Template Matching

정의

- Template
  - 입력 영상 중 찾고자 하는 영상

- Template Matching
  - 입력 영상 (input image) 전체에서 지정된 template 영상과 유사한 부분 영상위치를 찾는 것

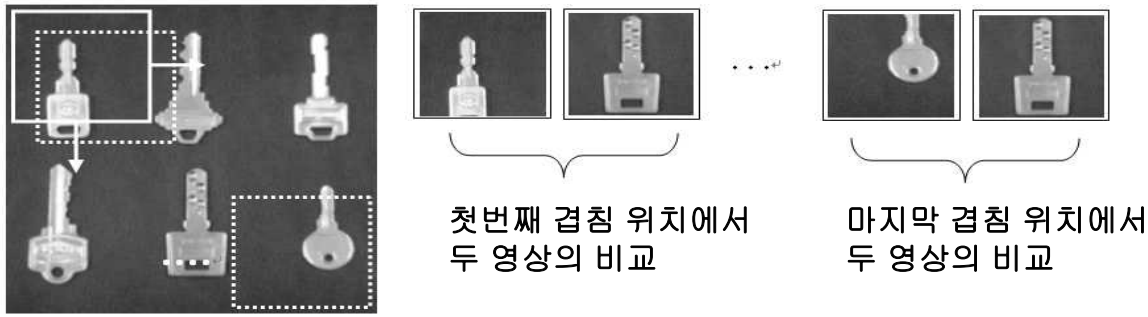
Input image

Template

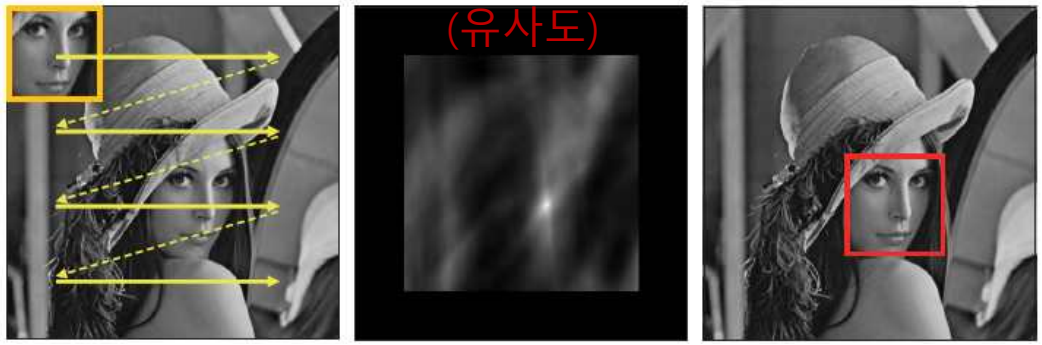
방법

- Matching
  - Template 을 입력 영상에 스캔하며 비교
  - Matching 이 가장 잘되는 부분 영상 선택
    = 유사도 (similarity) 가 가장 높은 위치 선택

(ex1)



(ex2)



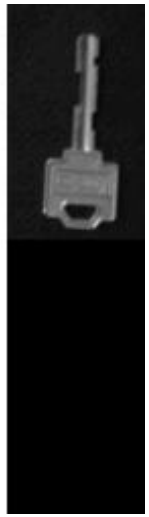
유사도 측정

- Mean Square Error

$$R(x,y)=\sum_{x',y'}\left(T(x',y')-I(x+x',y+y')\right)^2$$

  - template size: M x N, input size: R x C
  - overlap size: (R-M) x (C-N)
  - MSE 뺄셈횟수: (R-M) x (C-N) x (M x N) ≒ $O(n^4)$

(input)

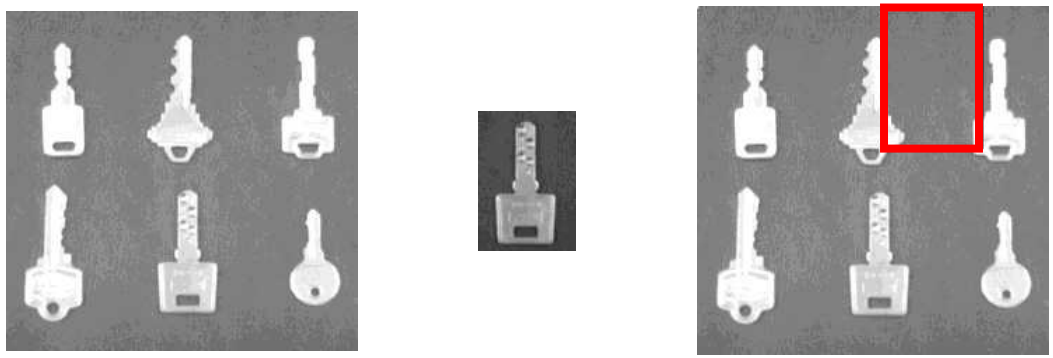


(template)

Min. MSE value = 12.1
Min. MSE position: (183, 6)

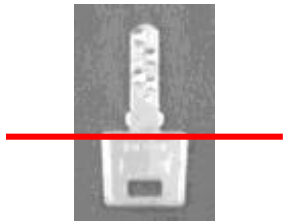
유사도 측정

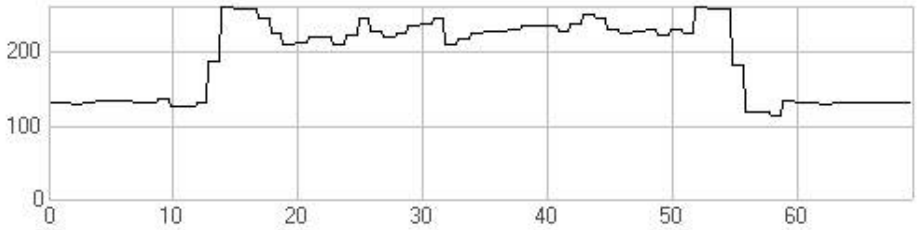
- Normalized MSE
  - template 와 image 의 intensity 분포에 차이가 큰 경우 MSE matching 어려움



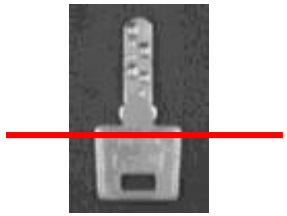
유사도 측정

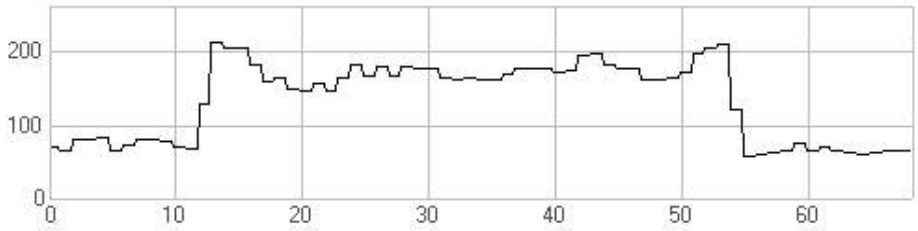
- Normalized MSE
  - Intensity distribution





Input 영상의 일부분



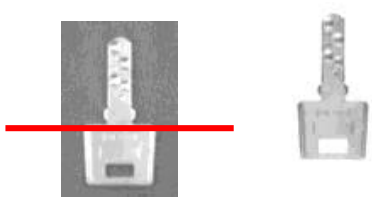


Template 영상

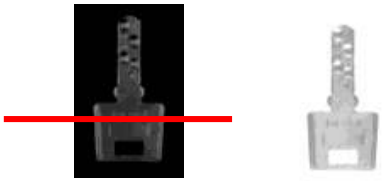
유사도 측정

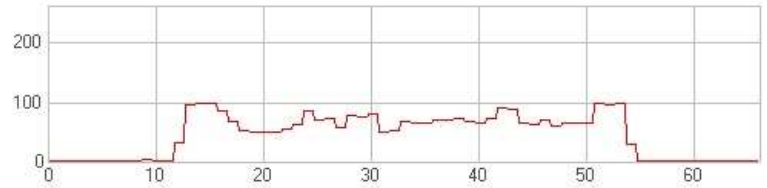
- Normalized MSE
  - Intensity normalization

$$R(x,y)=\frac{\sum_{x',y'}\left(T(x',y')-I(x+x',y+y')\right)^2}{\sqrt{\sum_{x',y'}T(x',y')^2\cdot\sum_{x',y'}I(x+x',y+y')^2}}$$



input 영상에 대한 밝기 정규화

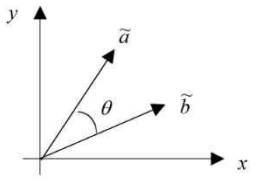




Template 영상에 대한 밝기 정규화

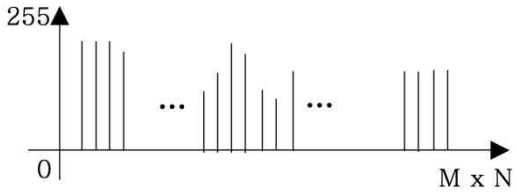
유사도 측정

- Correlation
  - Correlation (2 vectors)



$$\cos\theta = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{|\boldsymbol{a}|\,|\boldsymbol{b}|} = \begin{cases} 1 & ,\text{일치} \\ 0,\ -1 & ,\text{불일치} \end{cases}$$

  - Correlation (2 images)



  - Normalized correlation

$$R(x,y) = \frac{\sum_{x',y'} T(x',y') \cdot I(x+x',y+y')}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}}$$

유사도 측정

- Correlation Coefficient
  - Correlation 계산 시 템플릿 및 입력영상의 평균값 고려

$$R(x,y)=\frac{\sum_{x',y'}T'(x',y')\cdot I'(x+x',y+y')}{\sqrt{\sum_{x',y'}T'(x',y')^2\cdot\sum_{x',y'}I'(x+x',y+y')^2}}$$

$$T'(x',y')=T(x',y')-1/(w\cdot h)\cdot\sum_{x'',y''}T'(x'',y'')$$

$$I'(x+x',y+y')=I(x+x',y+y')-1/(w\cdot h)\cdot\sum_{x'',y''}I(x+x'',y+y'')$$

OpenCV 함수

- matchTemplate()

```
void matchTemplate(InputArray image, InputArray templ,
                   OutputArray result, int method, InputArray mask = noArray());
```

• image	입력 영상. 8비트 또는 32비트 실수형
• templ	템플릿 영상. 입력 영상 image보다 같거나 작아야 하며, image와 타입이 같아야 합니다.
• result	(출력) 비교 결과를 저장할 행렬. CV_32FC1 타입
• method	템플릿 매칭 비교 방법. TemplateMatchModes 열거형 상수 중 하나를 지정합니다.
• mask	찾고자 하는 템플릿의 마스크 영상. mask는 templ과 같은 크기, 같은 타입이어야 합니다. TM_SQDIFF와 TM_CCORR_NORMED 방법에서만 지원됩니다.

OpenCV 함수

- matchTemplate()

TemplateMatchModes 열거형 상수	설명
TM_SQDIFF	제곱차 매칭 방법 $R(x,y)=\sum_{x',y'}(T(x',y')-I(x+x',y+y'))^2$
TM_SQDIFF_NORMED	정규화된 제곱차 매칭 방법 $R(x,y)=\frac{\sum_{x',y'}(T(x',y')-I(x+x',y+y'))^2}{\sqrt{\sum_{x',y'}T(x',y')^2\cdot\sum_{x',y'}I(x+x',y+y')^2}}$
TM_CCORR	상관관계 매칭 방법 $R(x,y)=\sum_{x',y'}T(x',y')\cdot I(x+x',y+y')$
TM_CCORR_NORMED	정규화된 상관관계 매칭 방법 $R(x,y)=\frac{\sum_{x',y'}T(x',y')\cdot I(x+x',y+y')}{\sqrt{\sum_{x',y'}T(x',y')^2\cdot\sum_{x',y'}I(x+x',y+y')^2}}$
TM_CCOEFF	상관계수 매칭 방법 $R(x,y)=\sum_{x',y'}T'(x',y')\cdot I'(x+x',y+y')$ $T'(x',y')=T(x',y')-1/(w\cdot h)\cdot\sum_{x'',y''}T'(x'',y'')$ $I'(x+x',y+y')=I(x+x',y+y')-1/(w\cdot h)\cdot\sum_{x'',y''}I(x+x'',y+y'')$
TM_CCOEFF_NORMED	정규화된 상관계수 매칭 방법 $R(x,y)=\frac{\sum_{x',y'}T'(x',y')\cdot I'(x+x',y+y')}{\sqrt{\sum_{x',y'}T'(x',y')^2\cdot\sum_{x',y'}I'(x+x',y+y')^2}}$

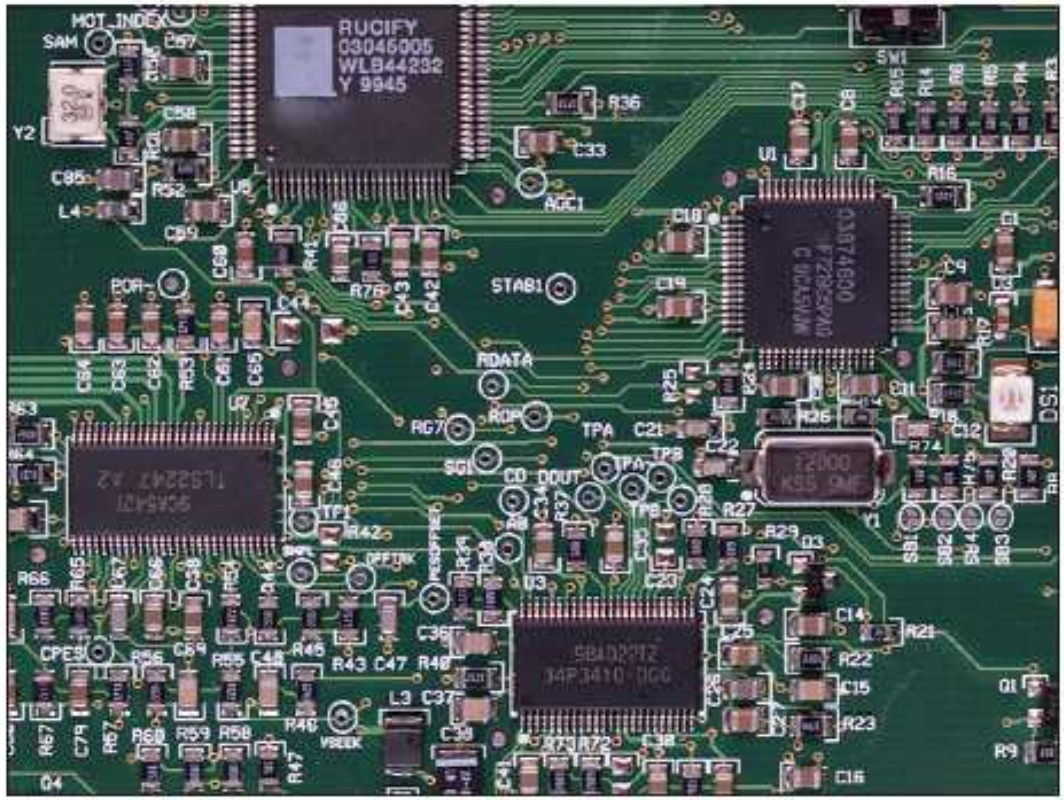
예제 프로그램

코드 13-1 템플릿 매칭 예제 [ch13/template]

```
01  void template_matching()
02  {
03      Mat img = imread("circuit.bmp", IMREAD_COLOR);
04      Mat templ = imread("crystal.bmp", IMREAD_COLOR);
05
06      if (img.empty() || templ.empty()) {
07          cerr << "Image load failed!" << endl;
08          return;
09      }
10
11      img = img + Scalar(50, 50, 50);
12
13      Mat noise(img.size(), CV_32SC3);
14      randn(noise, 0, 10);
15      add(img, noise, img, Mat(), CV_8UC3);
16
17      Mat res, res_norm;
18      matchTemplate(img, templ, res, TM_CCOEFF_NORMED);
19      normalize(res, res_norm, 0, 255, NORM_MINMAX, CV_8U);
20
21      double maxv;
22      Point maxloc;
23      minMaxLoc(res, 0, &maxv, 0, &maxloc);
24      cout << "maxv: " << maxv << endl;
25
26      rectangle(img, Rect(maxloc.x, maxloc.y, templ.cols, templ.rows),
                  Scalar(0, 0, 255), 2);
27
28      imshow("templ", templ);
29      imshow("res_norm", res_norm);
30      imshow("img", img);
31
32      waitKey(0);
33      destroyAllWindows();
34  }
```

예제 프로그램

- 입력

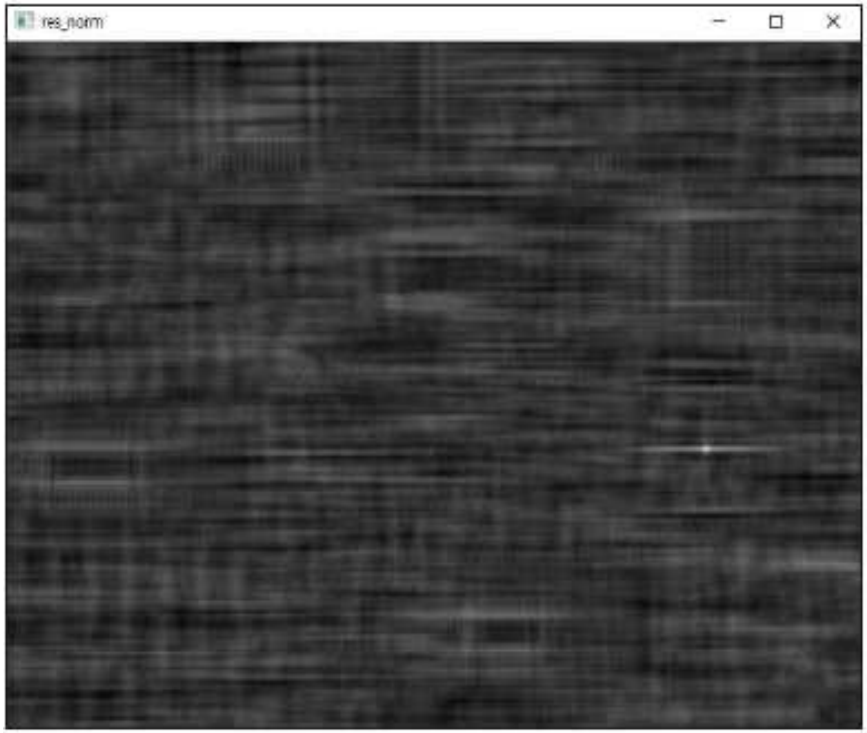


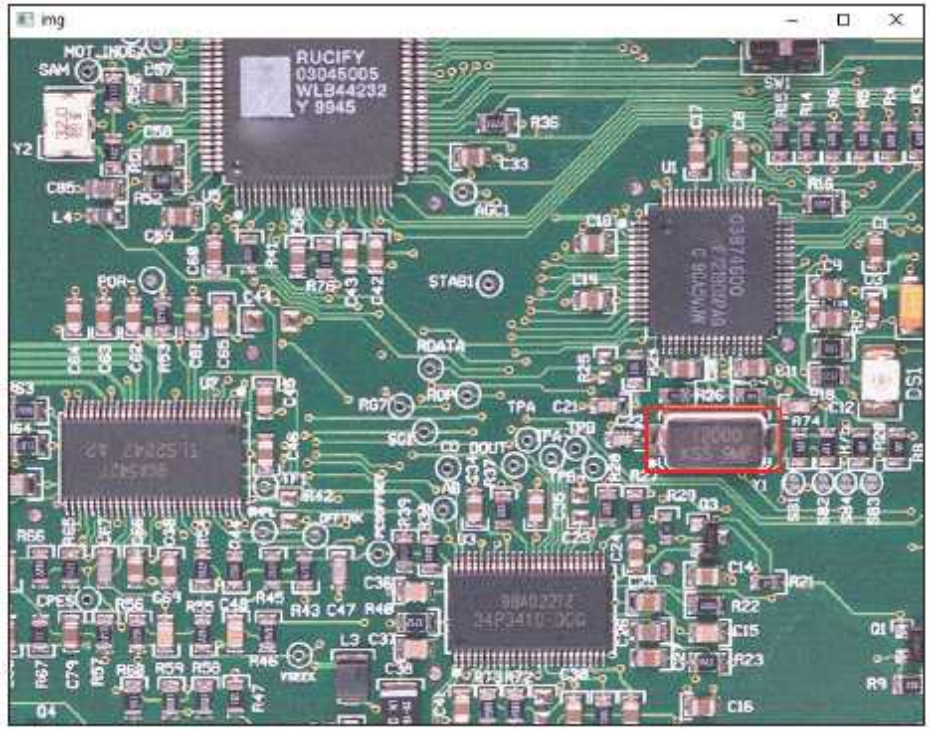
Input (img)

Template (temp1)

예제 프로그램

- 출력





유사도 (res_norm)

Maxv=0.976

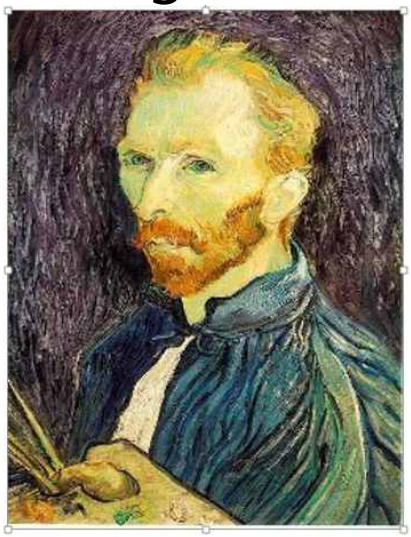
분석

- Object Detection / Recognition 적용 시
  - object shape 의 complexity 에 별 영향 받지 않음
  - object 의 위치 및 자세, 영상의 잡음 및 밝기 변화에 민감할 수 있음
  - 계산량 증가 우려
  - Template 등록 및 관리 필요

  ➢ 비교적 단순한 detection / recognition 분야에 활용 (예: 산업용 검사 시스템)

Image Pyramid

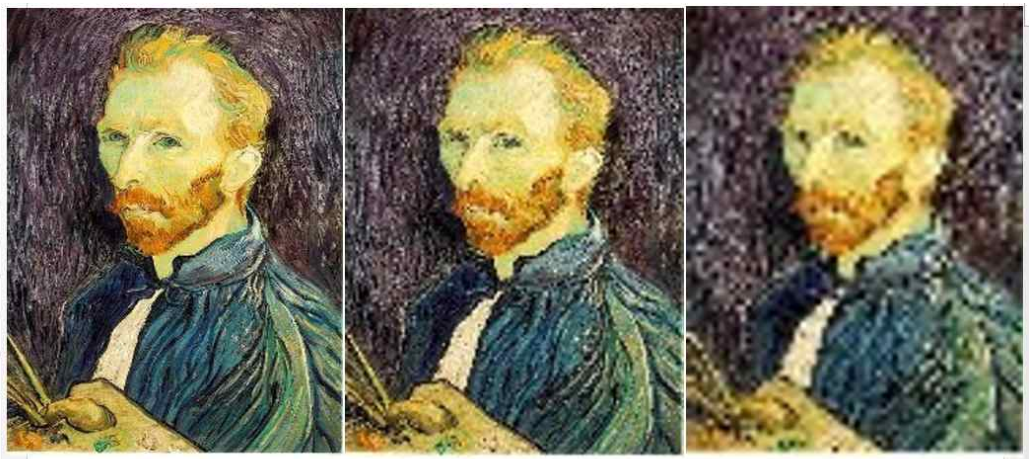
- Image sub-sampling



1/4

1/8

Throw away every other row and column to create a 1/2 size image - called *image sub-sampling*



1/2 1/4 (2x zoom) 1/8 (4x zoom)

Image Pyramid

- Image pyramid
  - 하위레벨에서 상위레벨까지 영상크기를 줄여나가는 기법
- Matching 방법
  - 피라미드의 상위레벨에서 대락적인 매칭을 수행하고
  - 최적 위치 근방에서 다음 하위 레벨의 영상 탐색
    - ➢ matching 소요 시간 감소