

第四章



目标检测

李静



三、目标检测方法

(二) 特征明显的目标检测——基于特征匹配的目标检测

根据图像间的相关信息来检测目标, 即在两幅或者多幅不同视点的图像中寻找同一目标的特征。

➤ 颜色特征

- RGB颜色空间
- HSV颜色空间

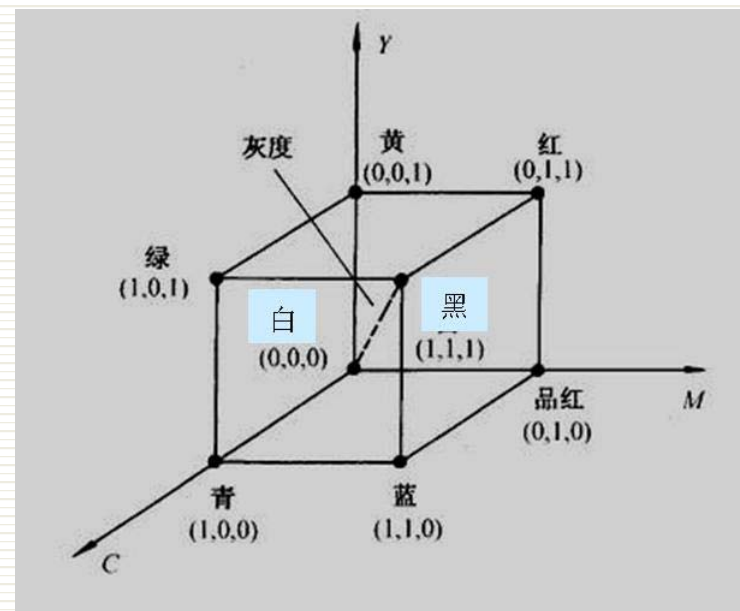
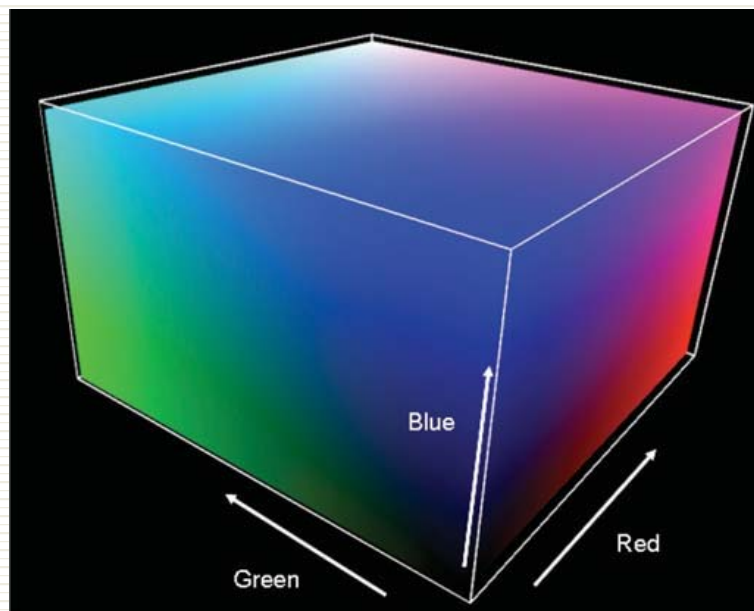
➤ 纹理特征

- LBP (Local Binary Pattern, 局部二值模式)
- Gabor滤波器

三、目标检测方法

1、颜色特征——（1）RGB颜色空间（三维立方体）

- 任意色光F都可以用R、G、B三色不同分量的相加混合而成；
- 当三基色分量都为0（最弱）时混合为黑色光，当三基色分量都为k（最强）时混合为白色光。

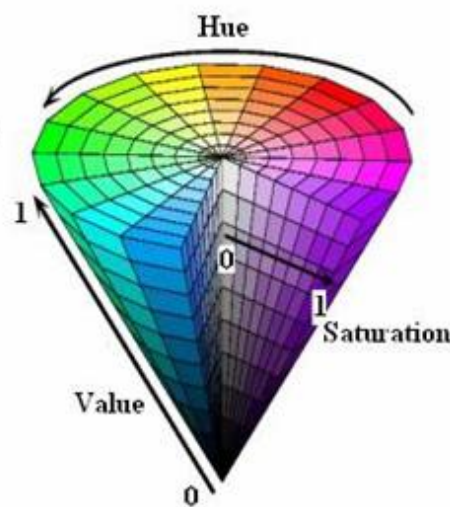
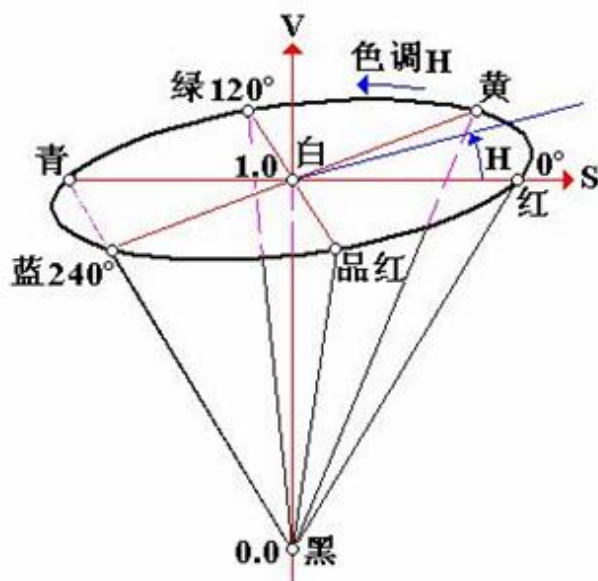




三、目标检测方法

1、颜色特征——（2）HSV颜色空间（圆锥空间模型）

- 每一种颜色都是由色调（Hue，简H），饱和度（Saturation，简S）和色明度（Value，简V）表示。
- 色调H由绕V轴的旋转角给定。红色对应于角度 0° ，绿色对应于角度 120° ，蓝色对应于角度 240° 。
- 饱和度S取值从0到1，所以圆锥顶面的半径为1。



$$R' = R / 255$$

$$G' = G / 255$$

$$B' = B / 255$$

$$C_{\max} = \max(R', G', B')$$

$$C_{\min} = \min(R', G', B')$$

$$\Delta = C_{\max} - C_{\min}$$

H 计算：

Hue

$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G' - B'}{\Delta} + 0 \right) & C_{\max} = R' \\ 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2 \right) & C_{\max} = G' \\ 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4 \right) & C_{\max} = B' \end{cases}$$

S 计算：

Saturation

$$S = \begin{cases} 0 & C_{\max} = 0 \\ \frac{\Delta}{C_{\max}} & C_{\max} \neq 0 \end{cases}$$

V 计算：

Value

$$V = C_{\max}$$



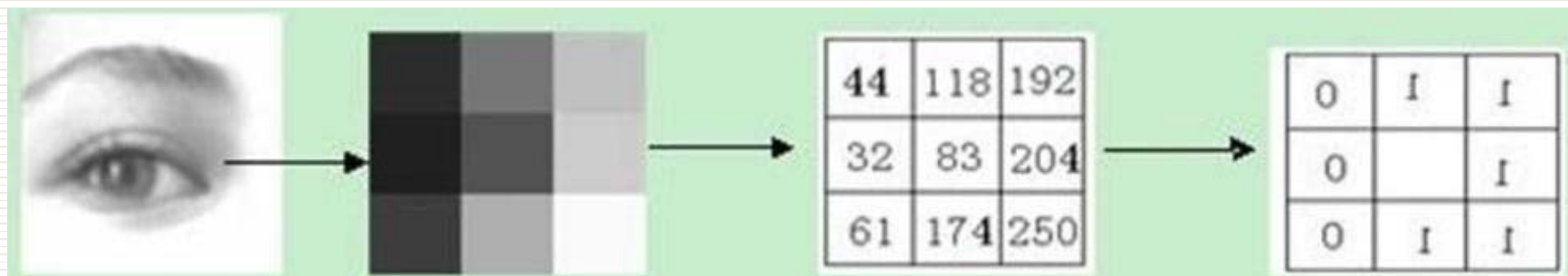
三、目标检测方法

2、纹理特征——（1）LBP

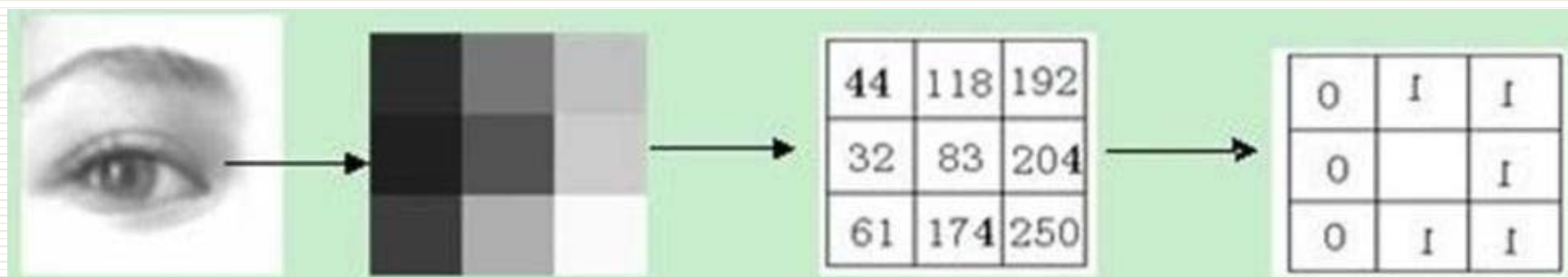
- LBP (Local Binary Pattern, 局部二值模式) , 是一种用来描述图像局部纹理特征的算子;

LBP原理:

- LBP算子定义在一个 3×3 的窗口内, 以窗口中心像素为阈值, 与相邻的8个像素的灰度值比较, 若周围的像素值大于中心像素值, 则该位置被标记为1; 否则标记为0。
- 如此可以得到一个8位二进制数 (通常还要转换为10进制, 即LBP码, 共256种) , 将这个值作为窗口中心像素点的LBP值, 以此来反应这个 3×3 区域的纹理信息。



三、目标检测方法



$$LBP(xc, yc) = \sum_{p=0}^7 s(I(p) - I(c)) * 2^p$$

1	2	3
0	(i,j)	4
7	6	5

其中， p 表示 3×3 窗口中除中心像素点外的第 p 个像素点； $I(c)$ 表示中心像素点的灰度值， $I(p)$ 表示领域内第 p 个像素点的灰度值； $s(x)$ 公式如下：

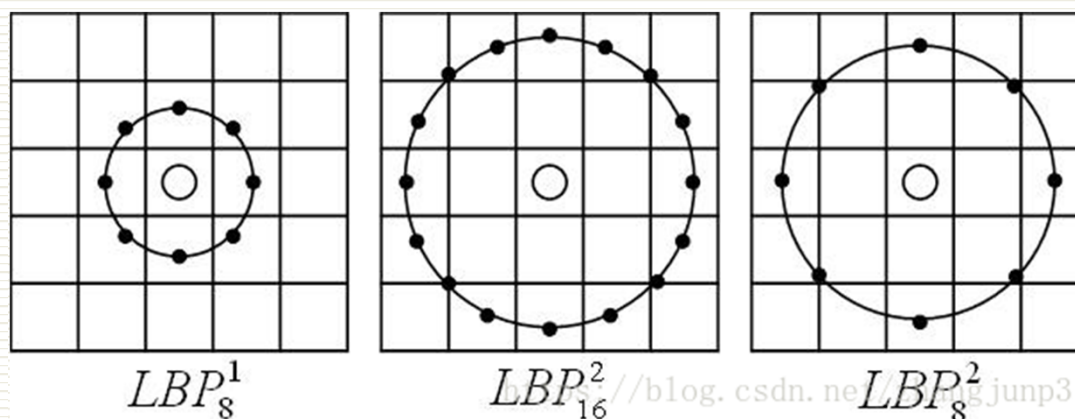
$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & otherwise \end{cases}$$

- ① LBP记录的是中心像素点与领域像素点之间的差值；
- ② 当光照变化引起像素灰度值同增同减时，LBP变化并不明显；
- ③ LBP对与光照变化不敏感，LBP检测的仅仅是图像的纹理信息；



三、目标检测方法

基本的LBP算子直接利用的灰度比较，所以其具有灰度不变性；存在缺点：1、产生的二进制模式多；2、不具有旋转不变性。**圆形LBP算子**如下：



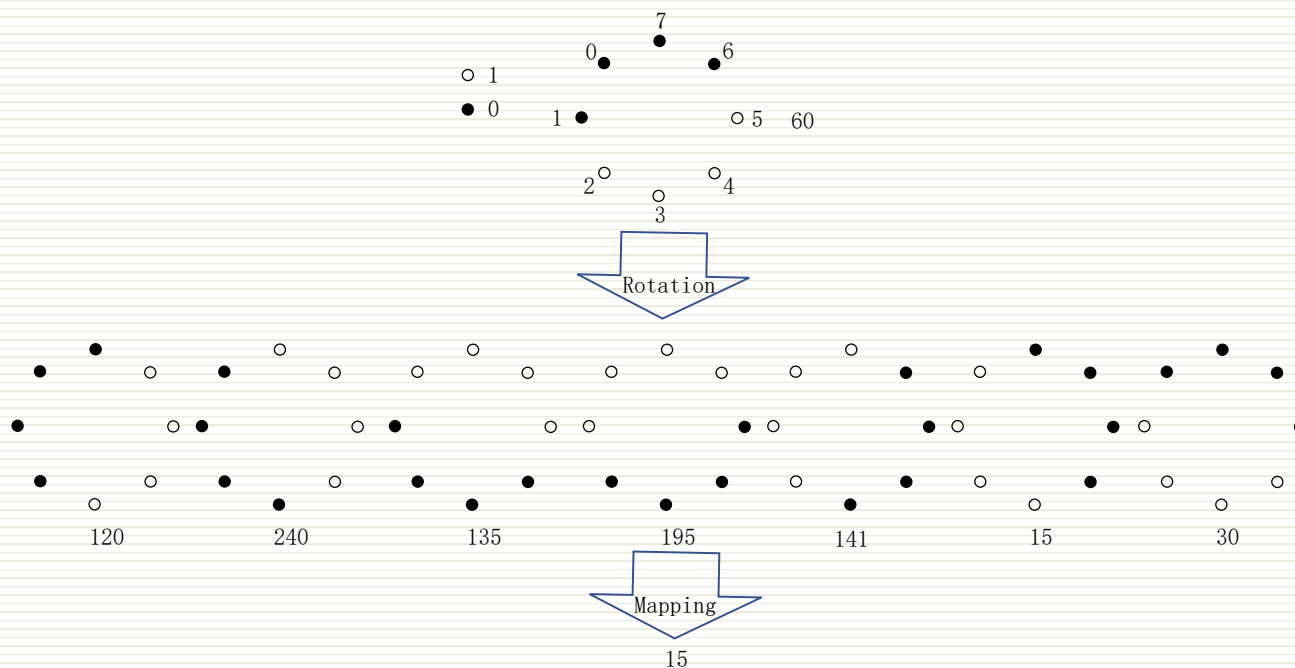
1,2指的是半径，
8,16指的是采样点数



三、目标检测方法

旋转不变的LBP算子

- ◆ 由于LBP的二进制模式是以一定的方向、顺序进行编码的，所以当图像发生旋转时，按这种编码的话，LBP值会发生改变，因此是不具有旋转不变性的。Maenpaa等人提出了具有旋转不变性的LBP算子。
- ◆ **解决办法：**不断旋转邻域得到一系列的LBP值，取其中最小值作为该邻域的LBP值。旋转过程实质上就是对二进制模式进行循环移位的过程。





三、目标检测方法

(1) 直方图匹配

提取灰度直方图或H分量直方图，采用直方图匹配方法进行特征匹配。

目标的特征采用直方图描述。假设得到的目标和模板的直方图分别为 $H_1(i)$ 和 $H_2(i)$ ，其中 i 为直方图的柱，描述两个直方图相似程度。

相关法：数值越大则匹配程度越高。完全匹配时数值为1，完全不匹配时是-1，值为0则表示无关联（随机组合）。

$$d_{\text{correl}}(H_1, H_2) = \frac{\sum_i H'_1(i) \cdot H'_2(i)}{\sqrt{\sum_i H'_1(i) \cdot H'_2(i)}} \quad H'_k(i) = H_k(i) - (1/N) \left(\sum_j H_k(j) \right)$$



三、基于特征匹配的目标检测

卡方法 (chi-square) :

$$d_{\text{chi-square}}(H_1, H_2) = \sum_i \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)}$$

对于chi-square, 数值越小的匹配程度越高。完全匹配时值为0, 完全不匹配时为无限值 (依赖于直方图的大小)。

直方图相交法:

$$d_{\text{intersection}}(H_1, H_2) = \sum_i \min(H_1(i), H_2(i))$$

对于直方图相交法, 数值越大表示匹配越好。如果两个直方图都被归一化到1, 则完全匹配是1, 完全不匹配是0。

Bhattacharyya距离法 (巴氏距离) :

$$d_{\text{Bhattacharyya}}(H_1, H_2) = \sqrt{1 - \frac{\sum_i \sqrt{H_1(i) \cdot H_2(i)}}{\sum_i H_1(i) \cdot \sum_i H_2(i)}}$$

对于Bhattacharyya匹配, 数值越小则匹配程度越高, 完全匹配时值为0, 完全不匹配时值1。



三、基于特征匹配的目标检测

double **compareHist**(InputArray H1, InputArray H2, int method);

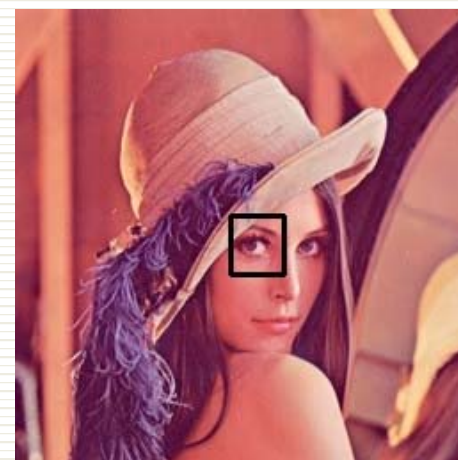
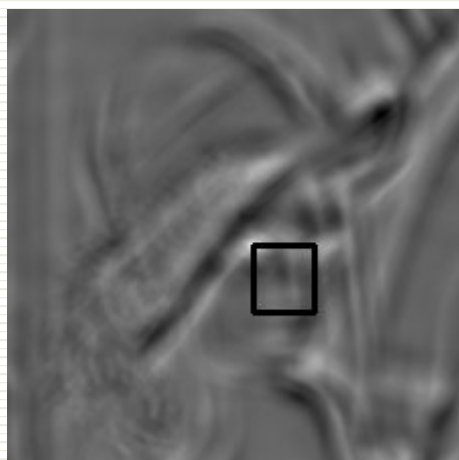
方法	参数
相关 Correlation	HISTCMP_CORREL
卡方 Chi-square	HISTCMP_CHISQR
相交 Intersection	HISTCMP_INTERSECT
巴氏距离 Bhattacharyya	HISTCMP_BHATTACHARYYA

三、目标检测方法

(2) 模板匹配

用于在源图像中寻找定位给定目标图像（即模板图像）。其原理是通过一些相似度准则来衡量两个图像块之间的相似度。

在图像中匹配模板时，需要滑动匹配窗口（即模板图像的大小），计算模板图像与该窗口对应的图像区域之间的相似度。对整张图像滑动完后，得到多个匹配结果。





三、基于特征匹配的目标检测

● 匹配方法

以8 位图像为例，模板 T ($m \times n$) 叠放在被搜索图 S ($W \times H$) 上平移，模板覆盖被搜索图的那块区域叫子图 S_{ij} 。 i 、 j 分别为子图左上角在被搜索图 S 上的坐标。搜索范围是： $1 \leq i \leq W - m, 1 \leq j \leq H - n$

通过比较 T 和 S_{ij} 的相似性，完成模板匹配过程。衡量模板 T 和 S_{ij} 的匹配程度, 可采用 TM_CCORR 、 TM_SQDIFF 、 TM_CCOEFF 方法。

互相关算法 (cross-correlation) 通过计算模板图像和匹配图像的**互相关程度**，来确定匹配的程度。**互相关程度**最大时的搜索窗口位置决定了模板图像在待匹配图像中的位置。

误差平方和算法 (Sum of Squared Differences, 简称SSD算法)，也叫差方和算法。

$$SSD(i, j) = \text{Min} \left(\sum_{m=1}^M \sum_{n=1}^N [S_{ij}(m, n) - T(m, n)]^2 \right)$$



三、基于特征匹配的目标检测

● 其他匹配方法

1. 平均绝对差算法 (Mean Absolute Differences, 简称MAD算法)
2. 绝对误差和算法 (Sum of Absolute Differences, 简称SAD算法)
3. 误差平方和算法 (Sum of Squared Differences, 简称SSD算法), 也叫差方和算法
4. 平均误差平方和算法 (Mean Square Differences, 简称MSD算法), 也称均方差算法 (MSD之于SSD, 等同于MAD之于SAD)
5. 归一化积相关算法 (Normalized Cross Correlation, 简称NCC算法), 利用子图与模板图的灰度, 通过归一化的相关性度量公式来计算二者之间的匹配程度。
6. 序贯相似性检测算法 (Sequential Similarity Detection Algorithm, 简称SSDA算法), 是对传统模板匹配算法的改进, 比MAD算法快几十到几百倍



三、基于特征匹配的目标检测

● 程序实现

```
void matchTemplate( const CvArr* image, const CvArr* templ, CvArr*  
result, int method );
```

image欲搜索的图像。它应该是单通道、8-比特或32-比特浮点数图像

templ搜索模板，不能大于输入图像，且与输入图像具有一样的数据类型

result比较结果的映射图像。单通道、32-比特浮点数。如果图像是 $W \times H$ 而 templ 是 $w \times h$ ，则 result 一定是 $(W-w+1) \times (H-h+1)$ 。

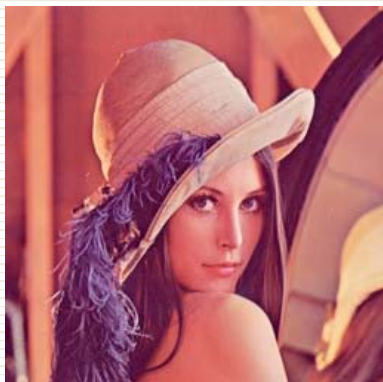
method指定匹配方法：

- ◆ CV_TM_SQDIFF 平方差匹配法，最好的匹配为0，值越大匹配越差
- ◆ CV_TM_SQDIFF_NORMED 归一化平方差匹配法
- ◆ CV_TM_CCORR 相关匹配法，采用乘法操作，数值越大表明匹配越好
- ◆ CV_TM_CCORR_NORMED 归一化相关匹配法
- ◆ CV_TM_CCOEFF 相关系数匹配法，最好的匹配为1，-1表示最差的匹配
- ◆ CV_TM_CCOEFF_NORMED 归一化相关系数匹配法

前面两种方法为越小的值表示越匹配，后四种方法值越大越匹配。

三、基于特征匹配的目标检测

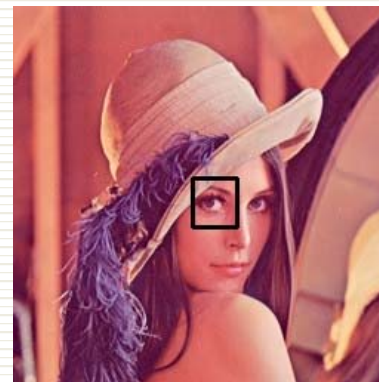
● 实验结果



(a) 原始图像



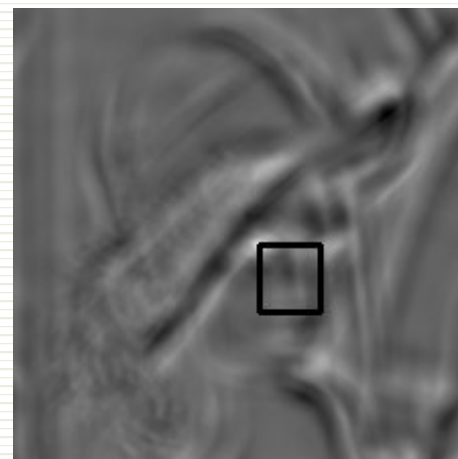
(b) 模板图像



(c) 匹配结果图像



模板匹配过程图



模板匹配结果图



三、目标检测方法

(3) 汉明距离匹配

通过汉明距离计算比较两个二进制码串的距离，汉明距离表示两个等长字符串在对应位置上不同字符的数目。

```
int HanmingDistance(string &str1, string &str2,int num) {  
    //判断当两个字符串的长度是否相等  
    if ((str1.size() != num) || (str2.size() != num))  
        return -1;  
    int difference = 0; //遍历字符串比较两个字符串的0与1的不相同的地方，不相同一次就  
    长度增加1从而计算总距离  
    for (int i = 0; i < num; i++)  
    {  
        if (str1[i] != str2[i])  
            difference++;  
    }  
    return difference;  
}
```

实验课 —— 编写程序



- 1、将RGB图像转化为HSV图像，编写直方图匹配算法实现目标检测；
- 2、编写模板匹配算法；
- 3、编写LBP提取纹理特征，并进行特征匹配实现目标检测。