HW#9

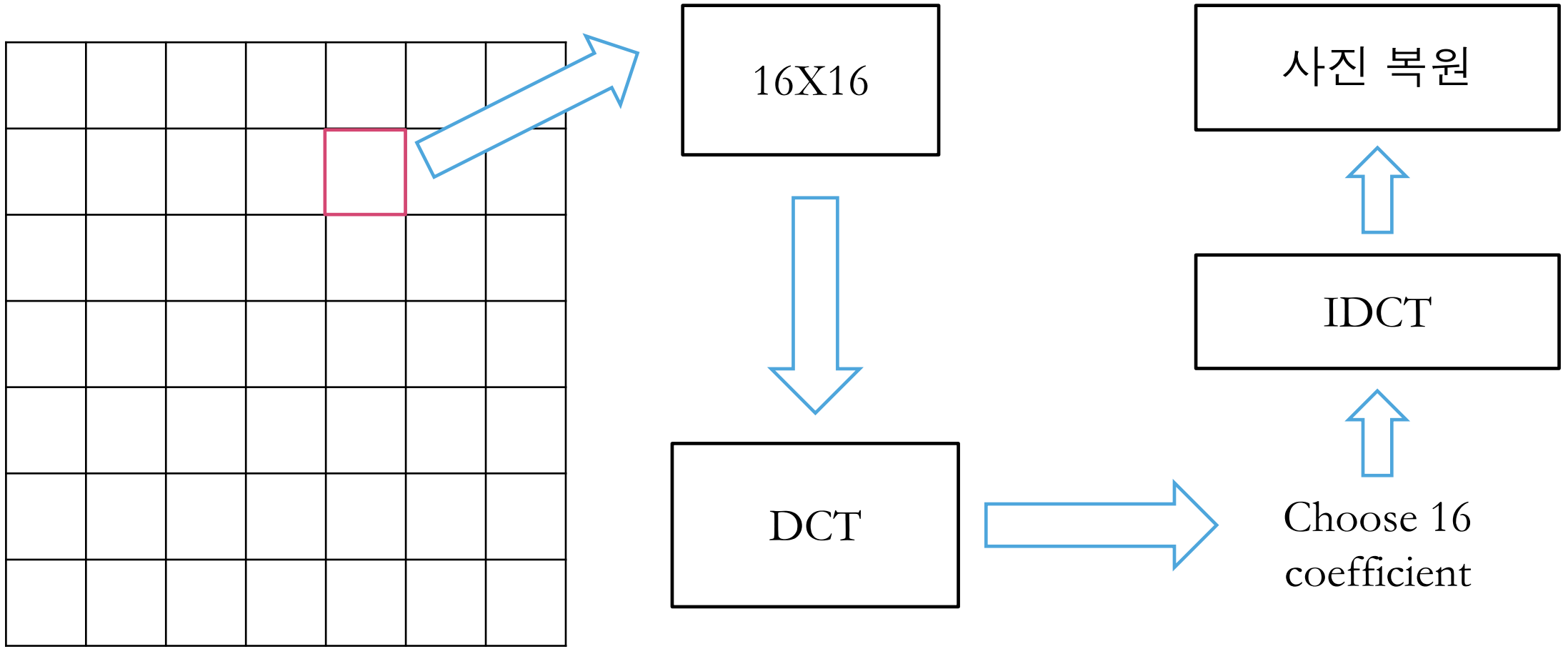
서왕규

2014004066

CONTENTS

01. Algorithm

DCT/IDCT



02. Implementation

DCT/IDCT

- **Procedure**

① 이미지를 16X16 사이즈 블록으로 나눔

② 각 블록의 RGB를 각각 저장

③ 각 블록의 R,G,B 값을 각각 DCT

④ Coeffiecient중 높은 값 16개를 선택하고, 나머지 값을 버림

⑤ 각 블록의 R,G,B 값을 IDCT(Inverse DCT)

⑥ 구한 값을 합쳐 사진을 복원

- **Implementation Enviroment**

✓ Language : python

✓ Open library `scipy.fftpack` 사용

02. Implementation

1. 이미지를 **16X16** 사이즈 블록으로 나눔

2. 각 블록의 **RGB**를 각각 저장

```
for element in testimg :
    rblock = []
    gblock = []
    bblock = []
    for i in range(0,int(len(element)/16)) :
        for j in range(0,int(len(element[0])/16)) :
            rblock.append(element[16*i:16*i+16,16*j:16*j+16,0].copy())
            gblock.append(element[16*i:16*i+16,16*j:16*j+16,1].copy())
            bblock.append(element[16*i:16*i+16,16*j:16*j+16,2].copy())
    block.append([rblock, gblock, bblock])
```

3. 각 블록의 **R,G,B** 값을 각각 **DCT**

4. Coeffiecient중 높은 값 **16**개를 선택하고**,** 나머지 값을 버림

```
def dct2(block):
    A = dct(dct(block.T, norm='ortho').T, norm='ortho')
    num = np.abs(A.flatten())
    num.sort()
    np.place(A, abs(A) < num[-16], 0)
    return A
```

02. Implementation

5. 각 블록의 R,G,B 값을 IDCT(Inverse DCT)

```
def idct2(block):
    return idct(idct(block.T, norm='ortho').T, norm='ortho')
```

6. 구한 값을 합쳐 사진을 복원

```
for i in range(0,heightBlockNum) :
    tmp = rblock[widthBlockNum * i]
    tmp2 = gblock[widthBlockNum * i]
    tmp3 = bblock[widthBlockNum * i]
    for j in range(1,widthBlockNum):
        tmp = np.hstack((tmp,rblock[widthBlockNum*i + j]))
        tmp2 = np.hstack((tmp2,gblock[widthBlockNum*i + j]))
        tmp3 = np.hstack((tmp3,bblock[widthBlockNum*i + j]))
    if i == 0 :
        rfinal = tmp
        gfinal = tmp2
        bfinal = tmp3
    else :
        rfinal = np.vstack((rfinal,tmp))
        gfinal = np.vstack((gfinal,tmp2))
        bfinal = np.vstack((bfinal,tmp3))
```

```
[[33 33 33 33 33 34 34 34 34 34 34 35 37 39 40 39]
 [33 33 33 33 33 34 34 34 34 34 34 35 37 39 40 39]
 [33 33 33 33 33 33 34 34 34 34 35 35 37 39 40 39]
 [33 34 33 33 33 33 34 34 34 35 35 36 38 40 40 40]
 [34 34 33 33 33 33 33 34 35 35 36 37 38 40 41 40]
 [34 34 33 33 33 33 33 34 35 36 36 37 39 41 41 40]
 [34 34 33 33 33 33 33 34 36 36 37 38 39 41 41 41]
 [34 34 33 33 33 32 33 34 36 37 37 38 40 41 41 41]
 [35 34 33 33 33 32 33 34 36 37 38 38 40 41 41 41]
 [35 34 33 33 33 32 32 34 36 37 38 38 40 41 41 41]
 [35 34 33 33 33 32 32 34 36 37 38 38 40 41 41 41]
 [35 34 33 33 33 32 32 34 36 37 37 38 40 41 41 41]
 [35 33 33 33 33 32 32 34 36 37 37 38 40 41 41 41]
 [35 33 33 33 33 32 32 34 36 37 37 38 39 41 41 41]
 [35 33 33 33 33 32 32 34 36 37 37 38 39 40 41 40]
 [35 33 33 33 33 32 32 34 36 37 37 38 39 40 41 40]]
```

IDCT로 복원한 Block

03. Result

Image 1

원본

복원

03. Result

Image 1

- 표현이 간단한 부분은 원본과 구분안될 정도로 차이 없이 복원

- 표현이 복잡한 부분은 사각형 부분의 표시가 선명
- 원본과 일치하지 않는 부분 발생

03. Result

Image 2

원본

복원

03. Result

Image 3

원본

복원