

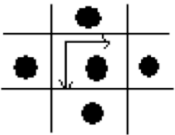
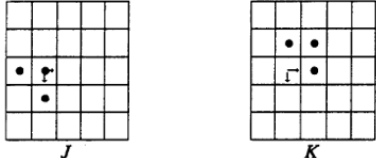
電腦視覺作業四

Computer Vision HW4
R04525092 工科碩二 鄭力文

使用語言 C++ with openCV

Write programs which do binary morphological dilation, erosion, opening, closing, and hit-and-miss transform on a binary image

Kernel use:

Kernel	Hit-and-Miss Kernel
	

1. Dilation

膨脹將每一點透過 kernel 進行運算，該點中心吻合 kernel 的周邊為 255。

```
//進行膨脹
void dilation(int x,int y,Mat im,Mat newIm){
    if(im.at<Vec3b>(x,y)[0] == 255)
    {
        for(int j=-1;j<2;j++){
            for(int i=-1;i<2;i++){
                if(kernel[i][j]==255){
                    for(int z=0;z<3;z++){
                        newIm.at<Vec3b>(x+i,y+j)[z] = 255;
                    }
                }
            }
        }
    }
}
```



2. Erosion

```
//消退
int erosion(int x,int y,Mat im){
    int centerX=1;
    int centerY=1;
    int count=0;
    int kn[3][3]={0};
    memcpy ( &kn, &kernel, sizeof(kernel) );
    if(x!=0||y!=0){
        if(int(im.at<Vec3b>(x,y-1)[0])==kn[centerX][centerY-1]){
            count++;
        }
        if(int(im.at<Vec3b>(x-1,y)[0])==kn[centerX-1][centerY]){
            count++;
        }
        if(int(im.at<Vec3b>(x,y)[0])==kn[centerX][centerY]){
            count++;
        }
        if(int(im.at<Vec3b>(x+1,y)[0])==kn[centerX+1][centerY]){
            count++;
        }

        if(int(im.at<Vec3b>(x,y+1)[0])==kn[centerX][centerY+1]){
            count++;
        }
    }
    if(count>=5)
        return 255;
    else
        return 0;
}
```

消退將每一點運算是否符合 kernel 若符合則將中心賦予值 255。



3. Opening

先消退再膨脹



4. Closing

先膨脹再消退



5. Hit and Miss

```
// J 對 img 侵蝕  
// K 對 img" 侵蝕  
// 再做交集
```

