

메디치소프트 기술연구소

CH16. OpenCV: 이미지 피라미드

16. 이미지 피라미드

- 16.1 테스트 환경

순 번	제 목	설치 버전
1	운영 체제	Windows 10 64bit
2	프로그래밍언어	Python3.6.5

16. 이미지 피라미드

▪ 16.2 이미지 피라미드란?

- 이미지를 작업을 할 때, 일반적으로 일정한 크기의 이미지를 가지고 작업을 한다. 하지만 동일한 이미지의 여러 해상도의 다양한 크기를 가지고 작업할 필요가 있을 때가 있다. 예를 들면 이미지에서 사람의 얼굴과 같이, 객체에서 특정한 무언가를 탐색할 때 이다. 우리가 확보한 얼굴 이미지가 있으며, 다른 이미지에서 이 얼굴과 동일한 이미지를 탐색한다고 할 때, 우리가 가진 이미지에서의 얼굴 크기와 다른 이미지에서 보이는 얼굴 크기가 다를 수 있다. 이를 경우에는 탐색 대상 이미지의 해상도를 다단계로 만들어 각 단계에서 얼굴을 탐색하면 좀 더 나은 결과를 보일 수 있는 것이다.
- 이러한 해상도에 따른 다단계 이미지 세트를 이미지 피라미드라 부른다. 이렇게 부르는 이유는 원본 이미지의 해상도가 제일 크고, 단계가 높아질 수록 이미지 해상도가 줄어들기 때문에 마치 피라미드와 같이 이미지 단계가 놓여져 있기 때문이다. 즉 하위 단계가 상위 단계 아래에 놓여져 있는 이미지가 됩니다.

16. 이미지 피라미드

▪ 16.2 이미지 피라미드란?

- 이미지 피라미드의 종류

(1) 가우시안 피라미드

가우시안 피라미드는 상위 단계(해상도가 낮은 이미지) 이미지를 생성하는 것(downsampling)과 하위 단계 이미지(해상도가 높은 이미지)를 생성하는 것(upsampling)이 있다. 다운샘플링은 하위 단계 이미지의 짝수열, 짝수행에 해당하는 픽셀을 제거함으로써 이미지 해상도를 줄인다.

일반적으로 $M \times N$ 이미지는 가로 세로가 절반으로 줄어든 $M/2 \times N/2$ 해상도 이미지로 변환된다. 업샘플링은 다운샘플링과 반대로 상위 단계 이미지의 짝수열, 짝수행에 픽셀을 추가하여 하위 단계 이미지를 만든다. 하지만 생성된 하위 단계 이미지는 제대로 된 이미지로 나타나지 않고 마치 블러링 효과를 낸 듯한 이미지로 보이게 된다.

이유는 어떤 이미지의 해상도를 높이게 되면 높은 해상도에 해당하는 이미지 정보가 제대로 복원되지 않기 때문이다. OpenCV에서 가우시안 피라미드를 수행하는 함수 2가지가 있다.

- `cv2.pyrDown()` : 다운 샘플링
- `cv2.pyrUp()` : 업샘플링

16. 이미지 피라미드

▪ 16.2 이미지 피라미드란?

- 이미지 피라미드의 종류

(2) 라플라시안 피라미드

라플라시안 피라미드는 가우시안 피라미드 결과로 생성한다. 어느 한 이미지(원본 이미지)를 가우시안 피라미드의 상위 단계 이미지2 로 생성한다. 이 이미지2 를 가우시안 피라미드의 하위 단계 이미지3 으로 만든다.

원본이미지와 이미지3은 해상도가 같지만(해상도가 다를 수도 있습니다) 이미지3은 제대로 된 원본이미지로 복구된 것이 아닐 것이다. 라플라시안 피라미드의 최하위 단계는 원본이미지에서 이미지3 을 '-' 연산함으로써 생성한다.

즉, 원본이미지 - 이미지3 한 결과가 라플라시안 피라미드의 최하위 단계이고, 그 상위 단계를 같은 방식으로 하면 상위 단계의 라플라시안 피라미드가 생성된다.

16. 이미지 피라미드

- 16.3 [16-01 example.py] – 가우시안 피라미드 다운 샘플링
 - 다음은 16-01 example.py 예제의 code이다.

[그림] 16-01 example.py의 code (1/3)

[그림] 16-01 example.py의 code (2/3)

16. 이미지 피라미드

- 16.3 [16-01 example.py] – 가우시안 피라미드 업 샘플링
 - 다음은 16-01 example.py 예제의 code이다.

[그림] 16-01 example.py의 code (1/3)

[그림] 16-01 example.py의 code (2/3)

16. 이미지 피라미드

- 16.3 [16-01, 16-02 example.py] – 가우시안 피라미드 다운 , 업 샘플링
 - 다음은 16-01 example.py , 16-02 example.py 예제의 실행 결과로 나오는 화면이다.



[그림] 16-01 example.py의 실행 결과

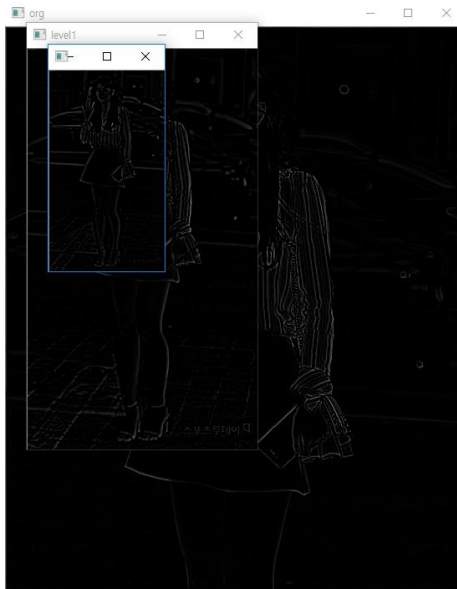


[그림] 16-02 example.py의 실행 결과

16. 이미지 피라미드

- 16.3 [16-03] – 라플라시안 피라미드

- 다음은 16-03 example.py 예제의 실행 결과로 나오는 화면이다.



[그림] 16-03 example.py의 실행 결과

16. 이미지 피라미드

▪ 16.3 [16-03 example.py] – 라플라시안 피라미드

```
>> LINE 25 ~ 26) tmp = cv2.resize(tmp1, dsize=(img_shape[i][1], img_shape[i][0]),  
                                interpolation=cv2.INTER_CUBIC)
```

- 라플 이미지를 만들 때 이용하는 함수 `cv2.subtract()` 함수는 두 개의 이미지를 '-' 연산하는 함수이다. 이 함수의 인자로 들어가는 이미지들은 그 크기가 동일해야 한다. `cv2.subtract()` 함수의 두 인자로 사용되는 이미지의 크기가 다르면 오류가 발생하고 프로그램은 동작하지 않는다. 따라서 위 예제 코드에서는 `cv2.subtract()` 함수에 들어가는 이미지 크기를 동일하게 맞추기 위해 위의 코드를 강제 삽입한 것입니다.
- 결과 화면을 보면 Canny Edge Detection의 결과와 비슷하지만 수지의 외형 모습만 매우 희미하게 흰색 픽셀로 보이는 것을 알 수 있다.

```
import numpy as np
import cv2

def pyramid():
    img = cv2.imread('images/suji.jpg', cv2.IMREAD_GRAYSCALE)
    tmp = img.copy()

    win_titles = ['org', 'level1', 'level2', 'level3']
    g_down = []
    g_down.append(tmp)
    for i in range(3):
        tmp1 = cv2.pyrDown(tmp)
        g_down.append(tmp1)
        tmp = tmp1

    for i in range(4):
        cv2.imshow(win_titles[i], g_down[i])
    cv2.waitKey(0)
    cv2.destroyAllWindows()

pyramid()
```