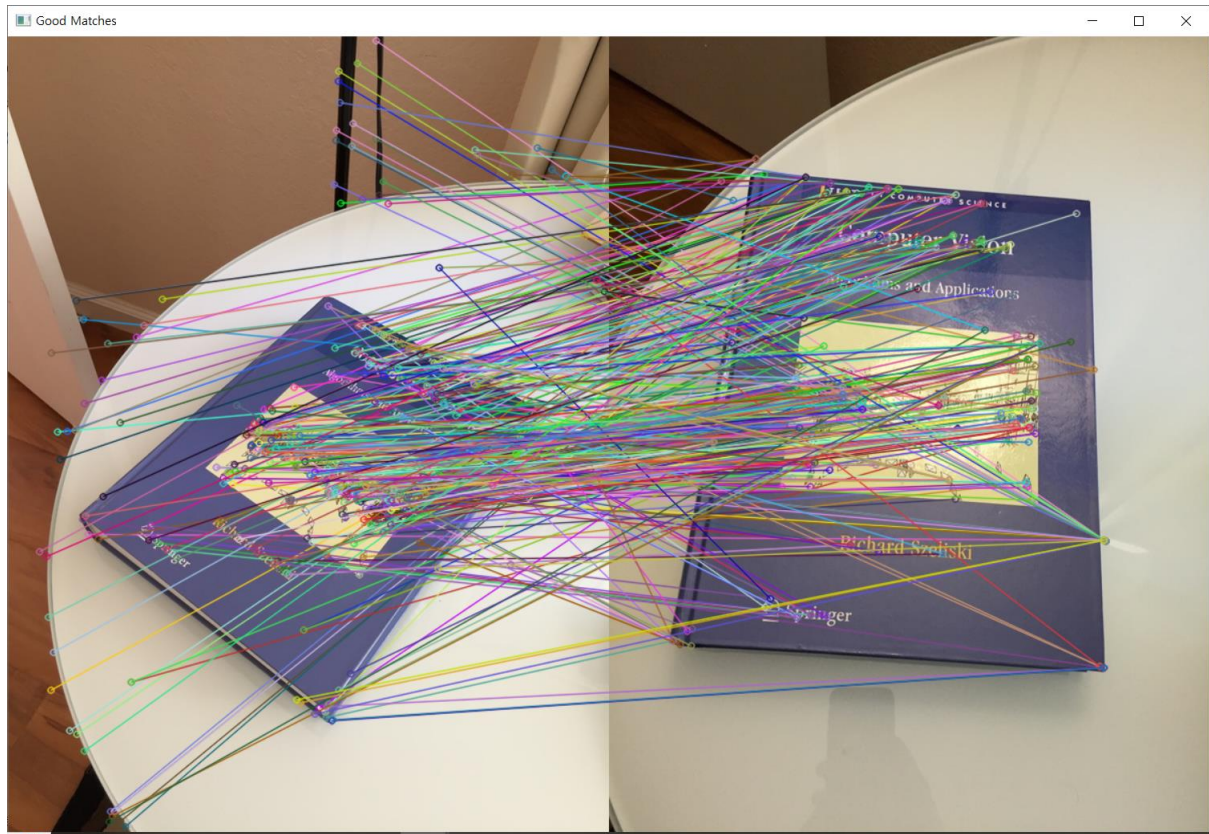
threshold 0.7



Threshold 0.8



threshold 0.9



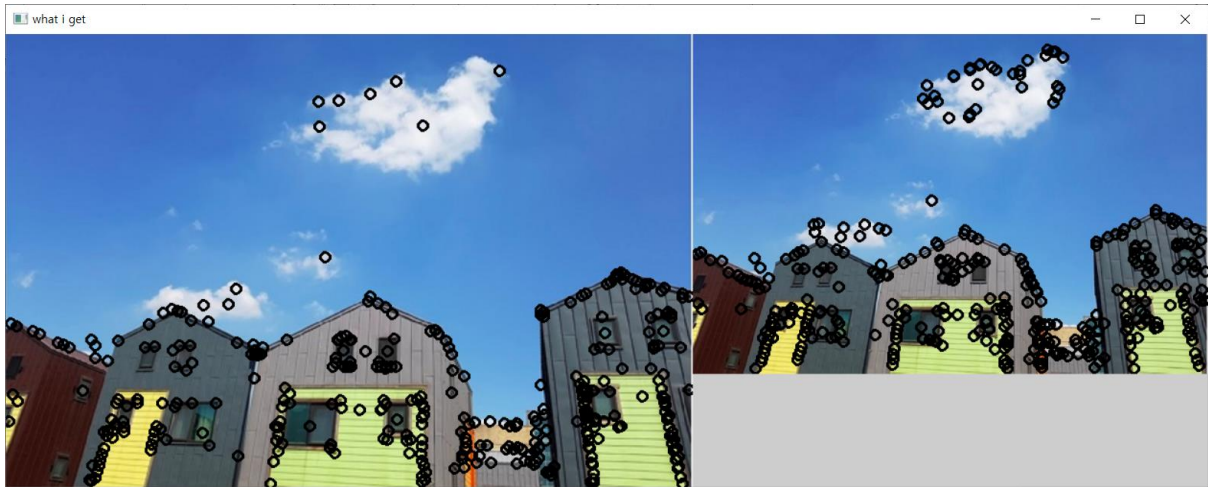
Threshold 1.0

Threshold 가 낮으면 유사도가 정말 높다고 여겨지는 특징점을 연결하는 모습을 보인다

Threshold 가 높을수록 많은 특징점을 연결하지만, 정확도는 떨어지는 모습을 보인다. 책 이미지의 경우 Threshold 가 0.7 을 넘어가는 시점부터 책과는 전혀 상관이 없는 가구와 매칭이 되는 모습을 보인다.

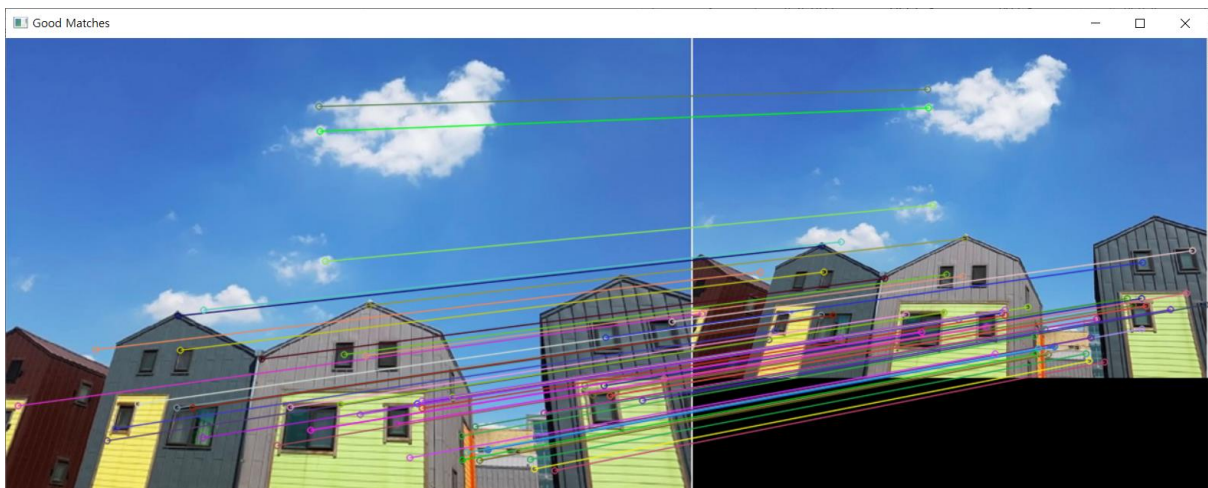
책에 대해 매칭하는 경우 Threshold 0.6 까지의 모습만 보면 매칭이 어느 정도 잘 되는 모습을 보인다. 다만 Threshold 가 낮을 때에도 왼쪽 이미지의 중앙 그림의 오른쪽 아래 끝 모서리와 왼쪽 이미지의 중앙 그림의 왼쪽 위 끝 모서리가 매칭되는 오류를 보인다. 패턴으로 생각을 해보면 그다지 유사하다고 여겨지는 부분은 아님에도 저런 매칭을 보이는 점은 의문점이다.

ii)집 이미지(디스플레이 한계로 인하여 입력 이미지에 resize 실행함)

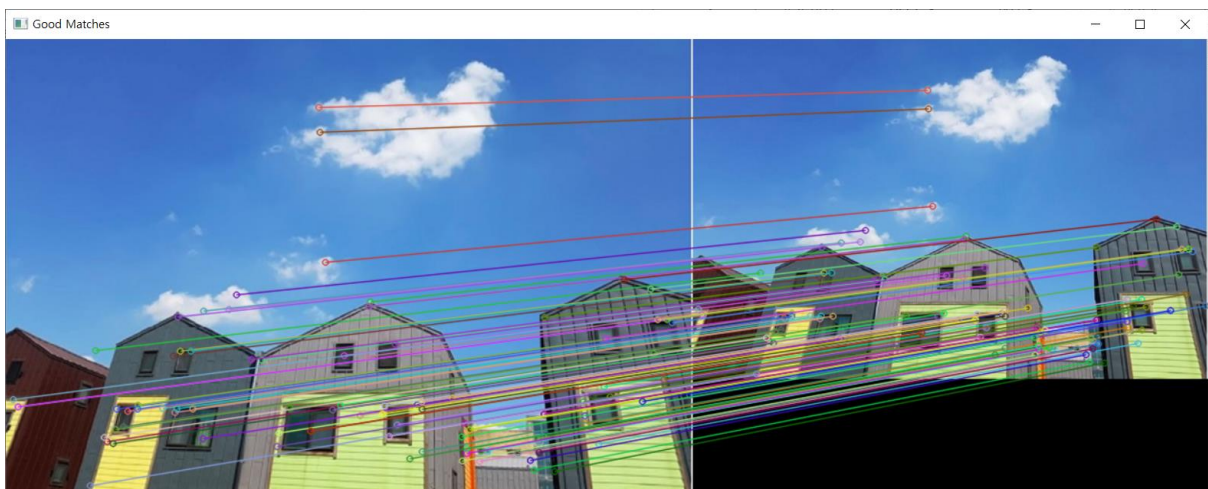


특징점

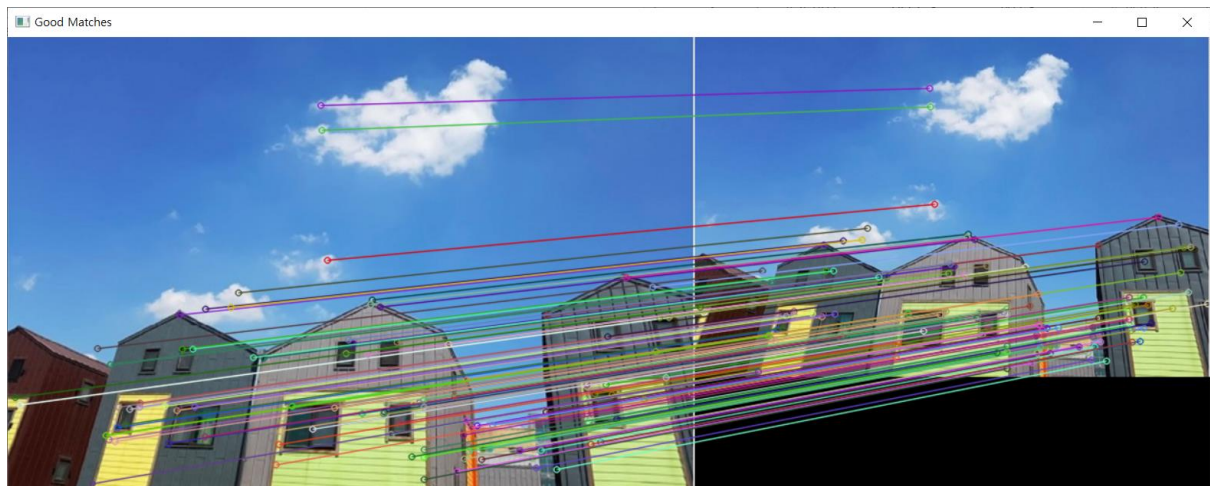
사이즈의 차이만 있어서 그런지 특징점도 밀도의 차이만 있을 뿐 대체적으로 유사하게 잡히는 것을 알 수 있었다.



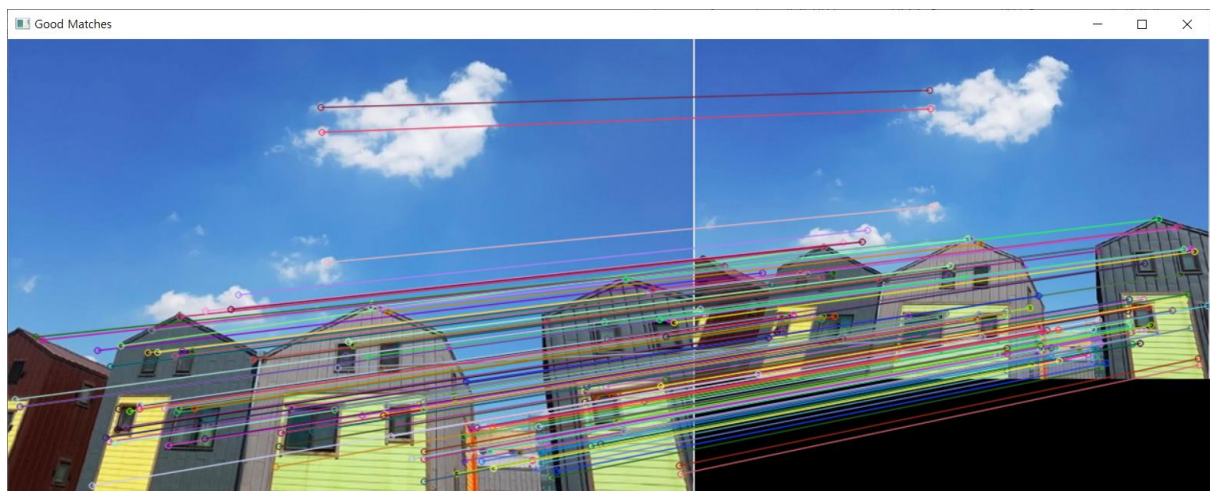
Threshold 0.2



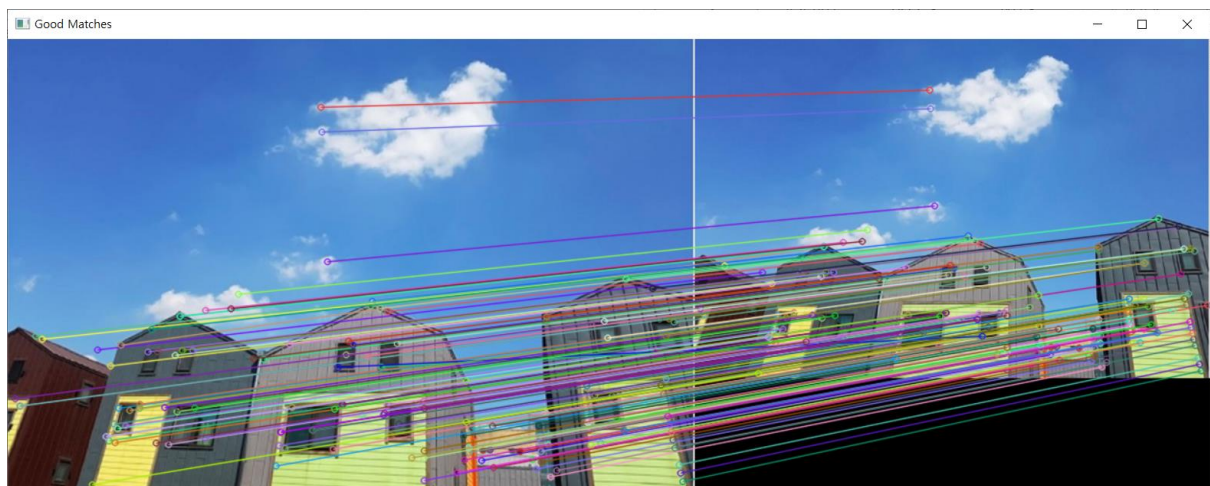
Threshold 0.3



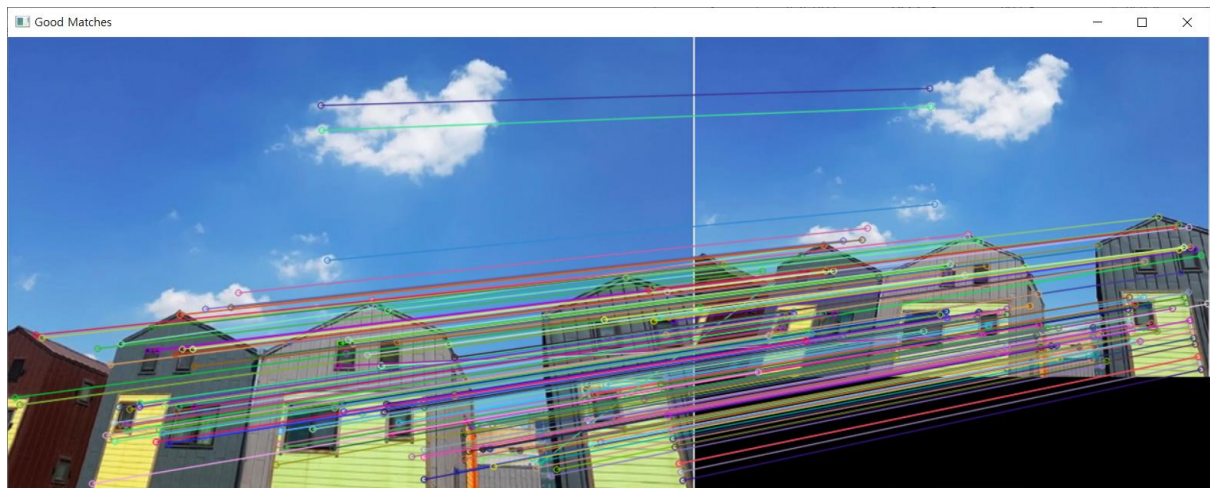
Threshold 0.4



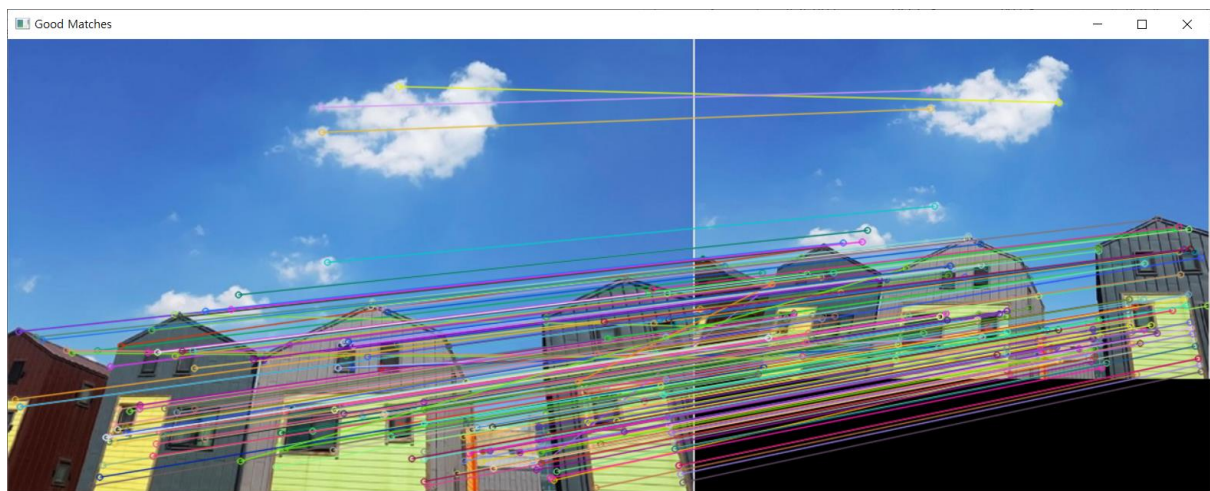
Threshold 0.5



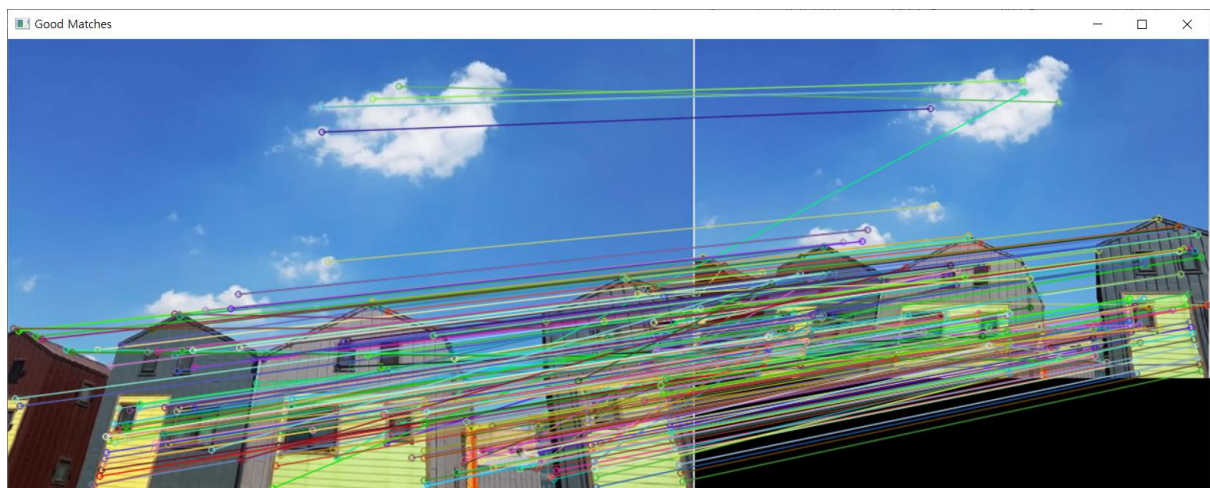
threshold 0.6



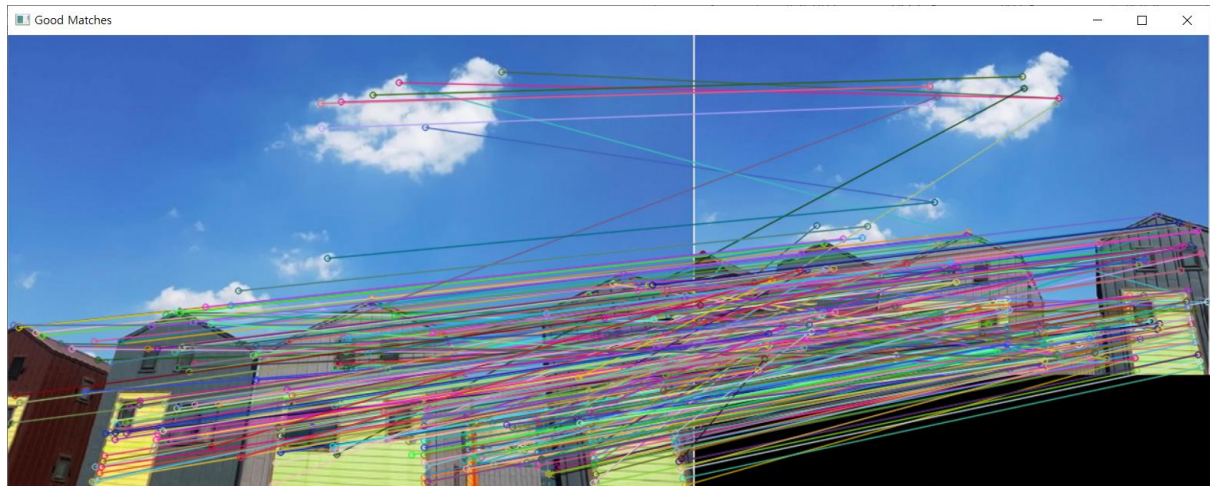
threshold 0.7



threshold 0.8



threshold 0.9



threshold 1.0

결국에는 Threshold 가 늘어나면 매칭하는 지점이 늘어나고 정확도가 떨어지는 것은 마찬가지지만, 보는 시점이 완전히 같은 이미지로 했을 때는 보는 시점이 아예 다른 이미지에 비해 Threshold 가 변할 때 매칭하는 지점이 그렇게 폭발적으로 늘지 않았다.

그리고 이 케이스 역시 Threshold 가 0.6 정도일 때는 육안으로 보면 어느 정도 매칭의 성능이 괜찮아 보였지만, 0.7 이 넘어서는 순간 육안으로도 엉뚱한 매칭이 확인할 수 있었다.

1-2 Harris Corner feature detection

i) 수식의 의미

opencv 에서 관련한 문서를 살펴보면 다음과 같이 나온다.

$$\text{dst}(x, y) = \det M^{(x,y)} - k \cdot \left(\text{tr} M^{(x,y)} \right)^2$$

$$M = \begin{bmatrix} \sum_{S(p)} (dI/dx)^2 & \sum_{S(p)} dI/dx dI/dy \\ \sum_{S(p)} dI/dx dI/dy & \sum_{S(p)} (dI/dy)^2 \end{bmatrix}$$

S(p)는 (x, y) 주변의 (block Size)x(block Size) 커널이다.

그리고 sobel operator 를 이용해 영상의 미분을 구현한다고 한다.

즉, M 의 경우 M(0,0)은 x 주변에서 Intensity 편미분의 제곱한 것들의 합,
M(0,1)과 M(1,0)은 x 주변에서 Intensity 편미분과 y 주변에서 Intensity 편미분의 곱한 것들의 합, M(1,1)은 y 주변에서 Intensity 편미분의 제곱한 것들의 합이다.

M 은 2x2 행렬이므로 M 의 행렬식은 M(0,0)xM(1,1) – M(0,1)xM(1,0)이다.

M(0,0)xM(1,1)은 (x 주변 변화량의 제곱의 합)x(y 주변 변화량의 제곱의 합)이고,
M(0,1)xM(1,1)은 ((x 주변 변화량) x (y 주변 변화량)의 합)^2 이다.

trM 은 2x2 행렬의 대각선 성분의 곱이다. 이는 M(0,0)xM(1,1)과 같다.

즉... R 은 $\text{tr}M - M(0,1)xM(1,0) - k \times \text{tr}M \Rightarrow \text{tr}M(1-k) - M(0,1)xM(1,1)$ 이다.

ii) 최대, 최소

```
dst max:6.031507e-03  
dst min:-5.161418e-03
```

Lena.png

```
dst max:5.317023e-03  
dst min:-1.144880e-02
```

Input.jpg

위 식에서 trM 을

(x 주변 픽셀들의 변화량의 제곱의 합) x (y 주변 픽셀들의 변화량의 제곱의 합)

이라고 생각을 한다면 제곱의 합을 곱한 이 값은 항상 0 이상의 양수이다.

또한, k 은 항상 0 이 아닌 양수이다.

$M(0,1) \times M(1,1)$ 은 $((x \text{ 주변 변화량}) \times (y \text{ 주변 변화량}))^2$ 이고 이 또한 제곱수이므로 항상 0 이상의 양수이다. 즉, 각 변수의 크기에 의해서만 R 이 결정된다는 말이다.

$\text{tr}M(1-k) > M(0,1) \times M(1,1)$ 이면 R 이 양수고,

$\text{tr}M(1-k) = M(0,1) \times M(1,1)$ 이면 R 이 0 이며,

$\text{tr}M(1-k) < M(0,1) \times M(1,1)$ 이면 R 이 음수다.

어느 한 쪽으로만 변화가 크다고 하면 $\text{tr}M$ 보다 $M(0,1) \times M(1,1)$ 이 더 값이 커지고 음수가 된다.

두 쪽 모두 변화가 크다고 하면 $\text{tr}M$ 이 $M(0,1) \times M(1,1)$ 보다 더 값이 커지고 양수가 된다. 따라서, Lena 이미지에 input 보다 양쪽에서 급한 변화가 있는 영역이 있었다고 볼 수 있다. Input 이 음수 값이 더 작은 것으로 보아 input 이미지에 Lena 보다 한 쪽으로만 더 급한 변화가 있는 영역이 있었다고 볼 수 있다.

iii) R 값의 결정



코너를 감지하는 데 좋은 영상은 아니라고 생각은 되지만, 영상마다 Flat 의 기준이 다르기 때문에 영상의 좋고 나쁨은 상관하지 않고 이 이미지를 기준으로 Flat 값을 결정해보았다.

$\text{pow}(10, -2) * 7.3$, 대략 0.073 보다 작은 값을 Flat 으로 고려할 때의 결과다.



대략 0.07 을 전후로 한 값이 이 이미지에는 적절한 것 같았다.

이것보다 낮은 수로 잡으면 ADAPT 부분에 너무 난잡하게 특징점이 잡히는 모습을 볼 수 있었다. 이것보다 높은 수로 잡으면 특징점이 너무 적어지는 문제가 있었다.

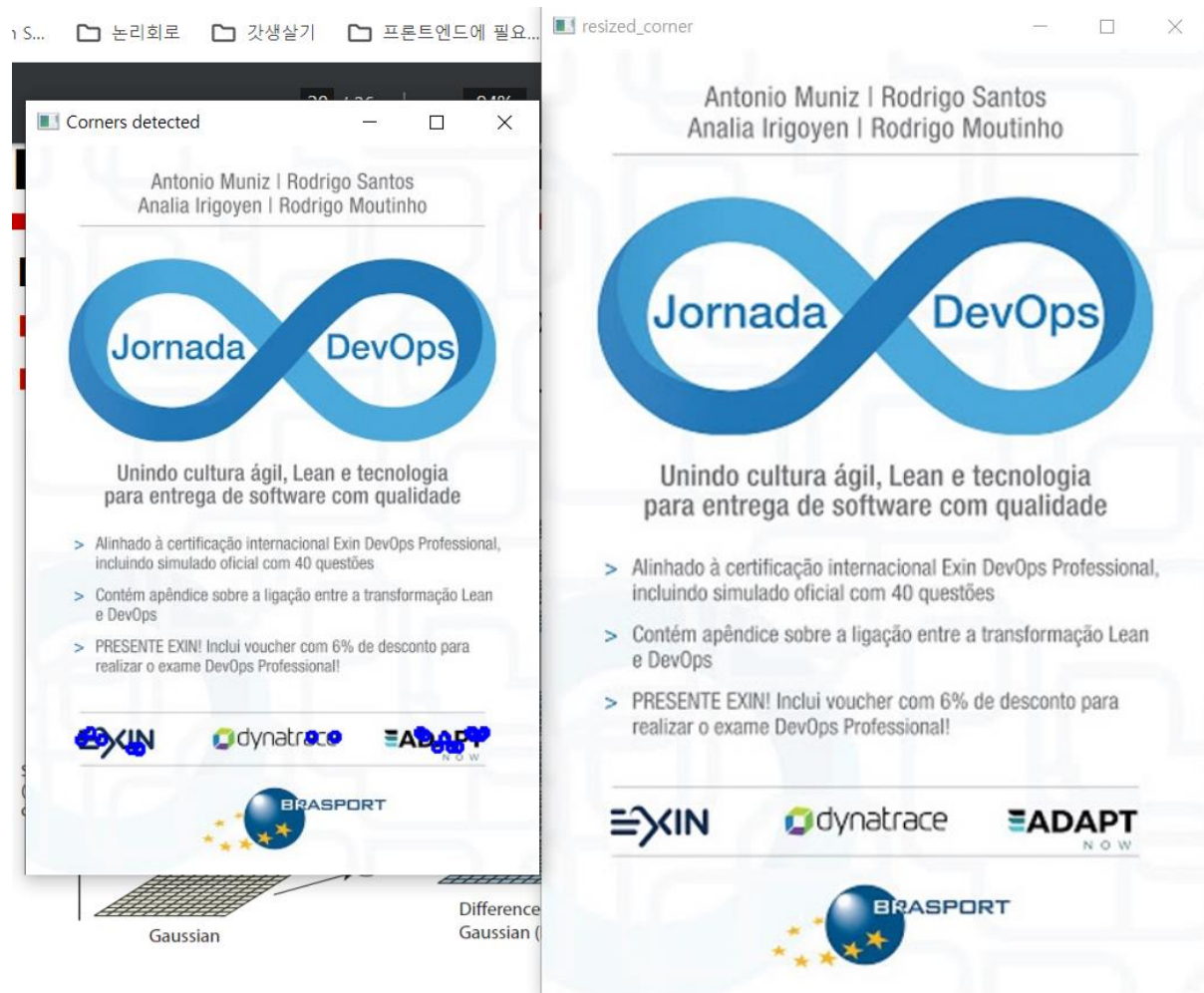
iv)코드의 Threshold 의 의미

Harris Corner 값이 음수, 양수인지와는 무관하게 0~255 로 R 값을 정규화하는 코드이다. 개인적으로 생각하기에는 절댓값으로만 판단하는 코드라고 생각이 된다.

그래서, 최초의 예제 코드에서 Threshold 의 의미는 R 값을 0~255 로 정규화하여 줄을 세워 봤을 때 Flat 인 것을 가려내기 위한 값이라고 생각이 된다.

2. Harris Corner feature detection 의 scale-variant 함을 보임

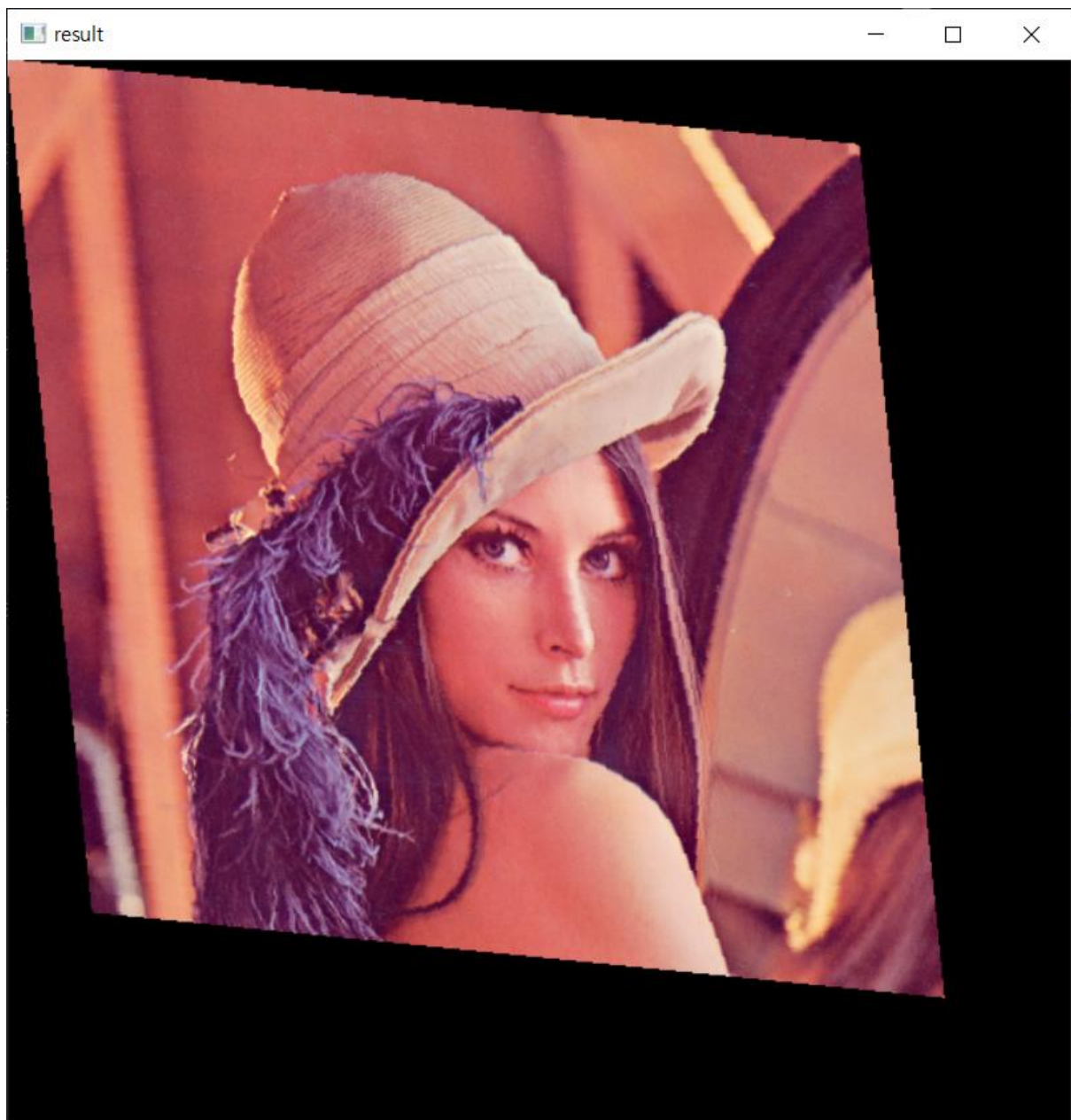
Threshold: $\text{pow}(10, -2) * 7.3$, 대략 0.073



scale-variant 한 것을 관찰할 수 있었다.

Opencv 의 예제에서 정규화된 R 값을 사용하는 것도 적절한 R 값이 영상의 크기에 따라 변하기 때문이라고도 생각이 된다.

3. image shearing



$$M = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x + hy \\ y \end{bmatrix}$$

$$\begin{bmatrix} h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} hx + y \end{bmatrix}$$

h 를 0.1 로 잡았다.

소수점으로 잡아주지 않으면 픽셀 좌표에 곱하는 형태이기 때문에 픽셀 좌표가 커질수록 비틀림이 커져 화면에 보이지 않게 되는 걸 알 수 있었다.