
워터마킹

비트 평면 분할 및 워터마킹



기타 제출일 : 2020.12.14

작성자 소속 : 산업경영시스템공학과

학번 : 2018316501

성명 : 박민지

■ 서론

이번 과제는 그냥 사진으로 보았을 때는 워터마크가 보이지 않고, 비트플레인으로 나누어 보았을 때 0번 비트플레인을 따로 검출해야 워터마크가 보이는 프로그램을 제작하였습니다. 따로 plt창에 출력하는 것이 아닌 트랙바를 설정하여 비트플레인을 조정할 수 있게 하였습니다.

■ 본론

1. 워터마크 생성

def watermark():

```
img = cv.imread(orig_img, cv.IMREAD_GRAYSCALE)
```

```
water_img = cv.imread(res_img, cv.IMREAD_GRAYSCALE)
```

```
result_img = img.copy()
```

```
row,col = img.shape
```

```
for i in range(0,row):
```

```
    for j in range(0, col):
```

```
        if water_img[i,j] == 0: # 그레이스케일 값이 0인 부분, 검은색 부분
```

```
            result_img[i,j] = img[i,j]&254
```

```
        else: # 그레이스케일 값이 0이 아닐 때, 검은색이 아니면 흰색으로 처리
```

```
            result_img[i,j] = img[i,j]|1
```

```
cv.imwrite('watermark.bmp',result_img)
```

- ➔ Img에 보여질 이미지를 읽어오고, water_img에 워터마크로 삽입할 이미지를 불러옵니다. result_img에 img를 복사한 후 row, col에 이미지의 shape를 저장해줍니다. for문을 돌려 워터마크로 삽입할 이미지의 그레이스케일 값이 0인 부분 즉, 검은색일때 원본 이미지의 최 하위 비트만 0으로 바꾸기 위해 $img[i,j] \& 254$ 로 연산을 합니다. (254의 2진수는 11111110) 상위 7비트는 원본을 그대로 유지하고 최하위 비트는 0으로 바꿉니다. 반대로 배경은 흰색으로 처리하면 되기 때문에 최하위 비트를 1로 변경합니다. (1의 2진수는 00000001) 상위 7비트는 원본 그대로 유지되고 최하위 비트는 1로 변경합니다.
- ➔ 원본 이미지에 워터마크를 삽입한 최종 이미지를 watermark.bmp로 저장합니다.

2. 비트플레인 설정

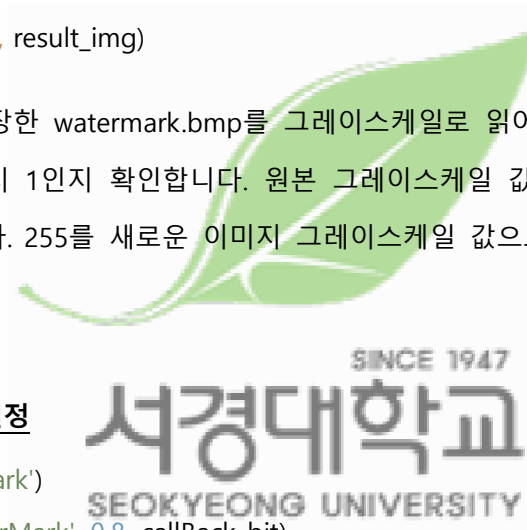
```
def callBack_bit(b):
    img = cv.imread(watermark_img, cv.IMREAD_GRAYSCALE)

    row, col = img.shape
    result_img = img.copy()

    for i in range(0, row):
        for j in range(0, col):
            result_img[i, j] = 255 if (img[i, j] & (1 << b)) else 0

    cv.imshow('WaterMark', result_img)
```

- ➔ Img에 위에서 저장한 watermark.bmp를 그레이스케일로 읽어옵니다. for문을 돌려 각 비트를 나눠서 0인지 1인지 확인합니다. 원본 그레이스케일 값과 비교해서 최상위 비트가 1이면 참이 됩니다. 255를 새로운 이미지 그레이스케일 값으로 설정하고 1이 아니면 0으로 설정합니다.



3. 트랙바 설정

```
cv.namedWindow('WaterMark')
cv.createTrackbar('bit', 'WaterMark', 0, 8, callBack_bit)
```

```
while(1):
    watermark()
    k = cv.waitKey(1)
    if k != -1:
        break
```

- ➔ 트랙바를 WaterMark로 이름을 정하고 최대 8까지 움직일 수 있도록 합니다.
- ➔ 처음 실행하게 되면 watermark 함수를 실행하고, 그 뒤 트랙바를 움직일때만 변동이 있을 수 있도록 설정합니다.

결론



➔ 실행을 하면 레나 영상을 비트플레인 7로 하였을 때의 모습이 나옵니다.



- ➔ Bit 4일때의 모습과 bit 0일때의 모습입니다. 비트플레인인 0이면 원본영상은 사라지고 워터마크로 삽입했던 이미지가 출력됩니다.

