

Computer and Robot Vision

Homework#6

R01944040 柳成蔭

這次的作業是對原圖進行 down sample，然後計算得到 Yokoi connectivity number。

我使用 VS2012 編寫程式

先將 binary 的 Lena 圖從 512x512 Downsample 到 64x64: 用 8x8 的 block 作為一個 unit, 選左上的 pixel 作為新 64x64 圖的 pixel 值。

```
//downsample
Mat imgDownSample(66,66,CV_8UC1,Scalar(0)); //the
boundary is zero
for(int i=1; i<=imgDownSample.rows-2; i++)
{
    for(int j=1; j<=imgDownSample.cols-2; j++)
    {
        imgDownSample.at<uchar>(i,j)=imgBinary.at<uchar>(8*(i-
1),8*(j-1));
    }
}
```

Down sample result:



(a) Formula h

- 4-connectivity

$$h(b, c, d, e) = \begin{cases} q & \text{if } b = c \text{ and } (d \neq b \text{ or } e \neq b) \\ r & \text{if } b = c \text{ and } (d = b \text{ and } e = b) \\ s & \text{if } b \neq c \end{cases}$$

- q : corner $1 \rightarrow 0$ transition
- r : corner all 1, no transition
- s : center 1, neighbor 0, nothing will happen

```
int h(int b, int c, int d, int e)
{
    if( b==c && (d!=b || e!=b) )
        return q;
    else if( b==c && (d==b && c==b) )
        return r;
    else if(b!=c)
        return s;
    else
        return -1;
}
```

(b) Formula f

$$f(a_1, a_2, a_3, a_4) = \begin{cases} 5 & \text{if } a_1 = a_2 = a_3 = a_4 = r \\ n & \text{where } n = \#\{a_k | a_k = q\}, \text{ otherwise} \end{cases}$$

- 5: no transition all 8 neighbors 1, thus interior
- n : 1 transition generates one connected component if center removed

```
int f(int a1, int a2, int a3, int a4)
{
    if(a1==r && a2==r && a3==r && a4==r)
        return 5;
    else
    {
        int n=0;
        if(a1==q)
```

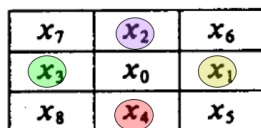
```

        n++;
        if(a2==q)
            n++;
        if(a3==q)
            n++;
        if(a4==q)
            n++;
        return n;
    }
}

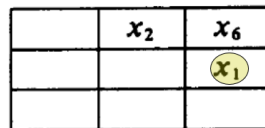
```

(c) Yokoi number

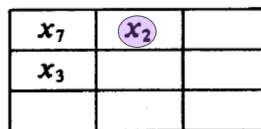
corner neighborhood



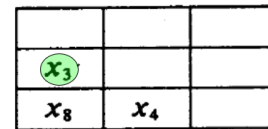
(a)



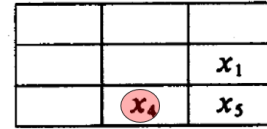
(b)



(c)



(d)



(e)

4- Connectivity number

$$y = f(a_1, a_2, a_3, a_4)$$

$$a_1 = h(x_0, x_1, x_6, x_2)$$

$$a_2 = h(x_0, x_2, x_7, x_3)$$

$$a_3 = h(x_0, x_3, x_8, x_4)$$

$$a_4 = h(x_0, x_4, x_5, x_1)$$

用Formula h來統計四個corner的連通值，用Formula f判斷中間點的Yokoi number。

```

int yokoi[64][64]={6};
int rows=64;
int cols=64;
for(int i=0; i<=rows-1; i++)

```

```

{
    for(int j=0; j<=cols-1; j++)
    {
        if(imgDownSample.at<uchar>(i+1,j+1)==255)
        {
            int x[9];
            x[0]=imgDownSample.at<uchar>(i+1,j+1);
            x[1]=imgDownSample.at<uchar>(i+1,j+2);
            x[2]=imgDownSample.at<uchar>(i,j+1);
            x[3]=imgDownSample.at<uchar>(i+1,j);
            x[4]=imgDownSample.at<uchar>(i+2,j+1);
            x[5]=imgDownSample.at<uchar>(i+2,j+2);
            x[6]=imgDownSample.at<uchar>(i,j+2);
            x[7]=imgDownSample.at<uchar>(i,j);
            x[8]=imgDownSample.at<uchar>(i+2,j);
            int a1=h(x[0],x[1],x[6],x[2]);
            int a2=h(x[0],x[2],x[7],x[3]);
            int a3=h(x[0],x[3],x[8],x[4]);
            int a4=h(x[0],x[4],x[5],x[1]);
            yokoi[i][j]=f(a1,a2,a3,a4);
            fout << yokoi[i][j];
        }
        else
            fout << " ";
    }
    fout << "\n";
}

```

最後將得到的 Yokoi number 記錄下來，結果單獨附在下頁。

11111111	12111111111122322221	111111111111	0 0
15555551	11555555511 2 11 11	115555555511	0
15555551	1 211555112 21112221	15555555551	21
15555551	1 2 155112 2221511	155555555511	1
15555551	22 2112 22 121 0 0	1555555555511	0
15555551	1 2 21 2 1 1	1555555555551	0
15555551	12 1 121111 1321	15555555555511	
15111551	1322 1155551111	15555555555551	
111 1551	1 12155555511	155555555555511	
11 1551	21155555511	1551155555511	
21 1551	2 15555555111	1551 11555511	
1 1551	2 155555555511	1551 115551	1
1551	112115555555551	1551 15511	12
1551	1555555555555511	1551 1111	111
1551	1 222115555555555511	1151 11	1151
1551	2 22 1 1555555555555511	151 11111	1551
1551	2 1 11555555555555551	151 115551	11551
1551	2 1155555555555555111511155511		115551
1551	12 115555555555555555555555551		155551
1551	11 0 2215555555555555555555555112		115551
1551	111 22 15555555555555555555551	1	155551
1551	1511 1 12511211111211155555555111		1155551
1551	15521 1 121 1 11 1 1555555111	0	1555551
1551	1151 132 2 1155555111	0	11555551
1551	151 0 322 115555111	121	15555551
1551	1221 2 1555551	131	115555551
1551	2 0 1 115555511	1	115555551
1551	2 0 0 1155555551	0	1 15555551
1551	2 1155555551		2115555551
1551	1 0 11555555551		1555555551
1551	1 11511115555521	1	11555555551
1551	1 1 11111 1155511	2	15555555551
1551	131 111 15111	2	15555555551
1551	121 0 1121 1 111 1	2	115555555551
1551	11 111 1 221 11 1	2	155555555551
1551	12 0 1 21 121 11 1111	2	155555555551
1551	1 12 22 151111111551	2	1155555555551
1551	1 2 1555551115511	1	1555555555551
1551	2 0 0 22 12555551 15551	1	1555555555551
1551	1 1555511 11511	2	11555555555551
1551	0 0 21 155551 1 151	2	15555555555551
1551	2 15555112 151	2	155555555555551
1551	1 1 1 115555511111	2	155555555555551
1551	2 22 111511111212	21155555555555551	
1551	0 1 12 151 2 1	15555555111555551	
1551	0 0 0 1111 121	155555551 1555551	
1551	0 1111111	155555551 1555551	
1551	0 115551	155555551 1555511	
1551	15551	211111111 155511	
11521	1 12 122155511	2 11 115511	
1 151 0	1 15555111	2111 15511	
22 1511	1 15555555111	155111 1511	
22 1511	1 15555555551	155551 1151	
2 151	0 1 11155555555511	155511 1511	
2 1521	0 1 155555555555511	15551 12151	
2 151	121 155555555555551	155511 1551	
2 1511	0 1555555555555551	115551 1511	
21 1511	11 1555555555555551	111111151	
11 151	0 11555555555555511	111511	
11 151	15555555555555551	151	
11 151	0 11555555555555551	211	
11 151	115555555555555511	1	
11 151	0 15555555555555551		
11 111	0 1211111111111111111		