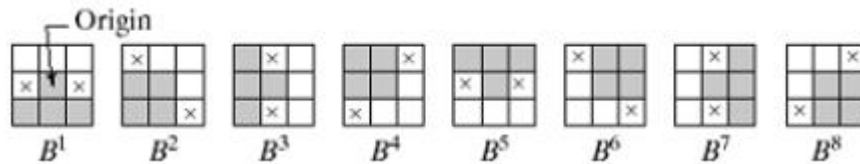


1. 首先读取图像，并使用 Otsu 算法处理成二值图像。
2. 生成如下图所示的 8 个结构元素 B。这 8 个结构元素分别代表了八个方向。



3. 循环地利用这 8 个结构元素对原图进行 thinning 操作，进行足够多的循环。
- 3.1. thinning 的实现：

$$A \otimes B = A - (A \circledast B) \\ = A \cap (A \circledast B)^c.$$

利用如图所示的公式，从代码上进行实现。

```
void thinning(IplImage *src, IplImage *dst, int stNo) {
    //cout << "Entering thinning\n";
    IplImage *con = cvCreateImage(cvGetSize(src), 8, 1);
    IplImage *temp = cvCreateImage(cvGetSize(src), 8, 1);

    hitOrMiss(src, temp, stNo);
    cvBitNot(temp, con);
    cvBitAnd(src, con, dst);
    cvReleaseImage(&con);
    cvReleaseImage(&temp);
    //cout << "Leaving thinning\n";
    return;
}
```

其中, cvBitNot 和 cvBitAnd 是我自己实现的以像素为单位的取反操作和取交集操作，在之后会讲述。首先计算括号中 A 和 B 的 hit or miss，然后再对结果取反，最后和原图像取交集即可。

- 3.2. hit-or-miss 算法实现：

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2).$$

利用如图所示的公式，从代码上进行实现。

```

void hitOrMiss(IplImage *src, IplImage *dst, int stNo) {
    //cout << "Entering hitOrMiss\n";
    IplImage *conSrc = cvCreateImage(cvGetSize(src), 8, 1);
    IplImage *temp1 = cvCreateImage(cvGetSize(src), 8, 1);
    IplImage *temp2 = cvCreateImage(cvGetSize(src), 8, 1);

    cvBitNot(src, conSrc);

    erode(src, temp1, stNo);
    erode(conSrc, temp2, CON(stNo));

    cvBitAnd(temp1, temp2, dst);
    cvReleaseImage(&conSrc);
    cvReleaseImage(&temp1);
    cvReleaseImage(&temp2);
    //cout << "Leaving hitOrMiss\n";
    return;
}

```

也是照着公式一步一步实现。首先用 B1 和 B2 分别对原图和原图的反图进行腐蚀（erode）操作，然后再取它们的交集。

至于腐蚀操作，在之前的作业中已经实现过，此处略过不表。

3.3. 二值图像逻辑运算 and, or, not

只需要对每个点进行一次逻辑运算即可，此处要注意的是，虽然颜色为 255 时是白点，可是此时仍然是需要保留的点（即数据为 255 时为二值图中的 1）。

3.4. thickening 算法实现（非骨架提取需要）：

$$A \odot B = A \cup (A * B)$$

与之前一样，按照公式用代码实现即可。

```

void thickening(IplImage *src, IplImage *dst, int stNo) {
    IplImage *temp = cvCreateImage(cvGetSize(src), 8, 1);

    hitOrMiss(src, temp, stNo);
    cvBitOr(src, temp, dst);
    cvReleaseImage(&temp);
    cvNamedWindow("img", 1);
    cvShowImage("img", dst);
    cvWaitKey(-1);
    cvDestroyWindow("img");
    return;
}

```

- 最后，显示出处理完的图片即可。我在代码中进行了改良，使之可以显示出中间所有的步骤，可以直观地看到 thinning 的过程。
- 环境：VS2013 Ultimate，OpenCV 3.0.0