图像处理与机器学习

基础实验

实验六 图像边缘提取

➢ 实验数据

-- 源图像：CMU等大学Face Detection/Recognition 收集的公开数据库

-- 格式：*.raw

FILE1.raw	FILE10.raw	FILE11.raw	FILE12.raw	FILE13.raw
FILE14.raw	FILE15.raw	FILE16.raw	FILE17.raw	FILE18.raw
FILE19.raw	FILE2.raw	FILE20.raw	FILE21.raw	FILE22.raw
FILE23.raw	FILE24.raw	FILE25.raw	FILE26.raw	FILE27.raw

➢ **基于一阶差分的图像增强**

$$G_x = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

$$G_y = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

Sobel 算子

$$y(j,i) = \sum_m \sum_n h(m,n) x(j+m,i+n)$$

边缘提取

-1	0	1
-2	0	2
-1	0	1

$h(m,n)=$

-1	-2	-1
0	0	0
1	2	1

实验六 图像边缘提取


```
//边缘提取
int sobelFlag;
int* sob_x, * sob_y;
BYTE* outSobelx, * outSobely;
void sobel(BYTE*, int, int, int*, int*);
void tranByte(int*, BYTE*, int, int);
```

```cpp
void CMFCApplication1View::sobel(BYTE* window, int wid, int hei, int* sob_x, int* sob_y)
{
    int so_x[9];//horizontal          int i, j, m, n;
    so_x[0] = -1;                     BYTE block[9];
    so_x[1] = 0;
    so_x[2] = 1;
    so_x[3] = -2;                     int value;
    so_x[4] = 0;                      for (i = 0; i < hei; i++)
    so_x[5] = 2;                          for (j = 0; j < wid; j++)
    so_x[6] = -1;                         {
    so_x[7] = 0;                              if (i == 0 || j == 0 || i == hei - 1 || j == wid - 1)
    so_x[8] = 1;                              {

                                                  sob_x[i * wid + j] = 0;
                                                  sob_y[i * wid + j] = 0;

    int so_y[9];                              }
    so_y[0] = -1;
    so_y[1] = -2;
    so_y[2] = -1;
    so_y[3] = 0;                              else
    so_y[4] = 0;                              {
    so_y[5] = 0;
    so_y[6] = 1;                                  //pick up 3*3 block
    so_y[7] = 2;                                  for (m = -1; m < 2; m++)
    so_y[8] = 1;                                      for (n = -1; n < 2; n++)
                                                          block[(m + 1) * 3 + n + 1] = window[(i + m) * wid + j + n];
                                              value = convolution(so_x, block);
                                              sob_x[i * wid + j] = value;

                                              value = convolution(so_y, block);
                                              sob_y[i * wid + j] = value;

                                          }
                                      }
}
```

```cpp
int CMFCApplication1View::convolution(int* operatr, BYTE* block)
{
    int value;
    int i, j;
    value = 0;
    for (i = 0; i < 3; i++)
        for (j = 0; j < 3; j++)
            value += operatr[i * 3 + j] * block[i * 3 + j];

    return value;

}
```
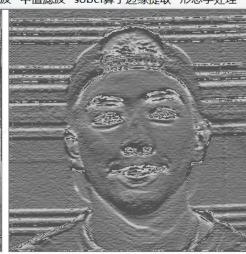
实验六 图像边缘提取

谢　谢

　　本课程所引用的一些素材为主讲老师多年的教学积累，来源于多种媒体及同事和同行的交流，难以一一注明出处，特此说明并表示感谢！