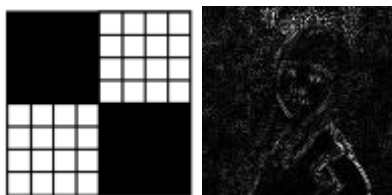


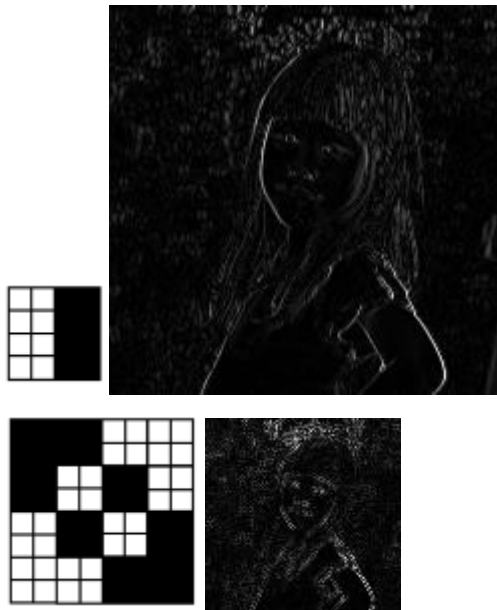
# 小波变换作业

计 54 骆轩源 2015011340

## A1 (采用不同的 Harry 小波基)



## A1 选做（新小波基）



A1 实现代码（用 python numpy 以及 opencv 实现）：

```
A1.py
1 import numpy as np
2 import cv2
3
4
5 def guiyi(img,r1,c1,r2,c2):
6     Max = 0.0
7     for r in range(r1,r2):
8         for c in range(c1,c2):
9             if img[r][c] > Max:
10                 Max = img[r][c]
11     for r in range(r1,r2):
12         for c in range(c1,c2):
13             img[r][c] = img[r][c]/Max * 255.0
14     return img
15
16 def getCore88():
17     core88 = -np.ones([8,8])
18     for i in range(0,4):
19         for j in range(0,4):
20             core88[i][j] = 1
21     for i in range(4,8):
22         for j in range(4,8):
23             core88[i][j] = 1
24     return core88
25
```

```
def harr1(img, core):
    m = core.shape[0]
    step = m//2
    outimg = np.zeros([n//step - 1, n//step - 1])
    for i in range(0, n//step - 1):
        for j in range(0, n//step - 1):
            sx = i * step
            sy = j * step
            outimg[i][j] = 0
            for x in range(0, m):
                for y in range(0, m):
                    outimg[i][j] += img[sx + x][sy + y] * core[x][y]
    outimg = np.abs(outimg)
    outimg = guiyi(outimg, 0, 0, n//step-1, n//step-1)
    return outimg
```

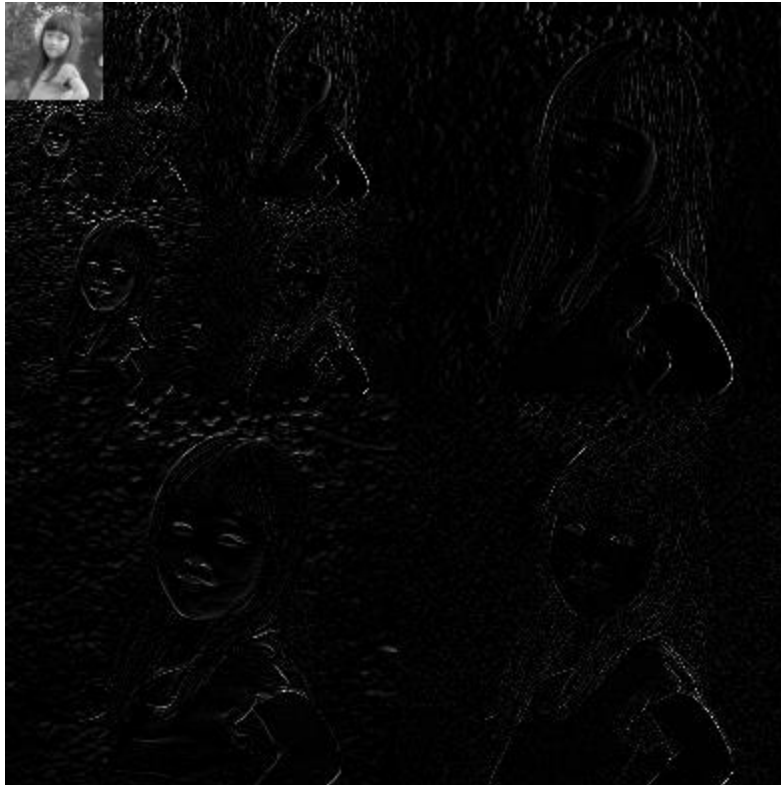
```
img = cv2.cvtColor(cv2.imread('origin.bmp'), cv2.COLOR_BGR2GRAY).astype('float32')
#print img

n = img.shape[0]
core22 = np.array([[1, -1], [-1, 1]])
core44 = np.array([[-1, -1, 1, 1], [-1, -1, 1, 1], [1, 1, -1, -1], [1, 1, -1, -1]])
core44New = np.array([[1, 1, -1, -1], [1, 1, -1, -1], [1, 1, -1, -1], [1, 1, -1, -1]])
core88 = getCore88()
core88New = np.array([
    [1, 1, 1, 1, -1, -1, -1, -1],
    [1, 1, 1, 1, -1, -1, -1, -1],
    [1, 1, -1, -1, 1, 1, -1, -1],
    [1, 1, -1, -1, 1, 1, -1, -1],
    [-1, -1, 1, 1, -1, -1, 1, 1],
    [-1, -1, 1, 1, -1, -1, 1, 1],
    [-1, -1, -1, -1, 1, 1, 1, 1],
    [-1, -1, -1, -1, 1, 1, 1, 1]
])
img22 = harr1(img, core22)
img44 = harr1(img, core44)
img88 = harr1(img, core88)
img88New = harr1(img, core88New)
img44New = harr1(img, core44New)

cv2.imwrite('core22.bmp', img22)
cv2.imwrite('core44.bmp', img44)
cv2.imwrite('core88.bmp', img88)
cv2.imwrite('core44New.bmp', img44New)
cv2.imwrite('core88New.bmp', img88New)
```

也即变换核中，白色为-1，黑色为1，做变换之后取绝对值,并归一化乘以 255。

## A2 (三层离散小波变换)



采用的是 PPT 上的做法，一开始先对所有的 Row，左边一半存均值，右边一半存差分，再对每个 Column 做同样的事情。就得到了一层。如此重复三次，就得到了 3 层离散小波变换。

对变换后的矩阵，每张小图分别**取绝对值**，并归一化乘以 255。

## A3 实现代码：

```
import numpy as np
import cv2

def harrRow(img, m):
    n = img.shape[0]
    newImg = np.zeros_like(img)
    for i in range(0, n):
        for j in range(0, n):
            newImg[i][j] = img[i][j]
    for i in range(0, m):
        for j in range(0, m/2):
            newImg[i][j] = (img[i][2*j] + img[i][2 * j + 1]) * 0.5
        for j in range(0, m/2):
            newImg[i][j + m/2] = (img[i][2*j] - img[i][2 * j + 1]) * 0.5
    return newImg
```

```
def harrCol(img, m):
    newImg = np.transpose(img)
    newImg = harrRow(newImg, m)
    newImg = np.transpose(newImg)
    return newImg
```

```

def guiyi(img,r1,c1,r2,c2):
    Max = 0.0
    for r in range(r1,r2):
        for c in range(c1,c2):
            if img[r][c] > Max:
                Max = img[r][c]
    for r in range(r1,r2):
        for c in range(c1,c2):
            img[r][c] = img[r][c]/Max * 255.0
    return img

```

```

img = cv2.cvtColor(cv2.imread('origin.bmp'),cv2.COLOR_BGR2GRAY).astype('float32')
#print img

n = img.shape[0]
img = harrRow(img,n)
img = harrCol(img,n)
img = harrRow(img,n/2)
img = harrCol(img,n/2)
img = harrRow(img,n/4)
img = harrCol(img,n/4)

cv2.imwrite('Harr4.bmp', img)
absimg = np.abs(img)
cv2.imwrite('Harr4(abs).bmp', absimg)

img = guiyi(img,n/2,n/2,n,n)
img = guiyi(img,0,n/2,n/2,n)
img = guiyi(img,n/2,0,n,n/2)

img = guiyi(img,0,n/4,n/4,n/2)
img = guiyi(img,n/4,0,n/2,n/4)
img = guiyi(img,n/4,n/4,n/2,n/2)

img = guiyi(img,0,n/8,n/8,n/4)
img = guiyi(img,n/8,0,n/4,n/8)
img = guiyi(img,n/8,n/8,n/4,n/4)

cv2.imwrite('Harr(abs-guiyi).bmp',img)

```