

模式识别

第八章 图像特征提取

郭园方

北京航空航天大学计算机学院

小调研

- * 各位同学的研究大方向分别是？
- * 例如：图像处理、计算机视觉、视频处理、文本处理、音频处理、数据挖掘、编译器等

引言

* 计算机不认识图像，只认识数字



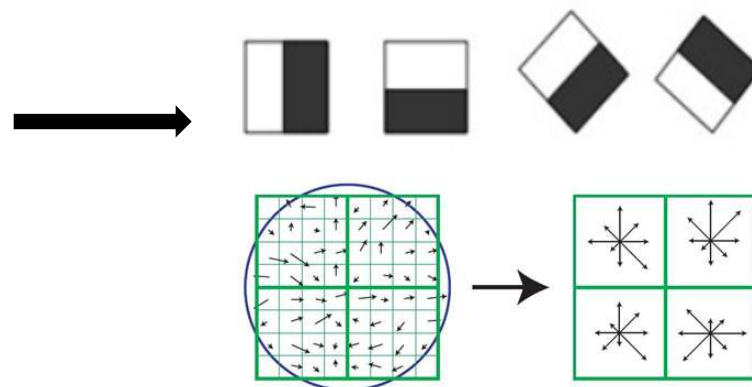
提取有用的数据或信息，得到图像的“非图像”的表示或描述



R
G
B

12	11	10	9	23	...	8	28	34	93	12
9	20	13	35	2	...	9	23	53	29	10
11	13	17	9	23	...	11	80	23	23	12
...
11	11	10	9	23	...	8	27	23	13	12
9	20	13	35	2	...	97	63	53	27	15
12	11	10	9	23	...	95	23	23	33	11

数值、向量和符号等



引言

* 图像特征的定义

图像特征可以定义为某一张或一类图像区别于其他图像的相应本质特点或特性的表示及描述，以及这些描述的集合。

* 特征提取的一般原则

为了识别出某图像所属的类别，选取的特征不仅要能够很好地描述图像，更重要的是还要能够很好地区分不同类别的图像。

我们希望选择那些在同类图像之间差异较小（较小的类内距），在不同类别的图像之间差异较大（较大的类间距）的图像特征。

引言



耳朵 胡须 爪子
(1, 1, 1)



耳朵 胡须 爪子
(1, 1, 1)



~~耳朵 胡须 爪子~~
(1, 0, 1)

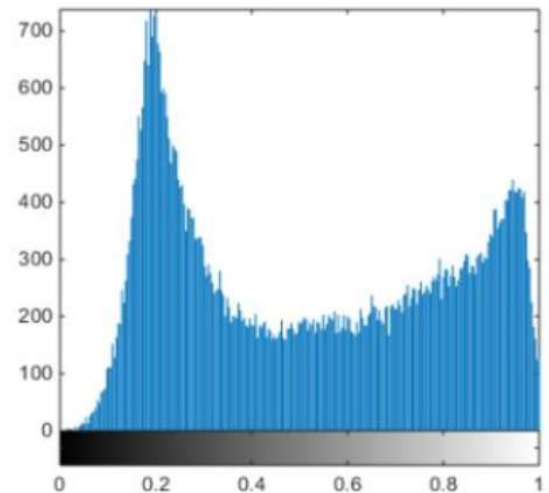
常见的图像底层特征提取算法

- * **颜色**特征：如颜色直方图、颜色集、颜色矩、颜色聚合向量等；
- * **纹理**特征：如Tamura纹理特征、自回归纹理模型、Gabor变换、小波变换、MPEG7边缘直方图等；
- * **形状**特征：如傅立叶形状描述符、不变矩、小波轮廓描述符等；

常见的图像底层特征提取算法



(a) 色织物图像



(b) 对应直方图

颜色直方图

- * 《数字图像处理》 拉斐尔C.冈萨雷斯, 理查德 E.伍兹
- * Gonzalez R C, Woods R E. Digital image processing[J]. 2012.

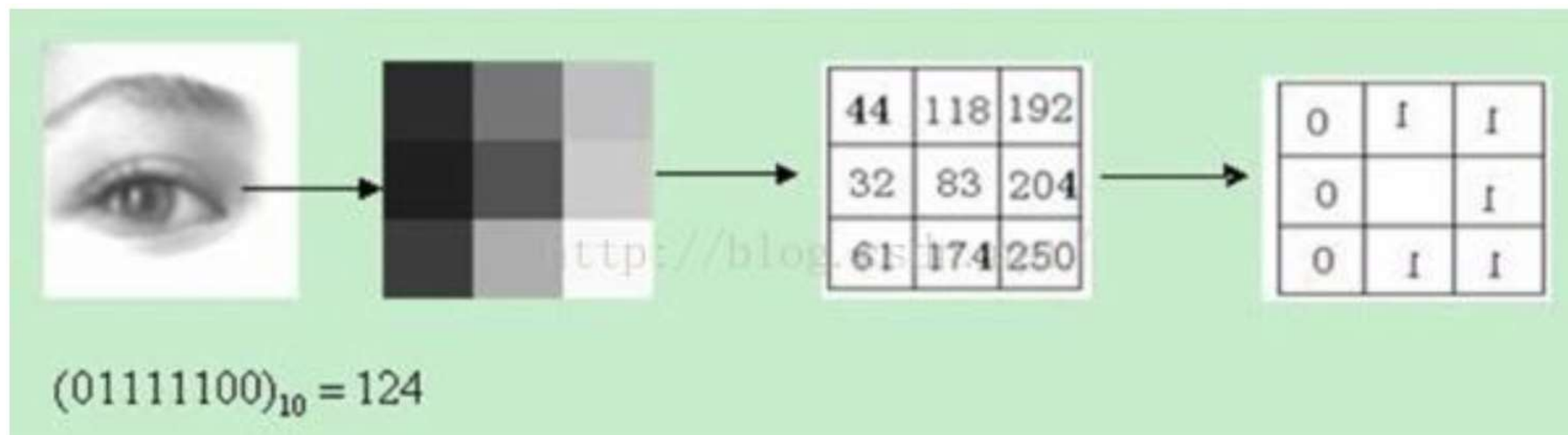
常见图像应用特征提取算法

- * **LBP** (Local Binary Patterns, 局部二值模式)
纹理分类、人脸分析
- * **LDA** (Linear Discriminant Analysis)
- * **HOG** (Histogram of Oriented Gradient, 方向梯度直方图)
行人检测、**DPM** (目标检测算法)
- * **Haar-like** 特征
人脸表示、人脸识别

局部二值模式 (Local Binary Patterns, LBP)

- * 原始的**LBP**算子定义为在**3*3**的窗口内，以窗口**中心像素**为阈值，将相邻的**8**个像素的灰度值与其进行比较，若周围像素值大于中心像素值，则该像素点的位置被标记为**1**，否则为**0**。
- * 这样，**3*3**邻域内的**8**个点经比较可产生**8位二进制数**（通常**转换为十进制数**即**LBP码**，共**256种**），即得到该窗口中心像素点的**LBP值**，并用这个值来反映该区域的**纹理**信息。

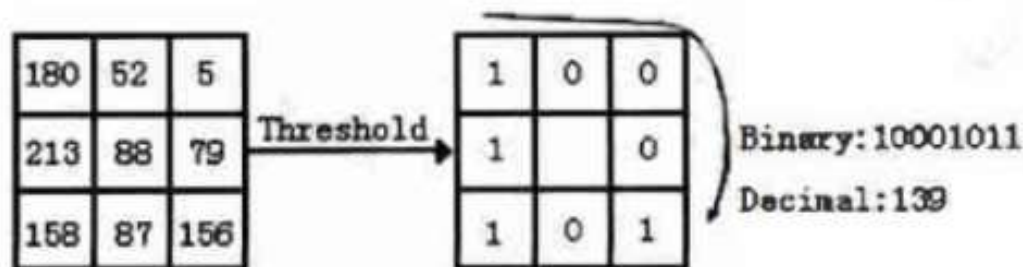
局部二值模式 (LBP)



LBP码

LBP的主要思想

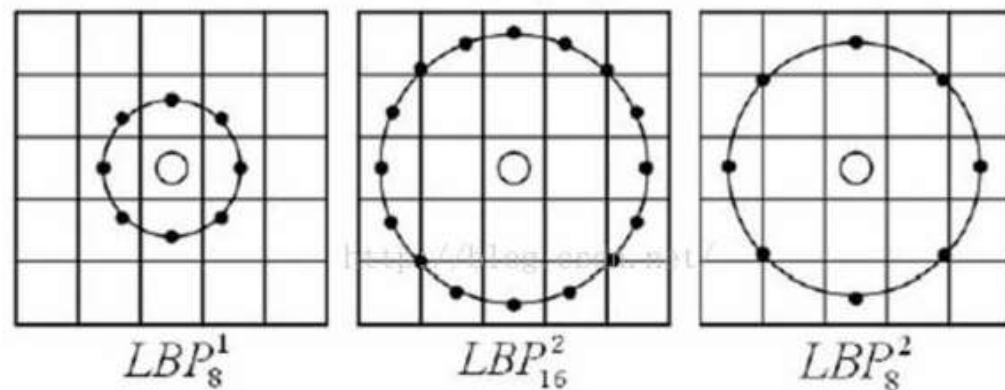
- * 以某一点与其邻域像素的相对灰度作为响应，正是这种相对机制使LBP算子对于单调的灰度变化具有不变性。
- * 人脸图像常常会受到光照因素的影响而产生灰度变化，但在一个局部区域内，这种变化常常可以被视为是单调的，因此LBP在光照不均的人脸识别应用中也取得了很好的效果。



LBP的改进版本-圆形LBP算子

- * 基本的 LBP 算子的最大缺陷在于它只覆盖了一个固定半径范围内的小区域，这显然不能满足不同尺寸和频率纹理的需要。
- * 为了适应不同尺度的纹理特征，并达到灰度和旋转不变性的要求，研究人员用圆形邻域代替了正方形邻域，改进后的 LBP 算子允许在半径为 R 的圆形邻域内有 P 个采样点。
- * 不在像素中心位置的点通过双线性插值确定其值。

LBP的改进版本-圆形LBP算子



原图



初始LBP算子



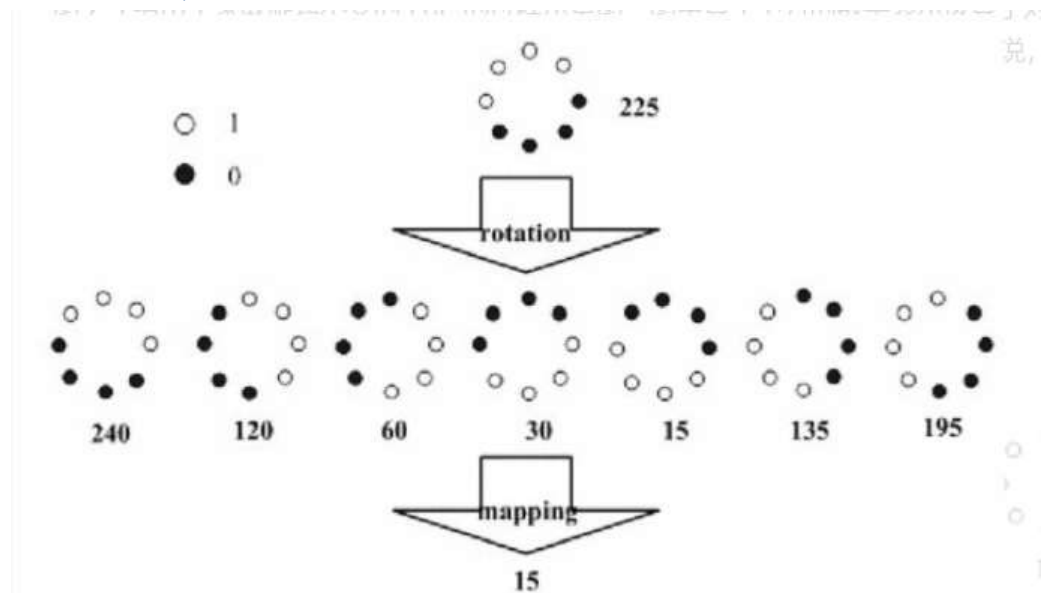
圆形LBP算子

LBP的改进版本 - 旋转不变模式

- * 从LBP的定义可以看出，LBP算子是灰度不变的，但却不是旋转不变的。图像的旋转就会得到不同的LBP值。
- * 研究人员又将LBP算子进行了扩展，提出了具有旋转不变性的LBP算子，即不断旋转圆形邻域得到一系列初始定义的LBP值，取其最小值作为该邻域的LBP值。

LBP的改进版本 - 旋转不变模式

- * 图中算子下方的数字表示该算子对应的LBP值，图中所示的8种LBP模式，经过旋转不变的处理，最终得到的具有旋转不变性的LBP值为15。也就是说，图中的8种LBP模式对应的旋转不变的LBP模式都是00001111

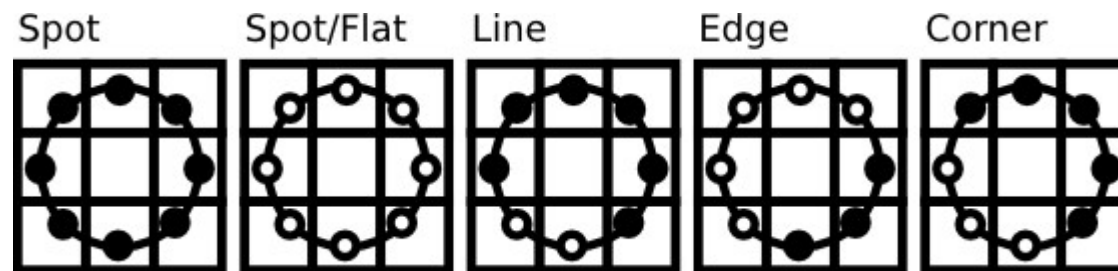


LBP等价模式

- * 目的：LBP等价模式目的即为对LBP算子进行降维。LBP算子可以产生非常多二进制模式，例如：5x5邻域有 $p=24$ 个采样点，则可得到 $2^{24}=16777216$ 种二进制模式。过多的二进制模式对模式识别是一种负担，因此需要对原始的LBP模式进行降维。
- * 定义：某个局部二进制模式所对应的循环二进制数从0—>1或从1—>0，最多有两次跳变，该局部二进制模式所对应的二进制就成为一个等价模式

LBP等价模式

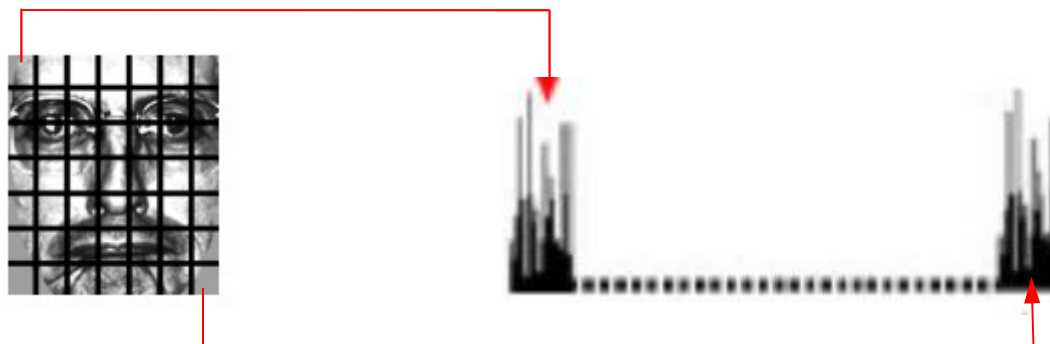
- * 做法：当小于两次跳变时，该类型保留；当超过2次时，该类型删除。
- * 效果：二进制模式的种类减少，而不会丢失任何信息，模式种类由原来的 2^p 减少为 $p*(p-1)+2$ 种。



几种等价模式的示意图

LBP特征应用

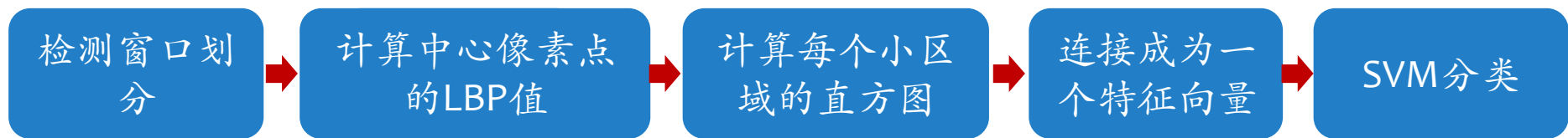
- * **LBP**的应用中，如纹理分类、人脸分析等，一般都不将**LBP**图谱作为特征向量用于分类识别，而是采用**LBP**特征谱的统计直方图作为特征向量用于分类识别



LBP特征转换直方图流程

LBP特征应用

* **LBP**的应用中，如纹理分类、人脸分析等，一般都不将**LBP**图谱作为特征向量用于分类识别，而是采用**LBP**特征谱的统计直方图作为特征向量用于分类识别

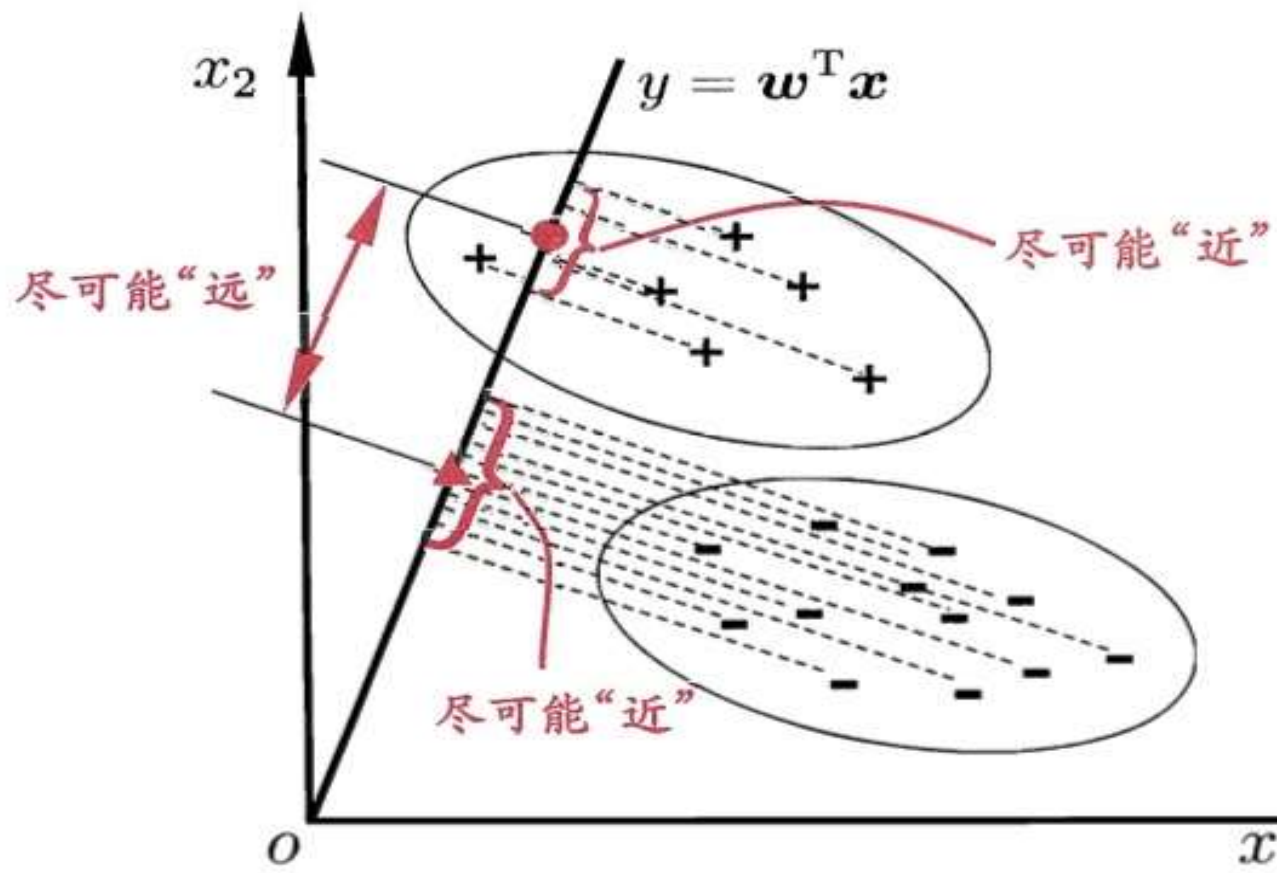


基于LBP特征的图像分类流程

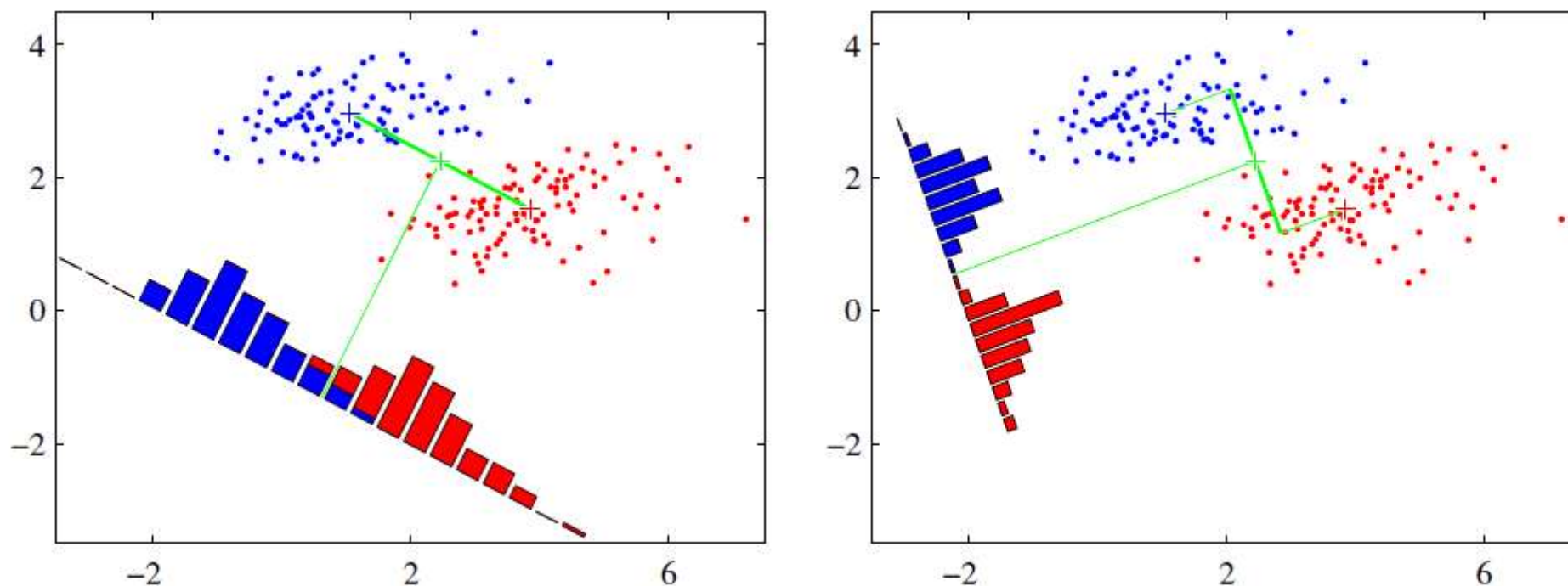
线性判别分析(Linear discriminant Analysis, LDA)

- * 线性判别分析(Linear discriminant Analysis, LDA)是一种监督学习的降维技术，目的是最优化分类的特征子空间。
- * LDA的思想可以用一句话概括，就是“**投影后类内方差最小，类间方差最大**”。如图：即要将数据在低维度上进行投影，投影后希望每一种类别数据的投影点尽可能的接近，而不同类别的数据的类别中心之间的距离尽可能的大。

线性判别分析(LDA)



线性判别分析(LDA)



哪一种能更好的满足我们的标准呢？从直观上可以看出，右图要比左图的投影效果好，因为右图的数据和蓝色数据各个较为集中，且类别之间的距离明显。左图则在边界处数据混杂。

LDA-二分类情况

* 假设我们的数据集:

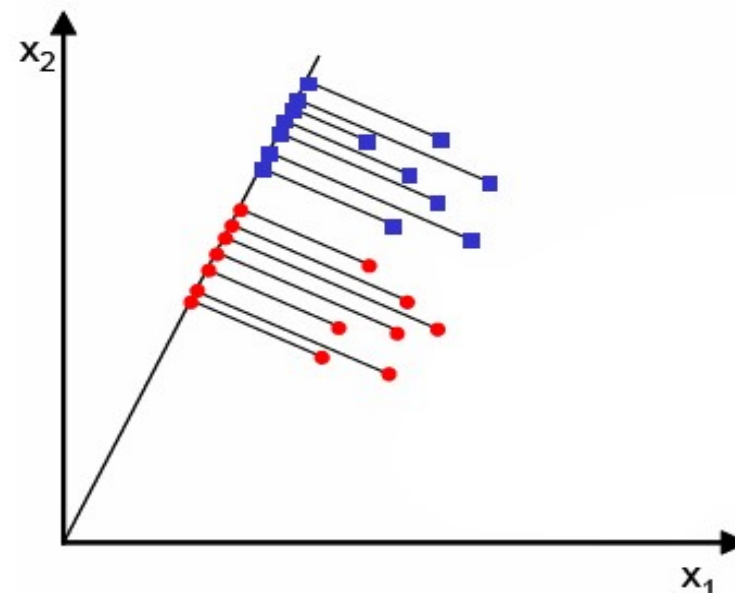
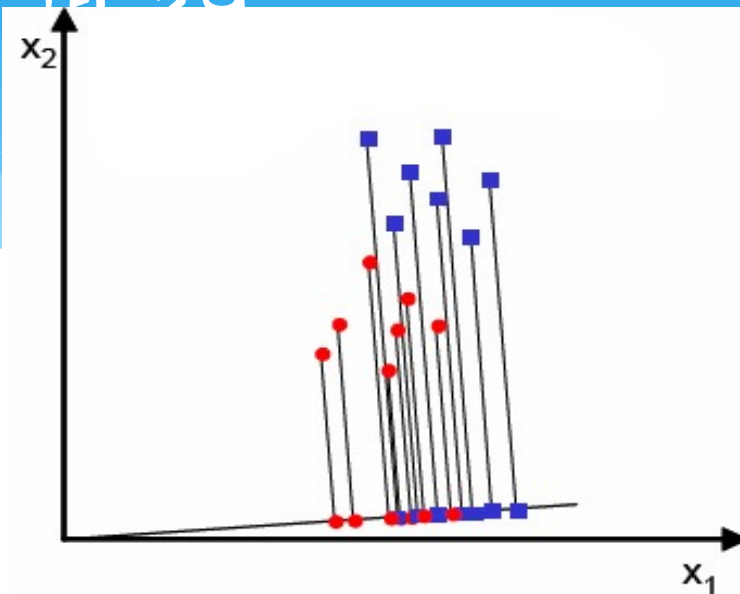
$D=\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$, N_j ($j=0,1$) 为第 j 类样本的个数。

* 我们需要寻找一个投影 y , 即将 x 投影到该直线上

$$y = w^T x \quad \text{where} \quad x = \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ x_m \end{bmatrix} \quad \text{and} \quad w = \begin{bmatrix} w_1 \\ \cdot \\ \cdot \\ w_m \end{bmatrix}$$

LDA-二分类情况

- * 满足的直线中，我们期望选择一个使得类内距离最小，类间距离最大。



LDA-二分类情况

* 我们定义以下变量:

* 均值向量

$$\mu_i = \frac{1}{N_i} \sum_{x \in \omega_i} x \quad \text{and} \quad \tilde{\mu}_i = \frac{1}{N_i} \sum_{y \in \omega_i} y = \frac{1}{N_i} \sum_{x \in \omega_i} w^T x$$
$$= w^T \frac{1}{N_i} \sum_{x \in \omega_i} x = w^T \mu_i$$

$$J(w) = |\tilde{\mu}_1 - \tilde{\mu}_2| = |w^T \mu_1 - w^T \mu_2| = |w^T (\mu_1 - \mu_2)|$$

* 将均值向量距离作为优化函数

LDA-二分类情况

* 上述优化未考虑类内散度，因此再定义散度：

$$\tilde{s}_i^2 = \sum_{y \in \omega_i} (y - \tilde{\mu}_i)^2$$

\tilde{s}_i^2 表示了x投影到直线 w_i 上的散度

$\tilde{s}_1^2 + \tilde{s}_2^2$ 表示了两类样本投影之后的散度

* 重新定义的优化函数为

$$J(w) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

LDA-二分类情况

* 我们一般定义类内散度矩阵 S_w 为：

$$S_i = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T$$

$$S_w = S_1 + S_2$$

LDA-二分类情况

- * 因此投影 y 可以表示为在 x 的特征空间下散度矩阵的函数:

$$\begin{aligned}\tilde{s}_i^2 &= \sum_{y \in \omega_i} (y - \tilde{\mu}_i)^2 = \sum_{x \in \omega_i} (w^T x - w^T \mu_i)^2 \\ &= \sum_{x \in \omega_i} w^T (x - \mu_i)(x - \mu_i)^T w \\ &= w^T S_i w\end{aligned}$$

$$\tilde{s}_1^2 + \tilde{s}_2^2 = w^T S_1 w + w^T S_2 w = w^T (S_1 + S_2) w = w^T S_W w = \tilde{S}_W$$

类内
散度矩阵

LDA-二分类情况

* 相应地，类间散度可表示为：

$$\begin{aligned}(\tilde{\mu}_1 - \tilde{\mu}_2)^2 &= (w^T \mu_1 - w^T \mu_2)^2 \\&= w^T \underbrace{(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T}_{S_B} w \\&= w^T S_B w = \tilde{S}_B\end{aligned}$$

LDA-二分类情况

* 最终，我们可以用**fisher**准则将优化函数写为：

$$J(w) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{w^T S_B w}{w^T S_W w}$$

类间散度矩阵

类内散度矩阵

* 即最大化类间差，最小化类内差

LDA-二分类情况

* 优化 $J(w)$ 问题可以等价求极值问题:

$$\frac{d}{dw} J(w) = \frac{d}{dw} \left(\frac{w^T S_B w}{w^T S_W w} \right) = 0$$

$$\Rightarrow S_B w - J(w) S_W w = 0$$

$$\Rightarrow S_W^{-1} S_B w - J(w) w = 0$$

$$w^* = \arg \max_w J(w) = \arg \max_w \left(\frac{w^T S_B w}{w^T S_W w} \right) = S_W^{-1} (\mu_1 - \mu_2)$$

LDA vs PCA

- * 两者均可以对数据进行降维。
- * 两者在降维时均使用了矩阵特征分解的思想。
- * 两者都假设数据符合高斯分布。
- * **LDA是有监督的降维方法，而PCA是无监督的降维方法**
- * **LDA降维最多降到类别数 $k-1$ 的维数，而PCA没有这个限制**
- * **LDA除了可以用于降维，还可以用于分类**

方向梯度直方图 (Histogram of Oriented Gradient, HOG)

- * **HOG**特征通过计算和统计图像局部区域的梯度方向直方图来构成特征。**HOG**特征结合**SVM**分类器已经被广泛应用于图像识别中，尤其在**行人检测**中获得了极大的成功。
- * 主要思想：在一幅图像中，局部目标的**表象和形状**能够被梯度或边缘的方向密度分布很好地描述。
(本质：梯度的统计信息，而梯度主要存在于边缘的地方)。
- * 实现方法：首先将图像分成小的连通区域，我们把它叫**细胞单元 (cell)**。然后采集细胞单元中各像素点的梯度的或边缘的方向直方图。最后把这些直方图组合起来就可以构成特征描述器。

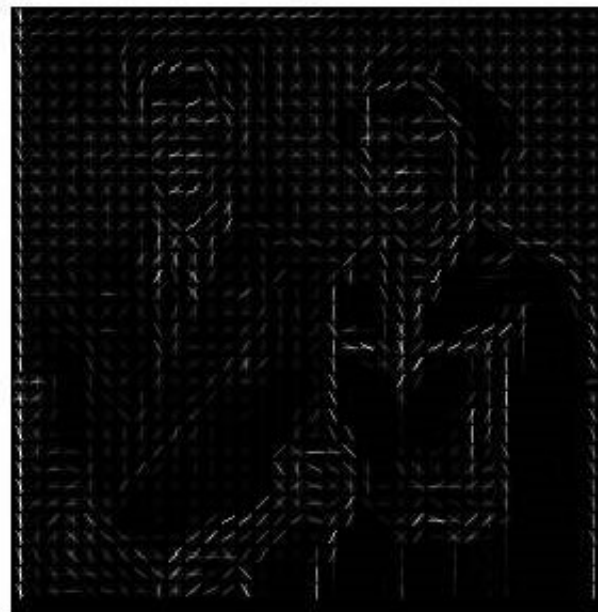
方向梯度直方图 (HOG)

* HOG示例

Input image



Histogram of Oriented Gradients



HOG特征提取实现过程

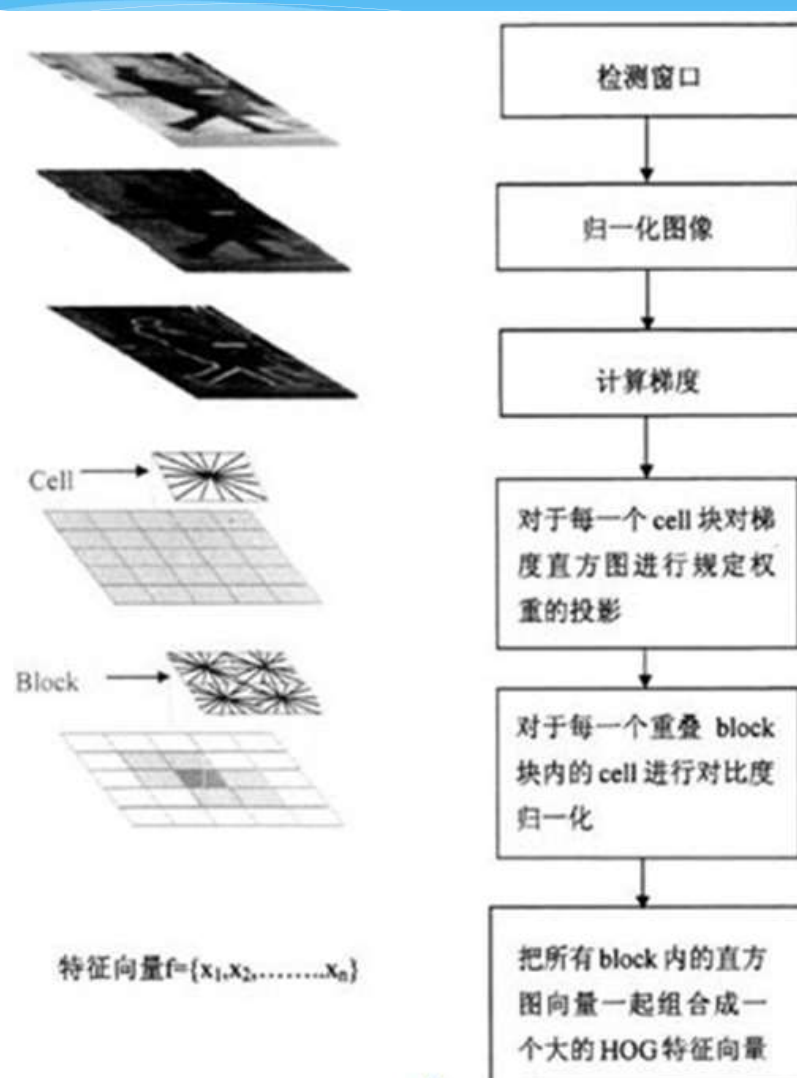
* **HOG**特征提取方法就是将一个图像（你要检测的目标或者扫描窗口）：

- 1) 灰度化（将图像看做一个 x, y, z （灰度）的三维图像），归一化；
- 2) 计算图像每个像素的梯度（包括大小和方向）；
- 3) 将图像划分成小**cells**（例如 6×6 像素/**cell**）；
- 4) 统计每个**cell**的梯度直方图（不同梯度的个数），即可形成每个**cell**的描述特征；

HOG特征提取实现过程

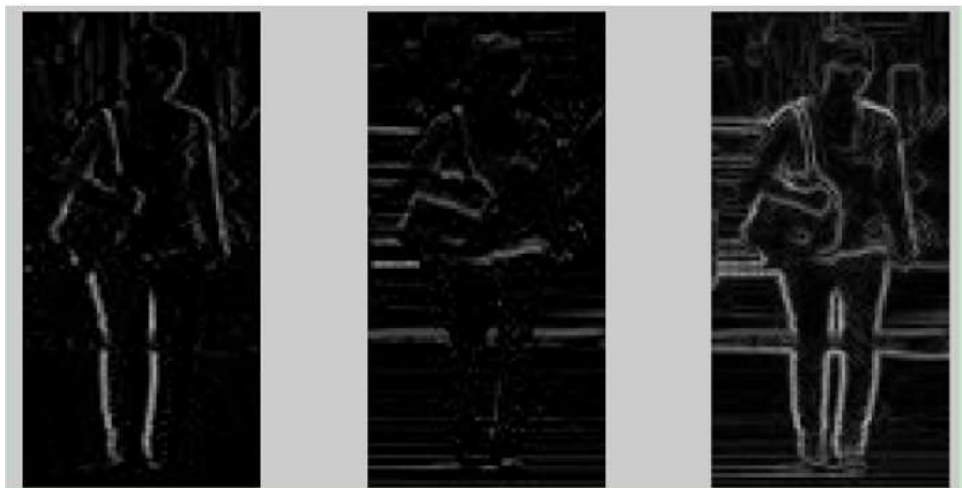
- 5) 将每几个**cell**组成一个**block**（例如3x3个**cell/block**），一个**block**内所有**cell**的特征串联起来便得到该**block**的HOG特征。
- 6) 将图像内的所有**block**的HOG特征串联起来就可以得到该**image**（你要检测的目标）的HOG特征了。这个就是最终的可供分类使用的特征向量了。

HOG特征提取实现过程



HOG - 计算图像梯度

- * 计算图像横坐标和纵坐标方向的梯度，并据此计算每个像素位置的梯度方向值；求导操作不仅能够捕获轮廓，人影和一些纹理信息，还能进一步弱化光照的影响。



从左到右分别是水平方向梯度图，垂直方向梯度图和梯度幅值图

HOG - 计算图像梯度

* 图像中像素点 (x, y) 的梯度为：

$$G_x(x, y) = H(x + 1, y) - H(x - 1, y)$$

$$G_y(x, y) = H(x, y + 1) - H(x, y - 1)$$

* 注：梯度求取方式多，可使用其他求取方式

HOG - 计算图像梯度

- * 其中 $G_x(x, y)$ 和 $G_y(x, y)$ 分别表示输入图像中像素点 (x, y) 的水平方向梯度和垂直方向梯度。
 $H(x, y)$ 表示像素值。再以此计算出梯度的大小和方向：

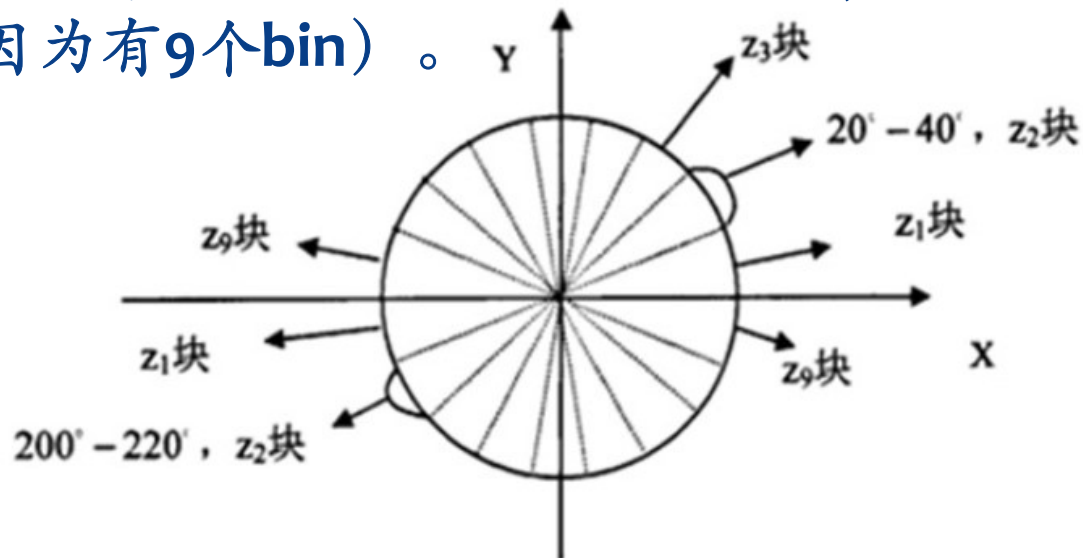
$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}$$
$$\alpha(x, y) = \tan^{-1} \frac{G_x(x, y)}{G_y(x, y)}$$

HOG - 构建梯度直方图

- * 构建梯度直方图的目的是为局部图像区域提供一个**编码**，同时能够保持对图像中对象的姿势和外观的弱敏感性。
- * 我们将图像分成若干个“单元格**cell**”（例如6x6像素/**cell**）。

HOG - 构建梯度直方图

- * 假设我们采用9个bin的直方图来统计这6x6个像素的梯度信息。如图所示：如果这个像素的梯度方向是20-40度，直方图第2个bin的计数就加对应梯度大小的权值，这样，对cell内每个像素映射到固定的角度范围，就可以得到这个cell的梯度方向直方图了，就是该cell对应的9维特征向量（因为有9个bin）。



HOG - 归一化梯度直方图

- * 由于局部光照的变化以及前景-背景对比度的变化，使得梯度强度的变化范围非常大。
- * 这就需要对梯度强度做归一化。归一化能够进一步地对光照、阴影和边缘的变化进行压缩。

(a) $L2-norm, v \leftarrow v / \sqrt{\|v\|_2^2 + \varepsilon^2}$ (ε 是一个极小的常数，避免分母为 0)

(b) $L2-Hys$ (方法同上，只是限制 v 的最大值到 0.2，然后再次重新标准化)

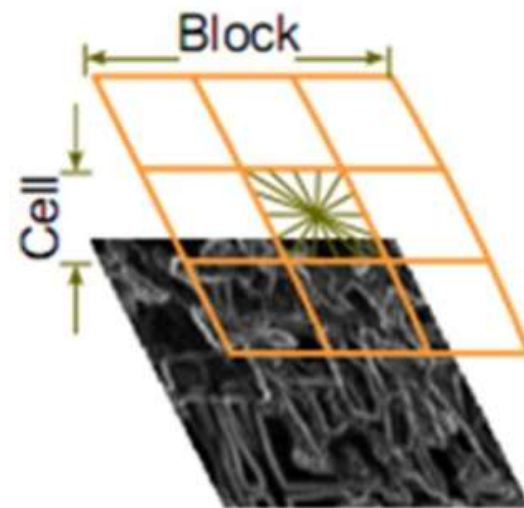
(c) $L1-norm, v \leftarrow v / \sqrt{\|v\|_1 + \varepsilon}$

(d) $L1-sqrt, v \leftarrow v / \sqrt{\|v\|_1 + \varepsilon}$

常用归一化函数

HOG - 归一化梯度直方图

- * 把各个细胞单元组合成大的、空间上连通的区间 (**blocks**)。这样，一个**block**内所有**cell**的特征向量串联起来便得到该**block**的**HOG**特征。这些区间是互有重叠的，这就意味着：每一个**cell**的特征会以不同的结果多次出现在最后的特征向量中。我们将归一化之后的块描述符（向量）就称之为**HOG**描述符。



HOG的可视化

* 所检测的局部物体**外形**能够被**HOG**较好地捕捉和描述。



HOG的优点

- * 核心思想是所检测的局部物体外形能够被梯度或边缘方向的分布所描述，**HOG**能较好地捕捉局部形状信息，对几何和光学变化都有很好的不变性；
- * **HOG**是在密集采样的图像块中求取的，在计算得到的**HOG**特征向量中隐含了该块与检测窗口之间的空间位置关系。

HOG的缺陷

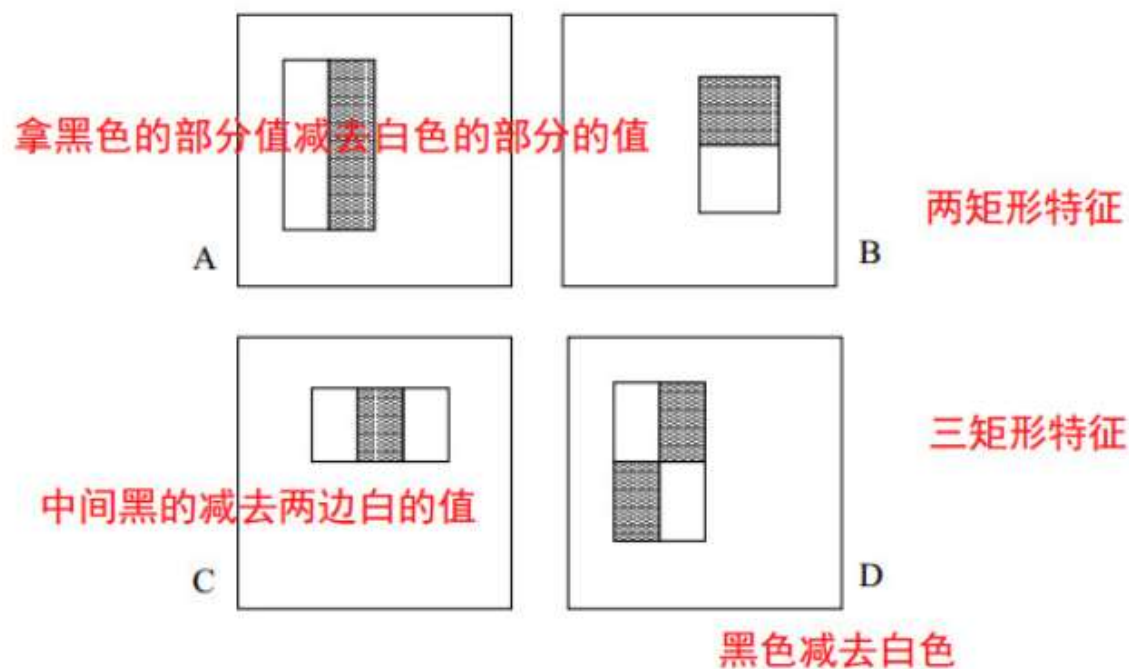
- * 很难处理**遮挡**问题，人体姿势动作幅度过大或物体方向改变也不易检测（这个问题后来在**DPM**中采用可变形部件模型的方法得到了改善）；
- * **HOG**没有选取主方向，也没有旋转梯度方向直方图，因而本身不具有**旋转不变性**（较大的方向变化），其旋转不变性是通过采用不同旋转方向的训练样本来实现的；
- * **HOG**本身不具有**尺度不变性**，其尺度不变性是通过缩放检测窗口图像的大小来实现的；
- * 此外，由于梯度的性质，**HOG**对**噪点**相当敏感，在实际应用中，在**Block**和**Cell**划分之后，对于得到各个像区域中，有时候还会做一次高斯平滑去除噪点。

Haar-like 特征

- * 一个Harr特征可以定义为矩形中几个区域的像素和的差值，这个值反映了图像的灰度变化情况。
- * 例如：脸部的一些特征能由矩形特征简单的描述，如：眼睛要比脸颊颜色要深，鼻梁两侧比鼻梁颜色要深，嘴巴比周围颜色要深等。
- * 但矩形特征只对一些简单的图形结构，如边缘、线段较敏感，只能描述特定走向（水平、垂直、对角）的结构，对于人类检测应用有很好的表现。

Haar-like 特征

拿黑色部分的所有的
像素值的和减去白色
部分所有像素值的和。
得到的就是一个特征
值



Haar-like 特征的计算-积分图

- * 在工程中需要进行快速计算某个矩形内的像素值的和，这就需要引入积分图的概念。
- * 积分图是一种能够描述全局信息的矩阵表示方法。积分图的构造方式是位置 (x, y) 处的值 $ii(x, y)$ 是原图像 (x, y) 左上角方向所有像素的和：

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

Haar-like 特征的计算-积分图

* 积分图构建算法:

1. 用 $s(x, y)$ 表示行方向的累加和, 初始化 $s(x, -1) = 0$;
2. 用 $ii(x, y)$ 表示积分图, 初始化 $ii(-1, y) = 0$;
3. 逐行扫描图像, 递归计算每个像素的 $s(x, y)$ 和 $ii(x, y)$

$$s(x, y) = s(x, y - 1) + i(x, y)$$

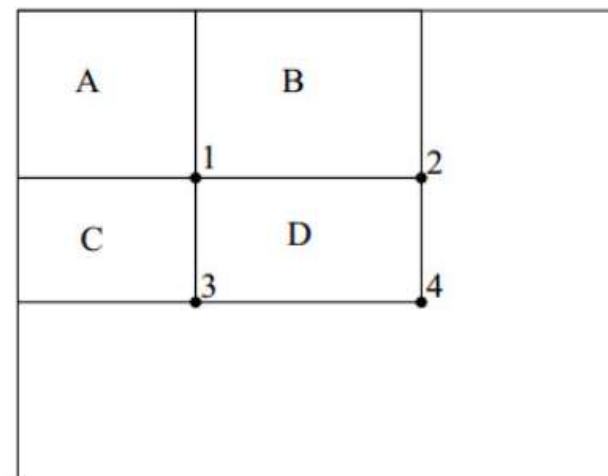
$$ii(x, y) = ii(x - 1, y) + s(x, y)$$

4. 扫描图像一遍, 当到达图像右下角像素时, 积分图像就构造好了

Haar特征构造与个数计算

* 设D的四个顶点分别为1、2、3、4，则D的像素和可以表示为

$$\begin{aligned} D_{sum} &= ii(1) + ii(4) - (ii(2) \\ &\quad + ii(3)) \end{aligned}$$



Haar特征构造与个数计算

- * 具体步骤:
- * 1、对于某特定大小的特征，在窗口内滑动计算。
例如特征大小为 2×1 ，对于 24×24 的图像，水平可滑动23步，垂直滑动24步，所以共有 23×24 个特征。
- * 2、对于一个特征，特征本身沿水平、竖直方向分别缩放。例如，还假设特征大小为 2×1 ，则沿水平方向可放大为： 4×1 ， 6×1 ， 8×1 ，...， 24×1 ；
竖直方向可放大为： 2×1 ， 2×2 ， 2×3 ，...， 2×24 。
即每个特征有 XY 种放大方式。

Haar特征构造与个数计算

* 因此特征个数的公式为：


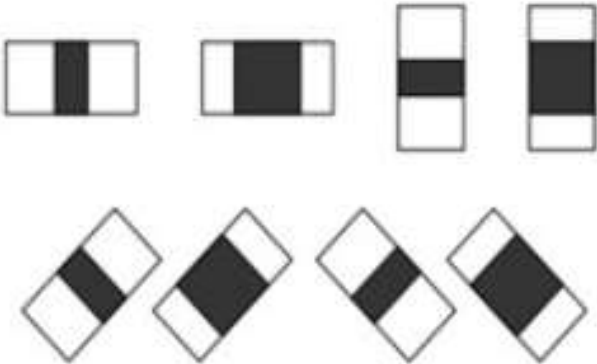

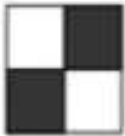
$$XY(W + 1 - w \frac{X + 1}{2})(H + 1 - h \frac{Y + 1}{2})$$

其中 $W \times H$ 为图片大小， $w \times h$ 为矩形特征大小， $X = \left\lfloor \frac{W}{w} \right\rfloor$ ， $Y = \left\lfloor \frac{H}{h} \right\rfloor$ 表示矩形特征在水平和垂直方向的能放大的最大比例系数

Haar-like 矩形特征拓展

- * 对**Haar-like**矩形特征库作了进一步扩展，加入了旋转45°的矩形特征。扩展后的特征大致分为4种类型：边缘特征、线性特征、中心环绕特征和对角线特征

Haar-like 矩形特征拓展

	
1.边缘特征(Edge features)	2、线性特征(Line features)
	
3、圆心环绕特征 (Center-surround features)	4、特定方向的特征 (Special diagonal line) feature used

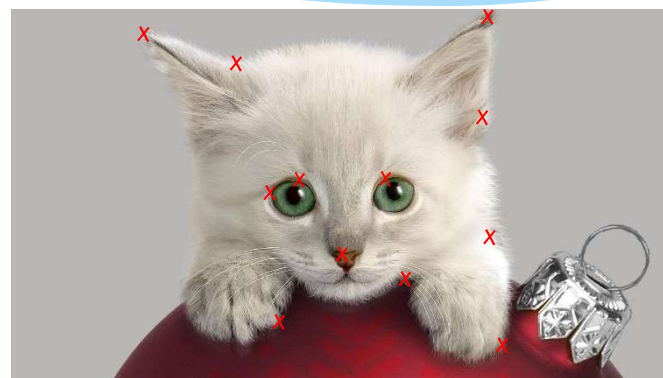
图像特征 VS 图像局部特征点



图像特征LBP、HOG、Haar...

反应了全局的纹理特征、主要物体形状等，容易受到光照、旋转、噪声等影响

识别、分类、检测任务



局部特征点

图像中蕴含数量丰富，反应图像上具有的特殊局部性，受到的干扰较少

图像匹配、检索等任务

常见图像局部描述子

- * **SIFT** (Scale-invariant feature transform) 描述子
- * **SURF** (Speeded Up Robust Features) 描述子
- * **DAISY** 描述子
- *

SIFT (Scale-invariant feature transform) 描述子

- * 在诸多的局部图像特征描述子中，SIFT是其中应用最广的，它在1999年首次提出，至2004年得到完善。
- * SIFT的提出是局部图像特征描述子研究领域一项里程碑式的工作。
- * 由于SIFT对尺度、旋转以及一定视角和光照变化等图像变化都具有不变性，并且SIFT具有很强的可区分性，自它提出以来，很快在物体识别、宽基线图像匹配、三维重建、图像检索中得到了应用，局部图像特征描述子在计算机视觉领域内也得到了更加广泛的关注，涌现了一大批各具特色的局部图像特征描述子

SIFT 算法的步骤

- (1) 构造高斯差分尺度空间
- (2) 检测尺度空间极值点
- (3) 精确定位极值点，去除不好的特征点
- (4) 给特征点赋值一个128维的方向参数。每个关键点都包含三个信息：位置、尺度和方向。
- (5) 关键点描述子的生成

SIFT 算法的步骤

- * 尺度空间的生成

- * 尺度空间的目的是模拟图像数据的多尺度特性，一幅二维图形的尺度空间定义为：

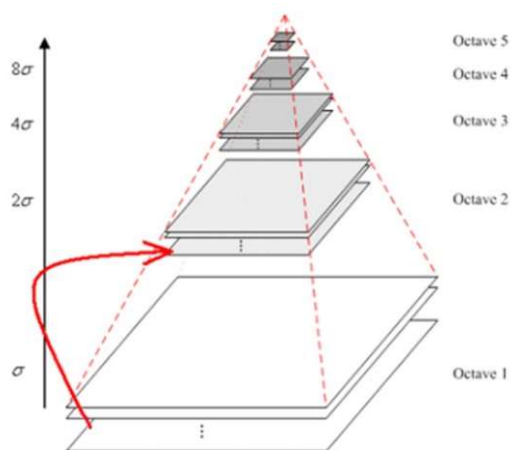
$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

- * 其中 $G(x, y, \sigma)$ 是尺度可变高斯函数 $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$, (x, y) 是空间坐标, σ 是尺度坐标, 决定图像的平滑程度, 大的 σ 值对应大尺度低分辨率。

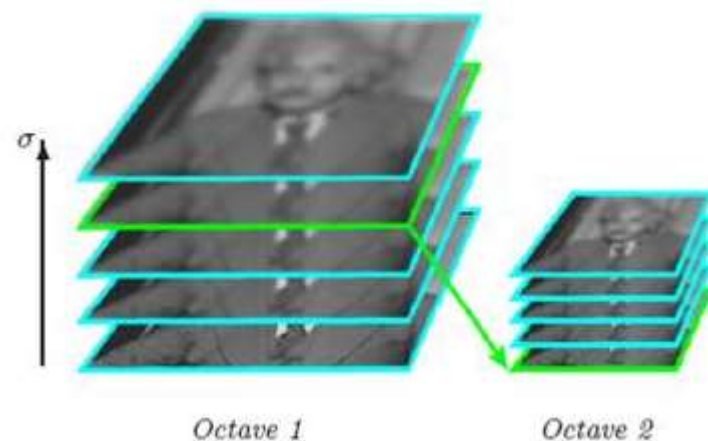
SIFT 算法的步骤

* 高斯金字塔的构建

- * 在对图像做不同尺度的高斯模糊后，再对图像做降采样并由大到小，从下到上构成塔状模型



高斯金字塔



高斯金字塔的另一种表示

SIFT 算法的步骤

- * 高斯差分金字塔

- * 构建尺度空间的目的是为了检测出在不同的尺度下都存在的特征点，而检测特征点较好的算子是 $\Delta^2 G$ （高斯拉普拉斯，LoG），

其中 $\Delta^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ 。

SIFT 算法的步骤

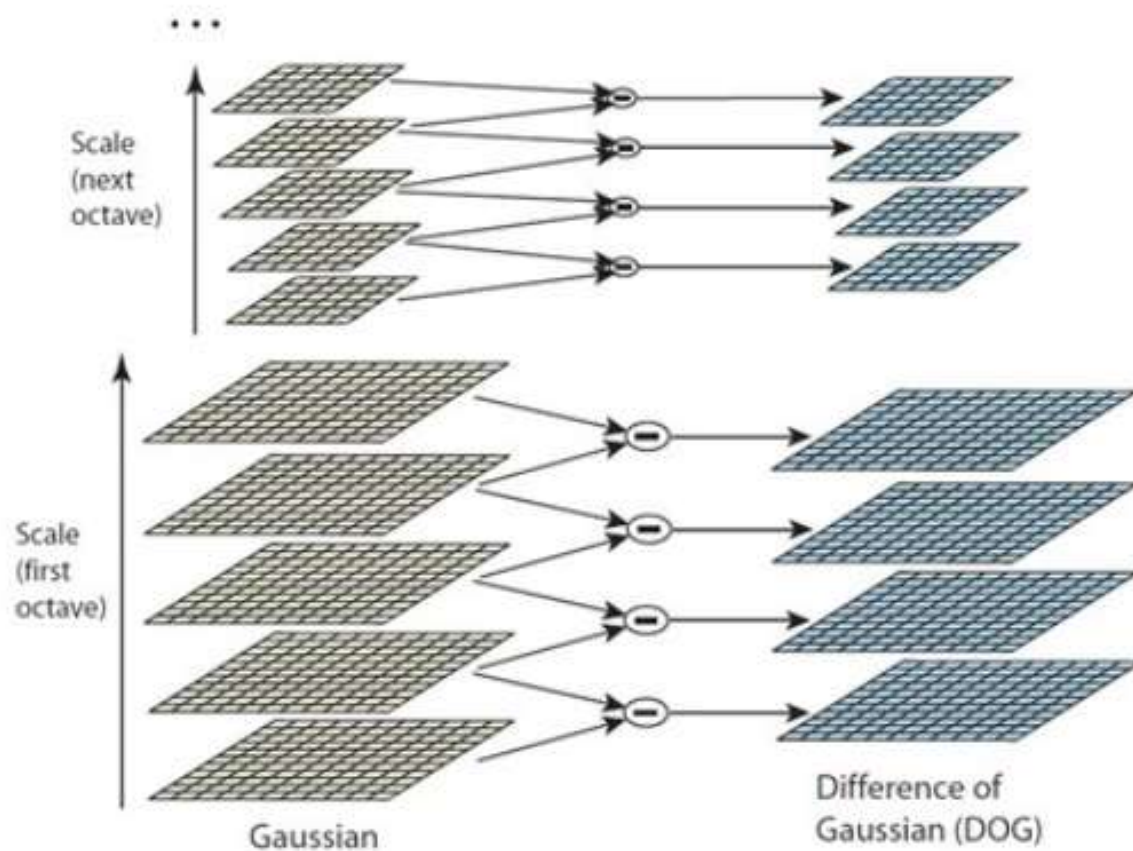
- * 高斯差分金字塔

- * 为了减少计算量，通常使用**DoG**（差分高斯）来近似计算，设 **k** 为相邻两个高斯尺度空间的比例因子，则高斯差分尺度空间定义如下

$$\begin{aligned} D(x, y, \sigma) &= [G(x, y, k\sigma) - G(x, y, \sigma)] * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned}$$

- * 其中 $L(x, y, \sigma)$ 是图像的高斯尺度空间

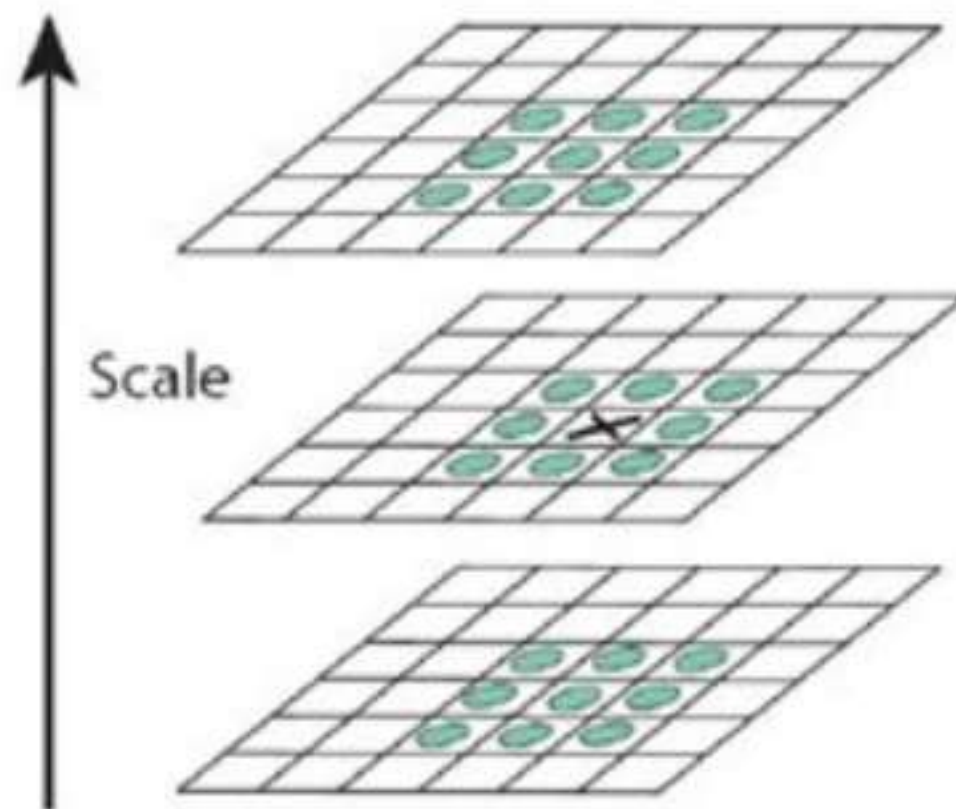
SIFT 算法的步骤



SIFT 算法的步骤

- * 检测**DoG**尺度空间极值点
 - * 为了寻找尺度空间的极值点，每一个采样点要和它同尺度的8个相邻点和上下相邻尺度对应的 9×2 个点共26个点比较，以确保在尺度空间和二维图像空间都检测到极值点。一个点如果在**DoG**尺度空间本层以及上下两层的26个领域中是最大或最小值时，就认为该点是图像在该尺度下的一个特征点,如图所示。

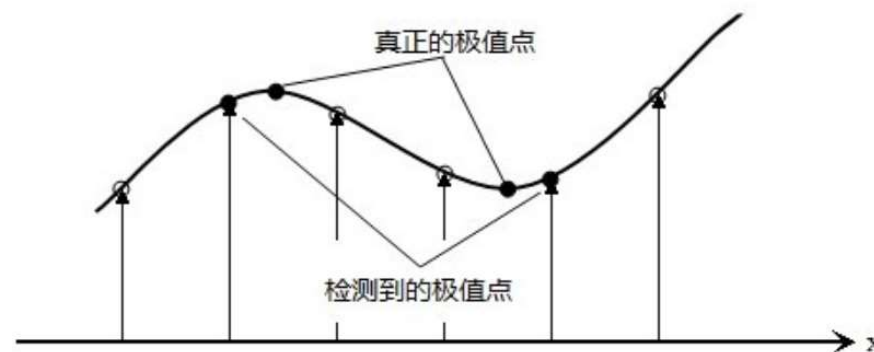
SIFT 算法的步骤



SIFT 算法的步骤

- * 精确定位极值点

- * 离散空间的极值点并不是真正的极值点，下图显示了二维函数离散空间得到的极值点与连续空间极值点的差别。利用已知的离散空间点插值得到的连续空间极值点的方法叫做子像素插值（Sub-pixel Interpolation）。



SIFT 算法的步骤

- * 精确定位极值点

- * 1. 通过对尺度空间DoG函数曲线拟合以精确确定关键点的位置和尺度（达到亚像素精度）

利用DoG函数在尺度空间的Taylor展开式(拟合

函数)为 $D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X$ ，其中

$X = (x, y, \sigma)^T$ 。求导并令其为0，得到精确的

位置 $\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x}$ 。

SIFT 算法的步骤

- * 精确定位极值点

- * 2. 在已检测到的特征点中，要去掉低对比度的特征点。

$|D(\hat{x})|$ 过小的点易受噪声的干扰而变得不稳定，

带入 $\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x}$ ，得到 $D(\hat{x}) = D(x, y, \sigma) +$

$\frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x}$ 。若 $|D(\hat{x})| \geq 0.03$ ，该特征点保留，否则丢弃。

SIFT 算法的步骤

- * 精确定位极值点

- * 3. 消除边缘响应

一个定义不好的高斯差分算子的极值在横跨边缘的地方有较大的主曲率，而在垂直边缘的方向有较小的主曲率。主曲率通过一个2x2 的 **Hessian** 矩阵 **H** 求出

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

通过对近邻样本点的差分来估计导数值。

SIFT 算法的步骤

- * 精确定位极值点

- * 3. 消除边缘响应

H的特征值与**D**的主曲率成比例。设 α 为最大量级的特征值， β 为最小量级的特征值，则

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta$$

$$Det(H) = D_{xx} + D_{yy} - D_{xy}^2 = \alpha \cdot \beta$$

SIFT 算法的步骤

- * 精确定位极值点

- * 3. 消除边缘响应

其中 $\text{Tr}(H)$ 为矩阵 H 的迹， $\text{Det}(H)$ 为行列式。设 $\alpha=\gamma\beta$ ，则

$$\frac{\text{Tr}(H)^2}{\text{Det}(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(\gamma\beta + \beta)^2}{\gamma\beta^2} = \frac{(\gamma + 1)^2}{\gamma}$$

SIFT 算法的步骤

- * 精确定位极值点

- * 3. 消除边缘响应

看主曲率是否高于某个阈值 T_γ

$$\frac{Tr(H)^2}{Det(H)} > \frac{(T_\gamma + 1)^2}{T_\gamma}$$

如果上式成立，则剔除特征点，否则保留。

(论文中取 $T_\gamma=10$)

SIFT 算法的步骤

- * 求取特征点主方向

- * 为了使描述符具有旋转不变性，需要利用图像的局部特征给每一个关键点分配一个基准方向。使用图像梯度的方法求取局部结构的稳定方向。
- * 对于在**DOG**金字塔中检测出的关键点，采集其所在高斯金字塔图像 **3σ** 邻域窗口内像素的梯度和方向分布特征。

SIFT 算法的步骤

- * 求取特征点主方向

- * 梯度的模值和方向如下

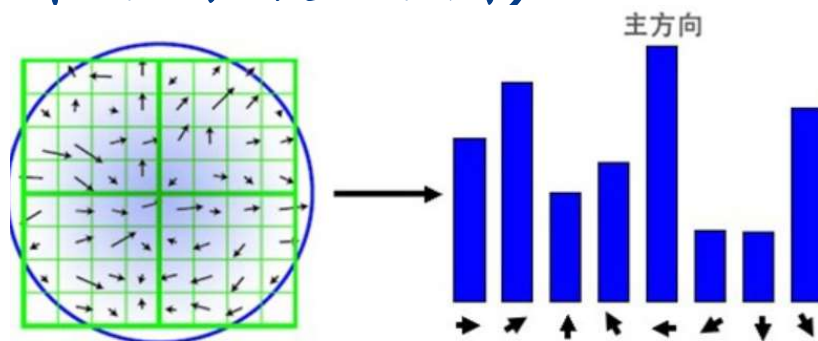
$$m(x, y) = \sqrt{[L(x+1, y) - L(x-1, y)]^2 + [L(x, y+1) - L(x, y-1)]^2}$$

$$\theta(x, y) = \arctan \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}$$

SIFT 算法的步骤

* 求取特征点主方向

- * 在完成关键点的梯度计算后，使用直方图统计邻域内像素的梯度和方向。如图所示，直方图的峰值方向代表了关键点的主方向，其他达到最大值80%的可作为辅助方向(为简化，图中只画了八个方向的直方图)。



SIFT 算法的步骤

* 关键点描述子生成

- * 通过以上的步骤已找到**SIFT**特征点位置、尺度和方向信息，下面就需要使用一组向量来描述关键点也就是生成特征点描述子，这个描述符不只包含特征点，也含有特征点周围对其有贡献的像素点。
- * 描述子应具有较高的独立性，以保证匹配率。
特征描述符的生成大致有三个步骤：
 - 1) 校正旋转主方向，确保旋转不变性。
 - 2) 生成描述子，最终形成一个128维的特征向量
 - 3) 归一化处理，将特征向量长度进行归一化处理，进一步去除光照的影响。

SIFT 算法的步骤

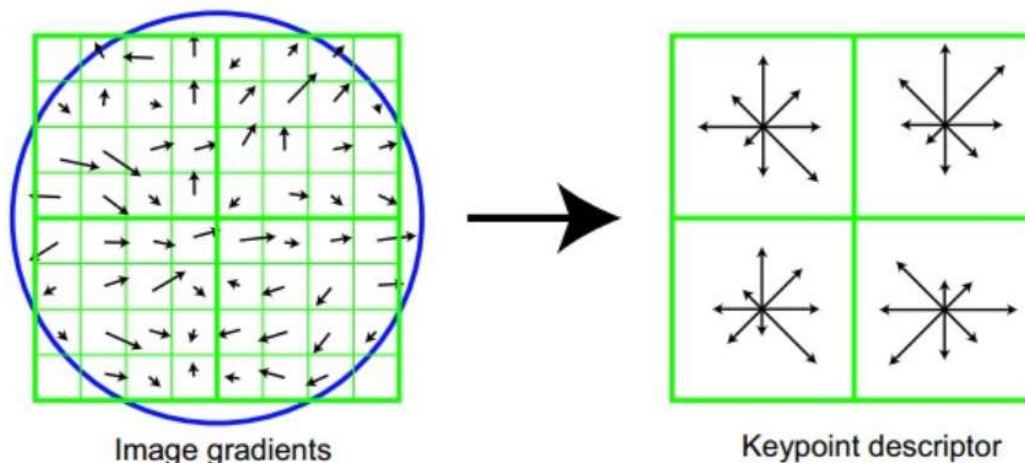
- * 关键点描述子生成

- * 为了保证特征矢量的旋转不变性，要以特征点为中心，在附件邻域内将坐标轴旋转 θ （特征点主方向）角度，旋转后的邻域内像素的新坐标为

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

SIFT 算法的步骤

- * 关键点描述子生成
 - * 旋转后以特征点为中心取 8×8 的邻域为采样窗口，将采样点与特征点的相对方向通过高斯加权后归入针对每个 4×4 小块的包含8个方向的直方图，获得 2×2 个种子点， $2 \times 2 \times 8$ 维特征向量



SIFT 算法的步骤

- * 关键点描述子生成
 - * 上述例子只给出 $2*2*8$ 维特征描述向量的例子，**实际使用中SIFT**使用 $4*4$ 个种子点（ $16*16$ 的采样区域），最终得到 $4*4*8=128$ **维**特征描述向量
 - * 对128维特征向量最后进行归一化处理得到最终的**SIFT**特征向量。

SURF特征

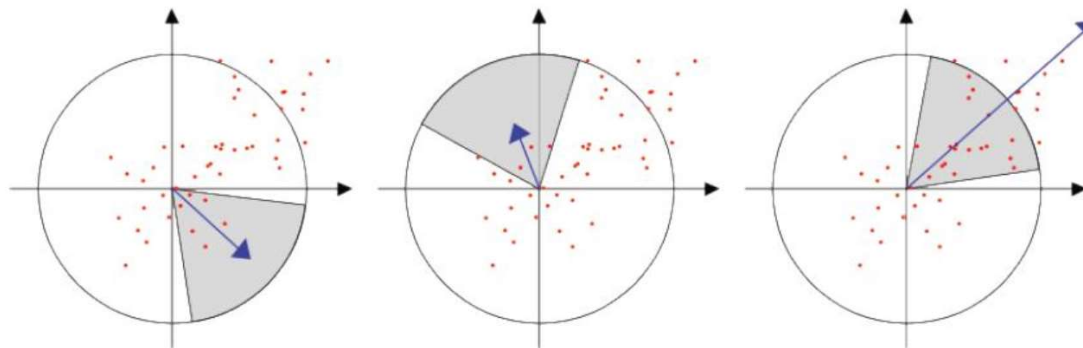
- * 经典的SIFT算法比较稳定，检测到的特征点也比较多，其最大的缺点之一是计算复杂度较高
- * SURF是SIFT角点检测算法的改进版，主要体现在速度上

SURF特征

- * 采用**Hessian**矩阵行列式构造高斯金字塔尺度空间，采用积分图计算，加快了速度
- * 统计**Haar**小波特征选取特征点主方向
 - * **SIFT**选取特征点主方向是采用在特征点领域内统计其梯度直方图，而在**SURF**中，不统计其梯度直方图，而是统计特征点领域内的**Haar**小波特征。

SURF特征

- * 统计Haar小波特征选取特征点主方向
 - * 即在特征点的领域内，统计60度扇形内所有点的水平Haar小波特征和垂直Haar小波特征总和，Haar小波的尺寸变长为 $4s$ ，这样一个扇形得到了一个值。然后60度扇形以一定间隔进行旋转，最后将最大值那个扇形的方向作为该特征点的主方向。



SURF特征

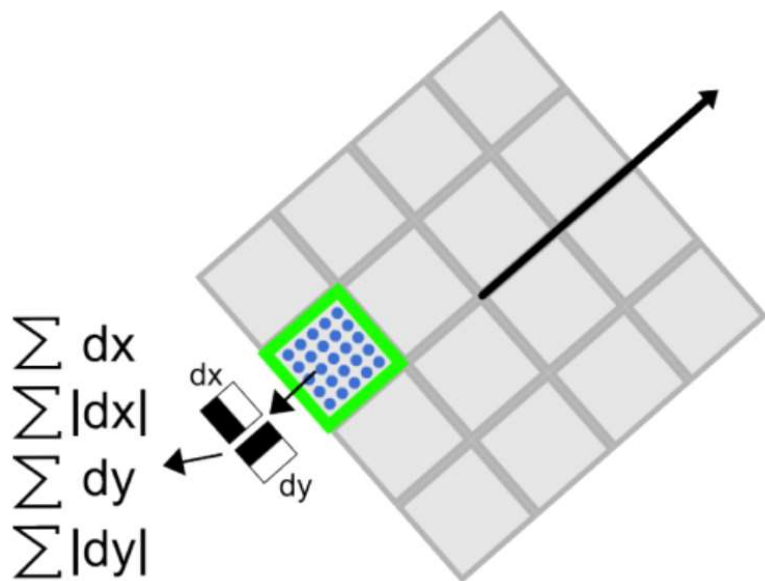
- * 采用统计Haar小波特征来生成描述算子
- * 在SIFT中，是在特征点周围取 $16*16$ 的邻域，并把该邻域化为 $4*4$ 个的小区域，每个小区域统计8个方向梯度，最后得到 $4*4*8=128$ 维的向量，该向量作为该点的SIFT描述子。

SURF特征

- * 采用统计Haar小波特征来生成描述算子
 - * 在SURF中，也是在特征点周围取一个正方形框，框的边长为 $20s$ (s 是所检测到该特征点所在的尺度)。该框方向为主方向。
 - * 然后把该框分为16个子区域，每个子区域统计25个像素的水平方向和垂直方向的Haar小波特征，这里的水平和垂直方向都是相对主方向而言的。
 - * 该Haar小波特征为水平方向值之和，水平方向绝对值之和，垂直方向之和，垂直方向绝对值之和。

SURF特征

- * 采用统计Haar小波特征来生成描述算子
- * 该过程的示意图如下所示

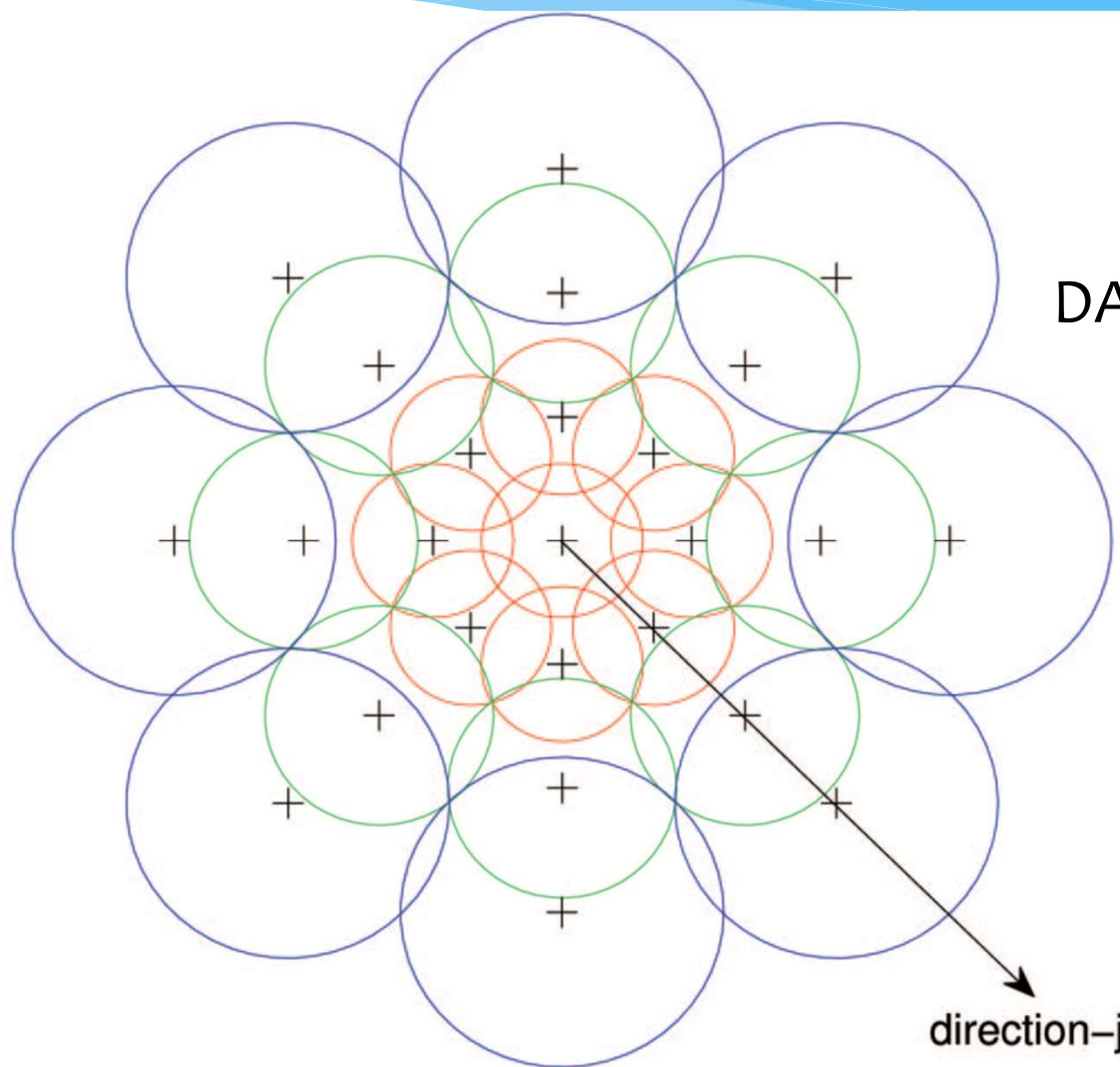


这样每个小区域就有4个值，
所以每个特征点就是 $16 \times 4 = 64$
维的向量，相比SIFT而言，少
了一半，这在特征匹配过程中
会大大加快匹配速度

DAISY特征

- * **DAISY**是面向稠密特征提取的可快速计算的局部图像特征描述子，它本质思想和**SIFT**是一样的：分块统计梯度方向直方图。
- * 不同的是，**DAISY**在分块策略上进行了改进，利用高斯卷积来进行梯度方向直方图的分块汇聚，这样利用高斯卷积的可快速计算性就可以快速稠密地进行特征描述子的提取。
- * 比较巧合的是，**DAISY**这种特征汇聚策略被一些研究者通过机器学习的方法证明相对于其他几种特征汇聚策略（卡迪尔坐标下分块、极坐标下分块）是最优的

DAISY特征

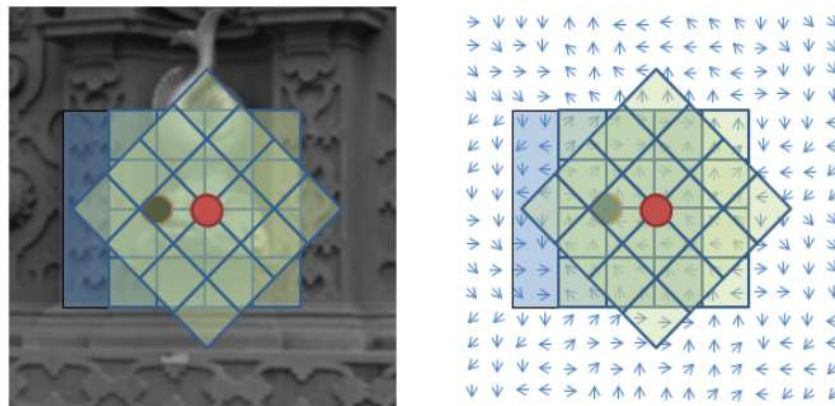


DAISY的采样特征点选择

DAISY特征

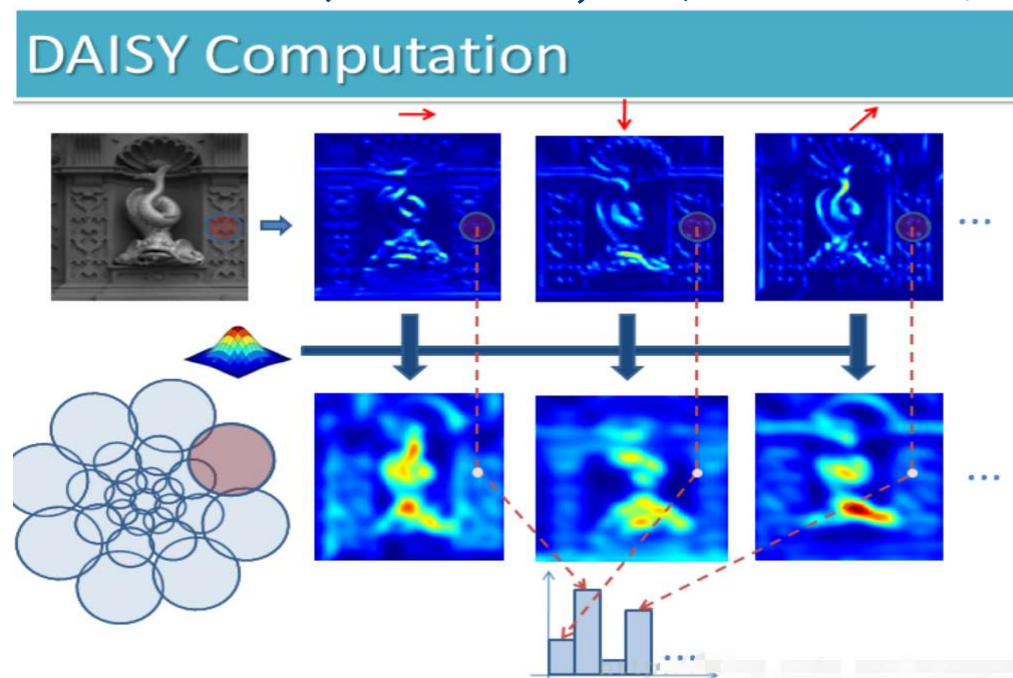
- * 传统的**SIFT**方法，使用了加权的梯度幅值直方图，当待计算点平移或旋转，直方图的数值也会随之改变，在提取稠密特征的时候就需要反复计算梯度直方图，耗时很长。

SIFT Computation



DAISY特征

- * **DAISY**算子使用的是高斯核卷积的方法来对梯度直方图加权，由于高斯核有各向同性，对于图像的旋转，结果体现到直方图上只是**bin**的顺序的改变，因此不需要重复计算梯度直方图。



谢谢