Computer Vision
HW10

R10921104 王怡堯

December 13, 2021

1 Primary Procedures

1.1 My Convolution

```python
def my_conv(lena,mask,theshold):
    m=len(lena)-len(mask)+1
    n=len(lena[0])-len(mask[0])+1
    mk=len(mask)
    nk=len(mask[0])
    res=[[0]*n for _ in range(m)]
    for i in range(m):
        for j in range(n):
            tmp=0
            for k in range(mk):
                for l in range(nk):
                    tmp+=mask[k][l]*lena[i+k][j+l]
            if tmp>=theshold:
                res[i][j]=1
            elif tmp<=-theshold:
                res[i][j]=-1

    return res
```

1.2 Laplace Mask1

```python
def Laplace1(lena,theshold):
    m=len(lena)
    n=len(lena[0])
    mask=[[0, 1, 0],
        [1, -4, 1],
        [0, 1, 0]]
    lena=expan(lena)
    res1=my_conv(lena,mask,theshold)
    res1=expan(res1)
    res2=[[1]*m for _ in range(n)]
    for i in range(m):
        for j in range(n):
            if res1[i+1][j+1]==1:
                tmp=False
                for k in range(3):
                    for l in range(3):
                        if res1[i+k][j+l]==-1:
                            tmp=True
                            break
                    if tmp:
                        break
                if tmp:
                    res2[i][j]=0
    return np.array(res2)*255
```

1.3 Laplace Mask2

```
def Laplace2(lena,theshold):
    m=len(lena)
    n=len(lena[0])
    mask=[[1, 1, 1],
        [1, -8, 1],
        [1, 1, 1]]
    lena=expan(lena)
    res1=my_conv(lena,mask,theshold)
    res1=expan(res1)
    res2=[[1]*m for _ in range(n)]
    for i in range(m):
        for j in range(n):
            if res1[i+1][j+1]==1:
                tmp=False
                for k in range(3):
                    for l in range(3):
                        if res1[i+k][j+l]==-1:
                            tmp=True
                            break
                    if tmp:
                        break
                if tmp:
                    res2[i][j]=0
    return np.array(res2)*255
```

1.4 Minimum variance Laplacian

```
def Minimum_Variance_Laplacian(lena,theshold):
    m=len(lena)
    n=len(lena[0])
    mask=[[2, -1, 2],
        [-1, -4, -1],
        [2, -1, 2]]
    lena=expan(lena)
    res1=my_conv(lena,mask,theshold)
    res1=expan(res1)
    res2=[[1]*m for _ in range(n)]
    for i in range(m):
        for j in range(n):
            if res1[i+1][j+1]==1:
                tmp=False
                for k in range(3):
                    for l in range(3):
                        if res1[i+k][j+l]==-1:
                            tmp=True
                            break
                    if tmp:
                        break
                if tmp:
                    res2[i][j]=0
    return np.array(res2)*255
```

1.5 Laplace of Gaussian

```
def Laplace_of_Gaussian(lena,theshold):
    m=len(lena)
    n=len(lena[0])
    mask=[[0, 0, 0, -1, -1, -2, -1, -1, 0, 0, 0],
[0, 0, -2, -4, -8, -9, -8, -4, -2, 0, 0],
[0, -2, -7, -15, -22, -23, -22, -15, -7, -2, 0],
[-1, -4, -15, -24, -14, -1, -14, -24, -15, -4, -1],
[-1, -8, -22, -14, 52, 103, 52, -14, -22, -8, -1],
[-2, -9, -23, -1, 103, 178, 103, -1, -23, -9, -2],
[-1, -8, -22, -14, 52, 103, 52, -14, -22, -8, -1],
[-1, -4, -15, -24, -14, -1, -14, -24, -15, -4, -1],
[0, -2, -7, -15, -22, -23, -22, -15, -7, -2, 0],
[0, 0, -2, -4, -8, -9, -8, -4, -2, 0, 0],
[0, 0, 0, -1, -1, -2, -1, -1, 0, 0, 0]]
    for i in range(5):
        lena=expan(lena)
    res1=my_conv(lena,mask,theshold)
    res1=expan(res1)
    res2=[[1]*m for _ in range(n)]
    for i in range(m):
        for j in range(n):
            if res1[i+1][j+1]==1:
                tmp=False
                for k in range(3):
                    for l in range(3):
                        if res1[i+k][j+l]==-1:
                            tmp=True
                            break
                    if tmp:
                        break
                if tmp:
                    res2[i][j]=0
    return np.array(res2)*255
```

1.6 Difference of Gaussian

```
def Difference_of_Gaussian(lena,theshold):
    m=len(lena)
    n=len(lena[0])
    mask=[[-1, -3, -4, -6, -7, -8, -7, -6, -4, -3, -1],
        [-3, -5, -8, -11, -13, -13, -13, -11, -8, -5, -3],
        [-4, -8, -12, -16, -17, -17, -17, -16, -12, -8, -4],
        [-6, -11, -16, -16, 0, 15, 0, -16, -16, -11, -6],
        [-7, -13, -17, 0, 85, 160, 85, 0, -17, -13, -7],
        [-8, -13, -17, 15, 160, 283, 160, 15, -17, -13, -8],
        [-7, -13, -17, 0, 85, 160, 85, 0, -17, -13, -7],
        [-6, -11, -16, -16, 0, 15, 0, -16, -16, -11, -6],
        [-4, -8, -12, -16, -17, -17, -17, -16, -12, -8, -4],
        [-3, -5, -8, -11, -13, -13, -13, -11, -8, -5, -3],
        [-1, -3, -4, -6, -7, -8, -7, -6, -4, -3, -1]]
    for i in range(5):
        lena=expan(lena)
    res1=my_conv(lena,mask,theshold)
    res1=expan(res1)
    res2=[[1]*m for _ in range(n)]
    for i in range(m):
        for j in range(n):
            if res1[i+1][j+1]==1:
                tmp=False
                for k in range(3):
                    for l in range(3):
                        if res1[i+k][j+l]==-1:
                            tmp=True
                            break
                    if tmp:
                        break
                if tmp:
                    res2[i][j]=0
    return np.array(res2)*255
```

2 Result

thresholds:15

thresholds:15

thresholds:10

thresholds:3000

thresholds:1