

# 第十章：图像识别—目标检测

---

中国科学技术大学  
电子工程与信息科学系



# 目标检测

---

- 简单形状的检测
  - 霍夫变换 (Hough Transform)
  - 倒角距离变换 (Chamfer Distance Transform)
- 人脸识别与检测
- 一般目标检测

# 霍夫变换 (Hough Transform)

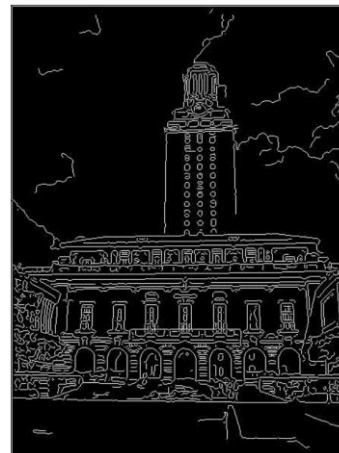
## □ 应用场景：直线拟合



如何从边缘检测的结果得到直线拟合的结果？

## □ 直线拟合的难点

- 边缘检测点杂乱且多余
- 不同的检测点属于不同的直线
- 部分线段可能漏检
- 边缘检测点上存在噪声





# 霍夫变换 (Hough Transform)

- 基本思想：基于投票的机制
  - 待检测形状为一个模型(model)
  - 让每个样本（像素）对所有与其兼容的模型进行投票
  - 假设噪声样本不会偏好任何单个模型
  - 即使有部分样别缺失，当有足够的样本保留时，仍可将目标模型（形状）检测出来

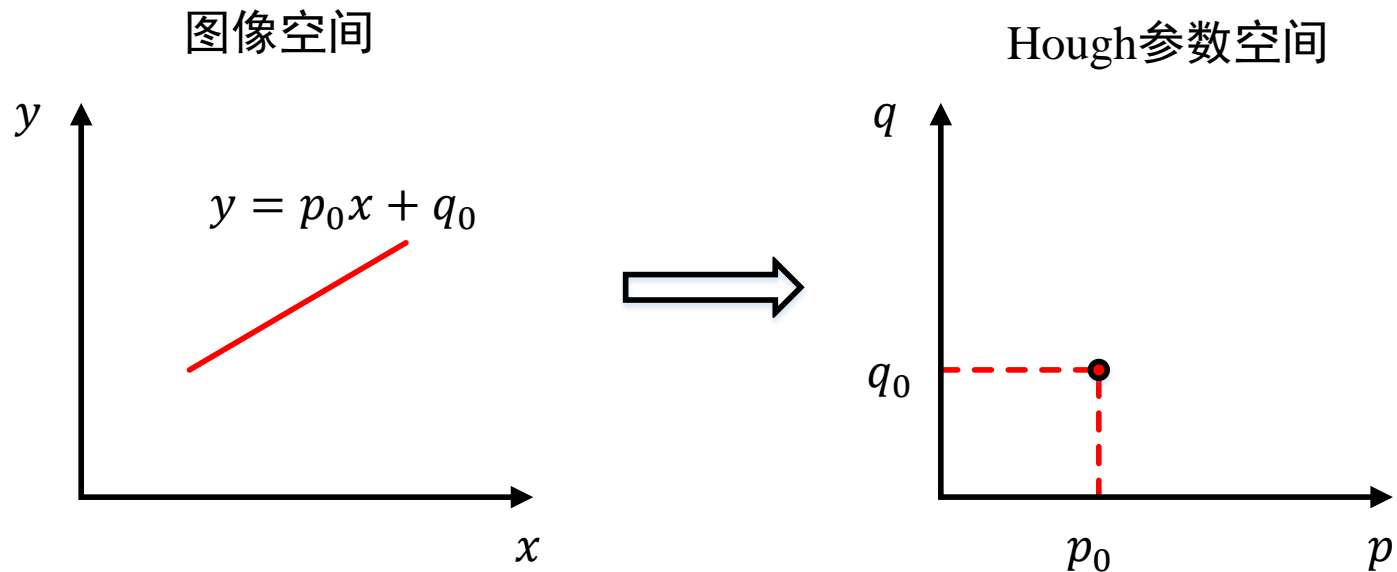
# 霍夫变换基本原理：以直线检测为例

## □ Hough Transform

- 图像空间与参数空间之间的一种变换

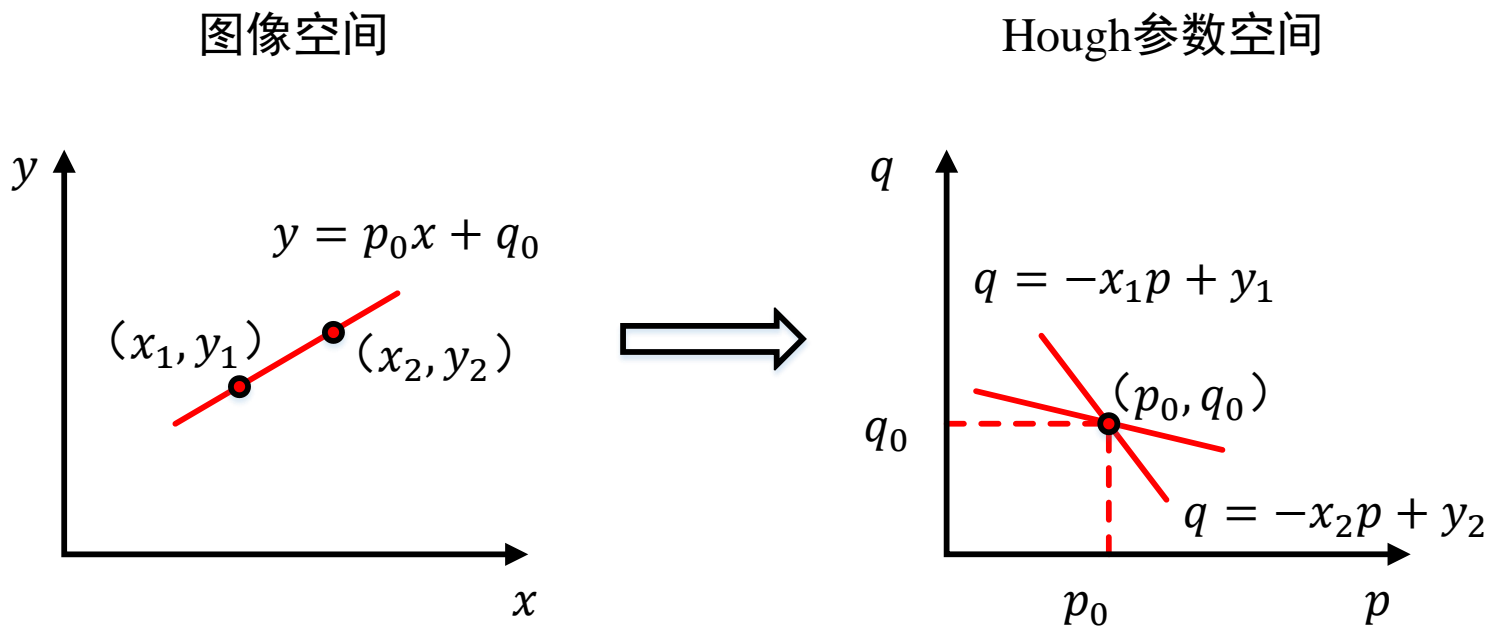
## □ Hough参数空间

- 在图像空间中的一条直线，对应于Hough参数空间中的一个点



# 霍夫变换：直线检测

- 霍夫参数空间的一条直线对应图像空间的一个点
- 图像空间中同一条直线上的任意两个点，在参数空间中对应于两条相交的直线
  - 即在图像空间共线的 $n$ 个点，对应于参数空间 $n$ 条共点 $(p_0, q_0)$ 的直线



# 霍夫变换：直线检测

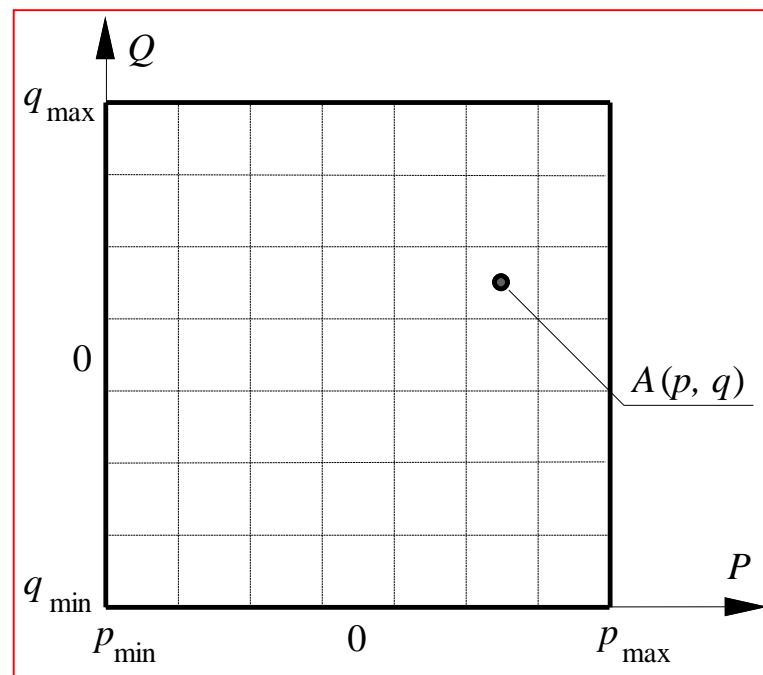
## □ 具体方法

- 将参数空间离散成一个2-D的累加数组 $A(p, q)$

$$p \in [p_{\min}, p_{\max}]$$

$$q \in [q_{\min}, q_{\max}]$$

- $A(p, q) = A(p, q) + 1$   
 $A(p, q)$ : 共线点数  
 $(p, q)$ : 直线方程参数



## □ 潜在的问题

- $p_{\max}/q_{\max}$ 可能为无穷大，难以对其进行离散化

# 霍夫变换：直线检测的改进形式

## □ 直线的极坐标方程

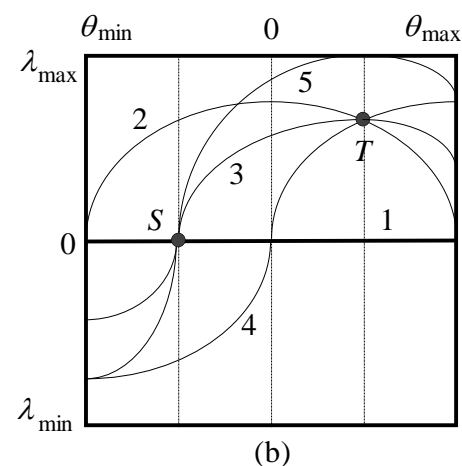
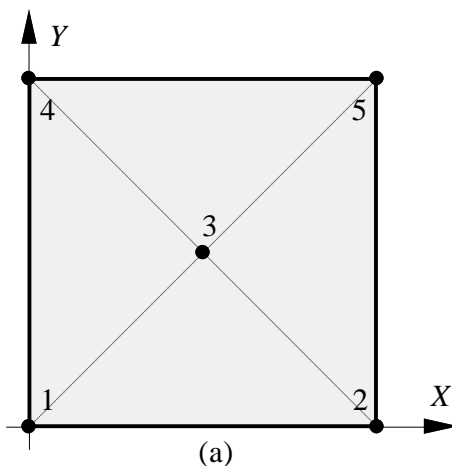
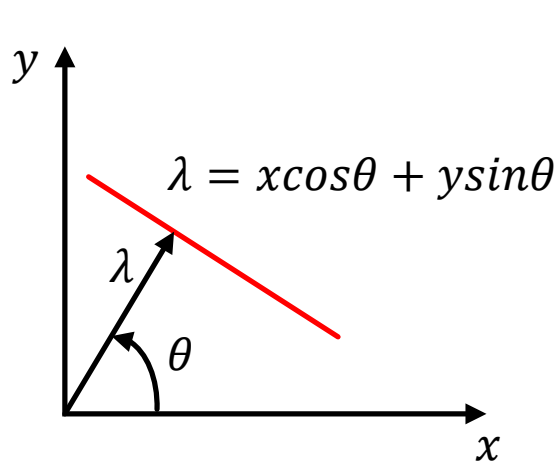
$$\lambda = x \cos \theta + y \sin \theta$$

- 参数 $\lambda$ 和 $\theta$ 唯一确定一条直线

## □ X-Y平面的一个点对应参数空间的一条正弦曲线

- $\lambda = x_0 \cos \theta + y_0 \sin \theta \leftrightarrow \lambda = A \sin(\theta + \alpha)$

$$\text{其中 } \alpha = \tan^{-1} \left( \frac{x_0}{y_0} \right), A = \sqrt{x_0^2 + y_0^2}$$





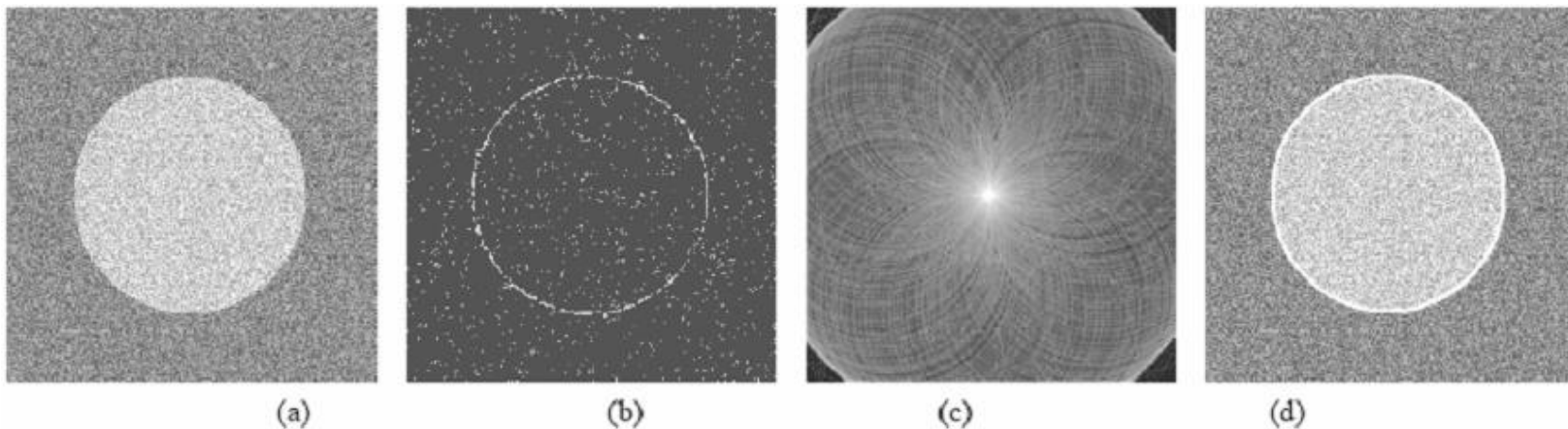


# 霍夫变换：其他形状的检测

- 对于满足解析式  $f(x, c) = 0$  形式的各类曲线，Hough Transform 可以将其检测出来，并把曲线上的点完整地连接起来
  
- 示例：以圆周检测为例
  - 圆周方程：  $(x - a)^2 + (y - b)^2 = r^2$
  - 三个参数  $a, b, r$ ，所以需要在参数空间中建立3-D累加数组，其中的元素可以记为  $A(a, b, r)$

# 霍夫变换：圆周检测

## □ 示例



图(a)为256x256，灰度256级，叠加随机噪声；

图(b)为求梯度(Sobel算子)取阈值后的结果；

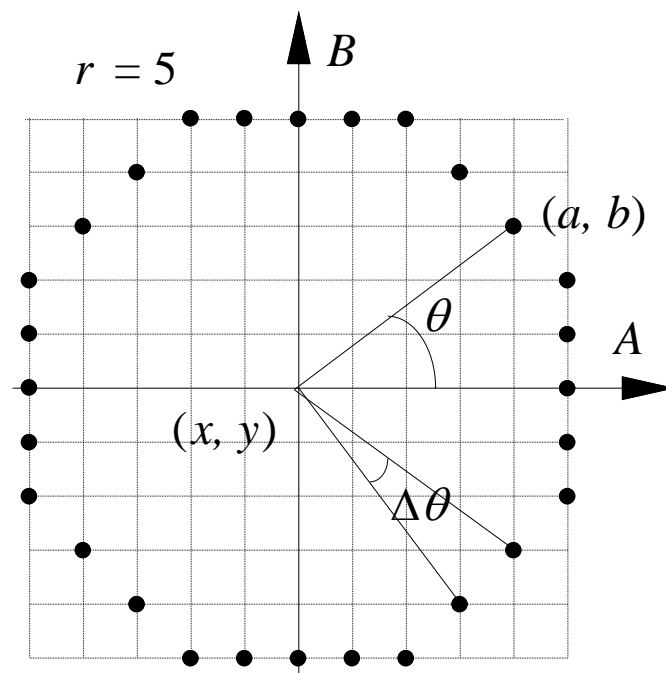
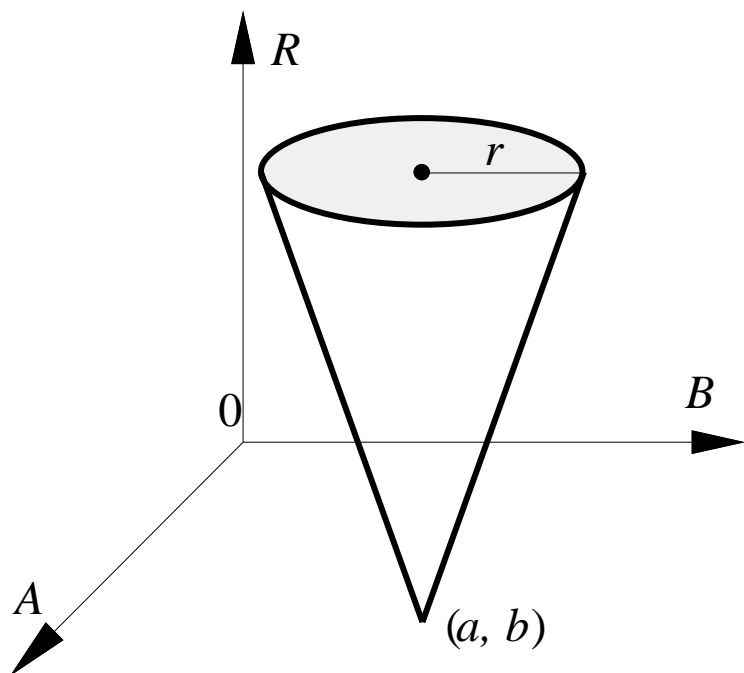
图(c)哈夫变换累记器图；

图(d)为检测出的圆周附加在原图上的效果

# 霍夫变换：圆周检测

## □ 利用梯度降维

- 使累加数组的维度减少一维
- 圆周——圆周对偶性



# 霍夫变换：圆周检测

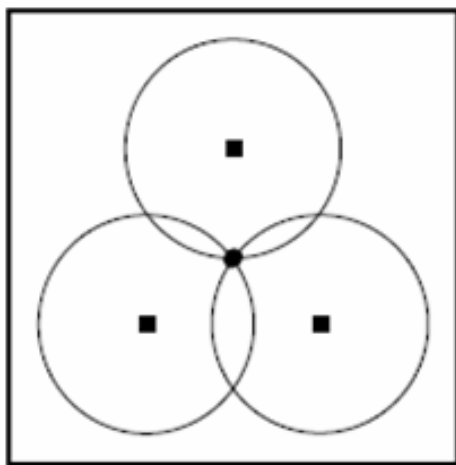
## □ 利用梯度降维

- 圆周圆心在圆周边缘点的梯度方向上

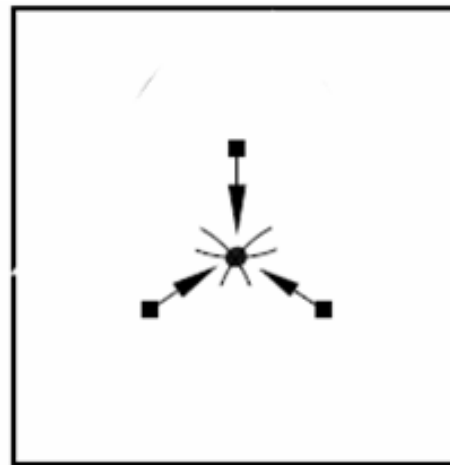
$$a = x - r \sin \theta$$

$$b = y + r \cos \theta$$

1个3-D累加器数组 → 2个2-D累加器数组



(a)



(b)

利用梯度与否 2 种情况下的累加数组示意



# 霍夫变换：利用梯度信息检测椭圆

## □ 椭圆方程

$$\frac{(x-p)^2}{a^2} + \frac{(y-q)^2}{b^2} = 1$$

## □ 对x求导

$$\frac{(x-p)}{a^2} + \frac{(y-q)}{b^2} \tan \theta = 0$$

## □ 联立得到

$$p = x \pm \frac{a^2 \tan \theta}{\sqrt{a^2 \tan^2 \theta + b^2}} \quad q = y \pm \frac{b^2}{\sqrt{a^2 \tan^2 \theta + b^2}}$$

建立2个3-D累加数组 $A_x(p, a, b)$ 和 $A_y(p, a, b)$



# 广义霍夫变换

---

- 广义Hough变换将一般的模板匹配与**Hough变换**相结合
- 先对模板与图象上的物点作**坐标变换**，然后求相关
- 并用类似Hough变换检测物体的表决方法来确定匹配点

# 广义霍夫变换原理

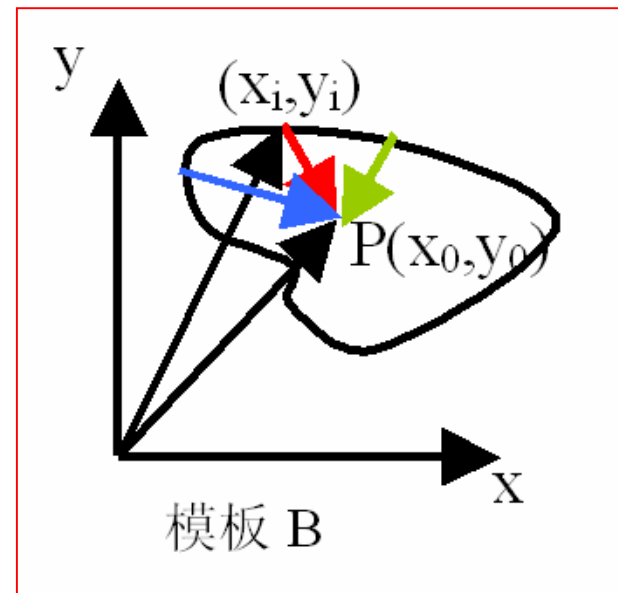
- B为模板物体的一组点集
- $P(x_0, y_0)$ 为一参考点，常把P取为B的中心点。
- 广义Hough变换是一组矢量的集合： $B$ 
  - $B = \{(dx_i, dy_i), i = 1, 2, \dots, m\}$

$$H(B, P)$$

$$\{(dx_i, dy_i), i = 1, 2, \dots, n\}$$

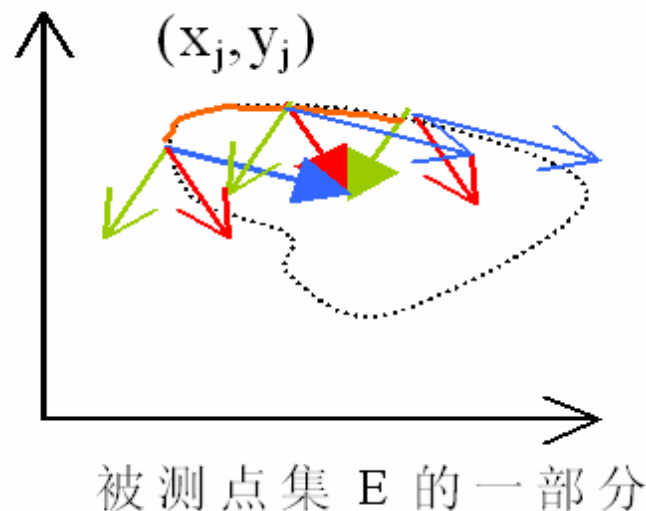
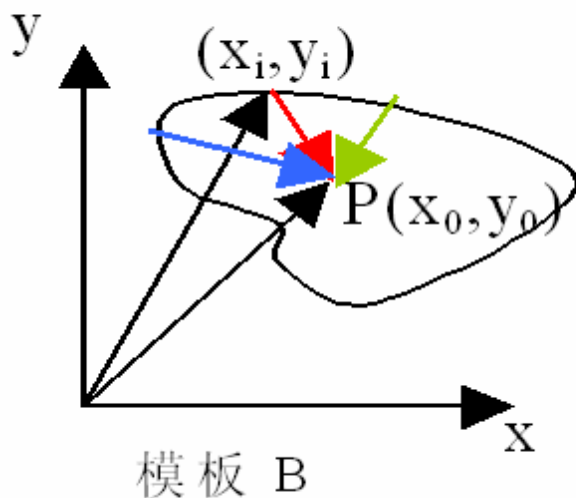
$$dx_i = x_0 - x_i = -(x_i - x_0)$$

$$dy_i = y_0 - y_i = -(y_i - y_0)$$



# 广义霍夫变换原理

- 检测时，将待测图象，记为点集 $E$ ：
  - $E = \{(x_j, y_j), j = 1, 2, \dots, n\}$
- 将待检测点集与变换后的模板 $B$ 做相关运算
  - $\forall i, j$ , 计算 $v(i, j) = (dx_i + x_j, dy_i + y_j)$ ,
  - 如果有很多点 $v(i, j)$ 都对应同一个坐标点，则该点为与 $B$ 相匹配的形狀的中心位置





# 广义霍夫变换原理

- 在所需检测的曲线或目标轮廓没有或不易用解析式表达时，可以利用**表格**来建立**曲线或轮廓点**与**参考点**间的关系，从而可继续利用哈夫变换进行检测

建立参考点与轮廓点的联系：

$$p = x + r(\theta) \cdot \cos(\phi(\theta))$$

$$q = y + r(\theta) \cdot \sin(\phi(\theta))$$

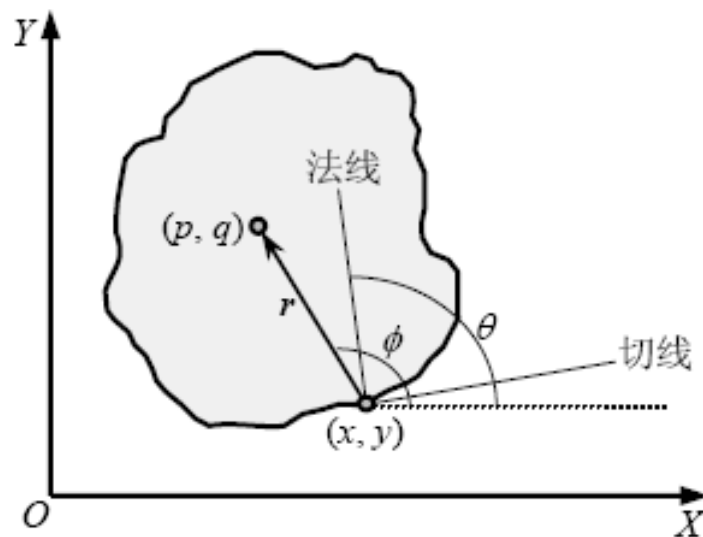


图 6.1.8 建立参考点和轮廓点的对应关系

# 广义霍夫变换原理

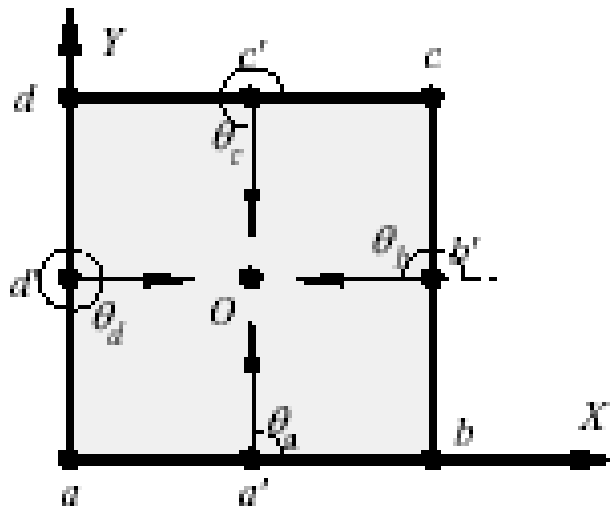
- 已知轮廓形状、朝向和尺度而只需检测位置信息
- 根据 $\theta, r$ 和 $\phi$  的函数关系作出参考表

梯度角 $\theta$	矢径 $r(\theta)$	矢角 $\phi(\theta)$
$\theta_1$	$r_1^1, r_1^2, \dots, r_1^{N_1}$	$\phi_1^1, \phi_1^2, \dots, \phi_1^{N_1}$
$\theta_2$	$r_2^1, r_2^2, \dots, r_2^{N_2}$	$\phi_2^1, \phi_2^2, \dots, \phi_2^{N_2}$
...	...	...
$\theta_M$	$r_M^1, r_M^2, \dots, r_M^{N_M}$	$\phi_M^1, \phi_M^2, \dots, \phi_M^{N_M}$

- 给定一个测试点 $(x', y')$  及其梯度角 $\theta'$ ，即可确定一组可能的参考点位置
  - 根据梯度角 $\theta'$ ，找到表中对应梯度角所在行的矢径和视角序列
  - 基于坐标 $(x', y')$ 和序列中每一组 $(r, \phi)$ 值，反推出形状参考点坐标 $p' = x' + r(\theta) \cdot \cos(\phi(\theta)), q' = y' + r(\theta) \cdot \sin(\phi(\theta))$

# 广义霍夫变换原理

轮廓点	$a$	$a'$	$b$	$b'$	$C$	$c'$	$d$	$d'$
矢径 $r(\theta)$	$\sqrt{2}/2$	$1/2$	$\sqrt{2}/2$	$1/2$	$\sqrt{2}/2$	$1/2$	$\sqrt{2}/2$	$1/2$
矢角 $\phi(\theta)$	$1\pi/4$	$2\pi/4$	$3\pi/4$	$4\pi/4$	$5\pi/4$	$6\pi/4$	$7\pi/4$	$8\pi/4$

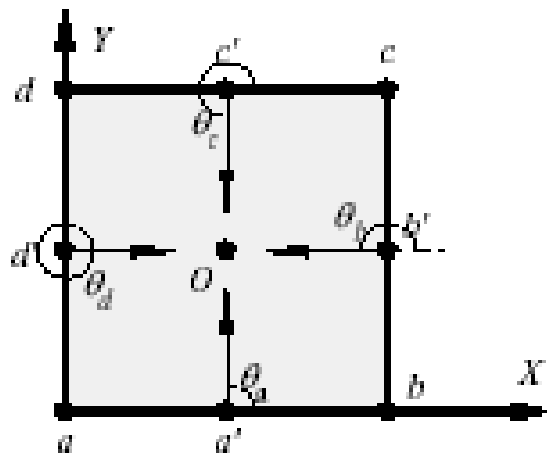


梯度角 $\theta$	矢径 $r(\theta)$		矢角 $\phi(\theta)$	
$\theta_a = \pi/2$	$\sqrt{2}/2$	$1/2$	$\pi/4$	$2\pi/4$
$\theta_b = 2\pi/2$	$\sqrt{2}/2$	$1/2$	$3\pi/4$	$4\pi/4$
$\theta_c = 3\pi/2$	$\sqrt{2}/2$	$1/2$	$5\pi/4$	$6\pi/4$
$\theta_d = 4\pi/2$	$\sqrt{2}/2$	$1/2$	$7\pi/4$	$8\pi/4$

# 广义霍夫变换原理

- 利用正方形上的8个轮廓点判断可能参考点位置( $p', q'$ )
- 对每个 $\theta$ 有 2个  $r$  及2个  $\phi$  与之对应

$$\begin{aligned} p' &= x' + r(\theta) \cdot \cos(\phi(\theta)) \\ q' &= y' + r(\theta) \cdot \sin(\phi(\theta)) \end{aligned}$$



梯度角	轮廓点	可能参考点		轮廓点	可能参考点	
$\theta_a$	$a$	$O$	$d'$	$a'$	$b'$	$O$
$\theta_b$	$b$	$O$	$a'$	$b'$	$c'$	$O$
$\theta_c$	$c$	$O$	$b'$	$c'$	$d'$	$O$
$\theta_d$	$d$	$O$	$c'$	$d'$	$a'$	$O$

点 $O$ 出现频率最高



# 广义霍夫变换的性能

---

- ☐ 运算量较小
- ☐ 抗干扰性也较强
- ☐ 可以求出曲线的某些参数
- ☐ 可适用于不规则曲线
- ☐ 仍不具有不变性

# 完整广义霍夫变换

□ 轮廓的平移 + 轮廓放缩、旋转

□ 累加数组：

□  $A(p_{\min}: p_{\max}, q_{\min}: q_{\max}, \beta_{\min}: \beta_{\max}, S_{\min}: S_{\max})$

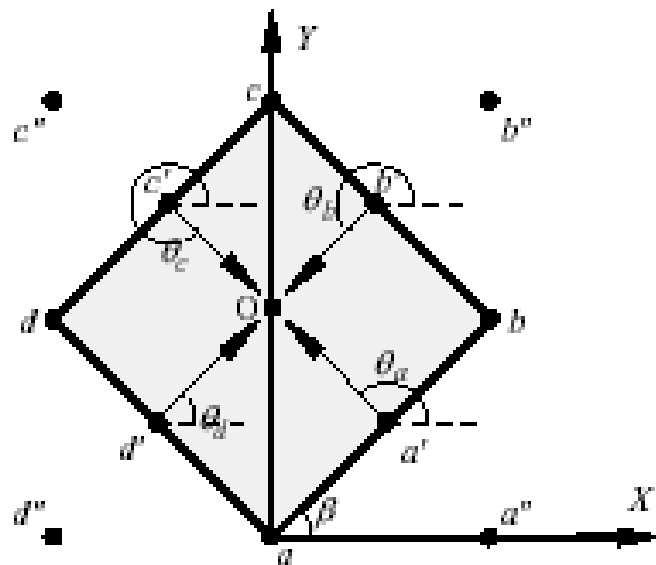
$$p = x + S \times r(\theta) \times \cos[\phi(\theta) + \beta]$$

$$q = y + S \times r(\theta) \times \sin[\phi(\theta) + \beta]$$

□ 累加数组的累加：  $A(p, q, \beta, S) = A(p, q, \beta, S) + 1$

# 完整广义霍夫变换

## 计算示例



原梯度角 $\theta$	新梯度角 $\theta'$	矢径 $r(\theta)$	新矢角 $\phi(\theta)$
$\theta_a = \pi/2$	$\theta'_a = 3\pi/4$	$\sqrt{2}/2$ 1/2	$2\pi/4$ $3\pi/4$
$\theta_b = 2\pi/2$	$\theta'_b = 5\pi/4$	$\sqrt{2}/2$ 1/2	$4\pi/4$ $5\pi/4$
$\theta_c = 3\pi/2$	$\theta'_c = 7\pi/4$	$\sqrt{2}/2$ 1/2	$6\pi/4$ $7\pi/4$
$\theta_d = 4\pi/2$	$\theta'_d = \pi/4$	$\sqrt{2}/2$ 1/2	$8\pi/4$ $1\pi/4$

梯度角	轮廓点	可能参考点	轮廓点	可能参考点
$\theta'_a$	$a$	$O$	$d'$	$a'$ $b'$ $O$
$\theta'_b$	$b$	$O$	$a'$	$b'$ $c'$ $O$
$\theta'_c$	$c$	$O$	$b'$	$c'$ $d'$ $O$
$\theta'_d$	$d$	$O$	$c'$	$d'$ $a'$ $O$



# 目标检测

---

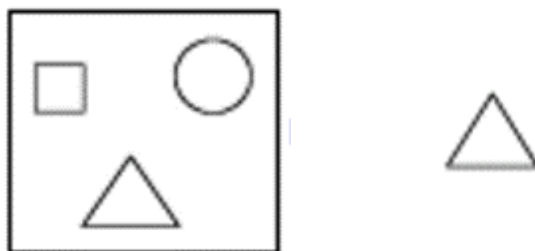
- 简单形状的检测
  - 霍夫变换 (Hough Transform)
  - 倒角距离变换 (Chamfer Distance Transform)
- 人脸识别与检测
- 目标检测



# 倒角距离变换 (Chamfer Distance Transform)

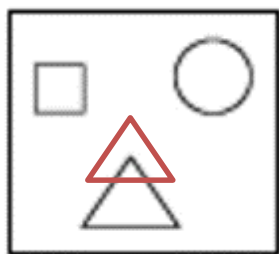
## □ 从一个问题出发

- 如何从下面左图中检测匹配右图所示的三角形形状？



## □ 基本思路：

- 把右图作为模板，将其中心置于左图所有可能的像素位置
- 对于每个放置位置 $x$ ，计算模板形状与测试图像边缘的匹配距离



$$D_{chamfer}(x) = \frac{1}{|T|} \sum_{t \in T} d_I(t + x)$$

- $T$  是模板形状（像素点的集合）
- $I$  是待匹配的边缘图像（像素点的集合）
- $d_I(t)$  是模板中的点  $t$  到  $I$  中所有边缘点的最小距离

- 将最小匹配距离 所对应的位置  $x$ ，视为匹配位置

# 倒角距离变换 (Chamfer Distance Transform)

## □ 计算复杂度分析

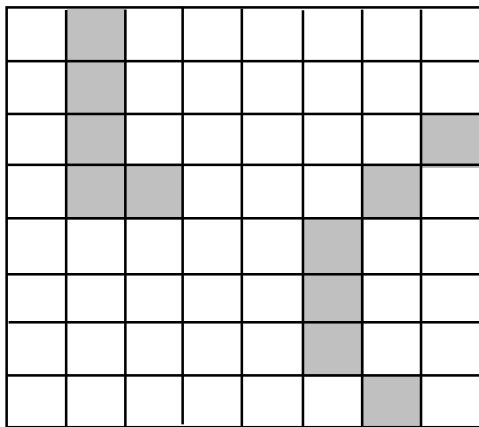
- 假设测试图像中边缘点数量为 $M$ 、像素点数量为 $P$ ，模板中边缘点数量为 $N$ ，上述距离计算的复杂度为 $O(MNP)$
- 上面距离测度存在大量计算冗余

## □ 如何降低冗余？

- 事先计算测试图像中边缘点的距离变换
- 图  $I$  中的边缘点集合  $E$ ，像素点  $p$  的距离变换结果为：

$DF_I(p) = \min_{x \in E} \text{dist}(p, x)$ ，其中  $\text{dist}(\cdot, \cdot)$  表示距离函数，如棋盘距离

边缘图



距离变换结果

1	0	1	2	3	4	3	2
1	0	1	2	3	3	2	1
1	0	1	2	3	2	1	0
1	0	0	1	2	1	0	1
2	1	1	2	1	0	1	2
3	2	2	2	1	0	1	2
4	3	3	2	1	0	1	2
5	4	4	3	2	1	0	1

# 倒角距离变换 (Chamfer Distance Transform)

## □ 倒角距离变换:

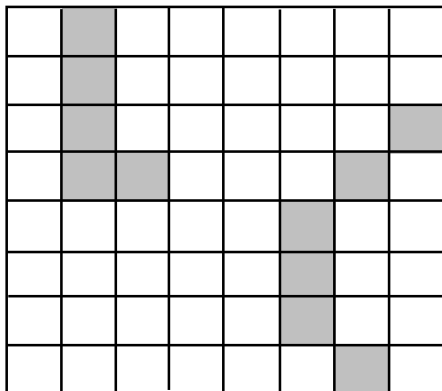
- 基于  $DF_I$ , 对模板形状进行倒角匹配(Chamfer matching)

$$D_{chamfer}(\mathbf{x}) = \frac{1}{|T|} \sum_{t \in T} d_I(\mathbf{t} + \mathbf{x}) = \frac{1}{|T|} \sum_{t \in T} DF_I(\mathbf{t} + \mathbf{x})$$

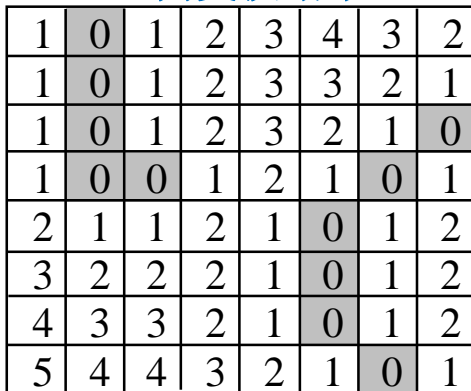
- $T$  是模板形状 (像素点的集合)
- $I$  是待匹配的边缘图像 (像素点的集合)
- $d_I(t)$  是模板中的点  $t$  到  $I$  中边缘点的最小的距离

- 计算复杂度:  $O(MNP) \rightarrow O(NP)$

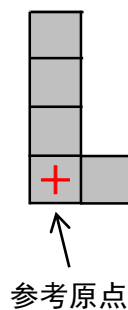
边缘图



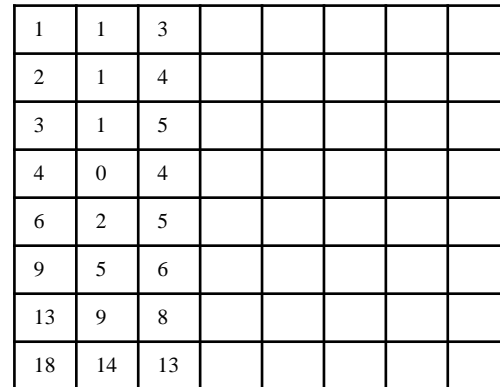
距离变换结果



模板



倒角变换距离



(后面五列结果自行计算)

# 倒角距离变换 (Chamfer Distance Transform)

## □ 距离变换实例

- 在距离变换的结果中，每个位置的值表示这个位置到最近的边缘点（或者其他二值化的图片结构）的距离

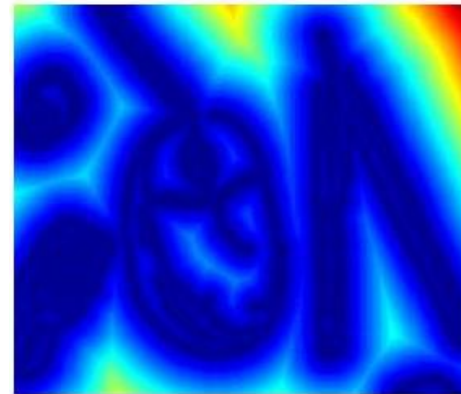
$$DF_I(p) = \min_{x \in E} \text{dist}(p, x)$$



原图



边缘检测结果



距离变换结果



# 倒角距离变换：如何做距离变换？

## □ 1-D距离变换

■ 1-D  $L_1$  范数的距离变换是一个计算复杂度为 $O(n)$ 的算法

■ 算法步骤

1. 对图中任意位置的值进行初始化，当 $j$ 在特征 $P$ 中时初始化为0，否则初始化为 $\text{inf}$ 。将第 $j$ 个位置的值记为 $D[j]$

2. 前向过程: for  $j$  from 1 up to  $n-1$ ，更新 $D[j]$

$$D[j] = \min(D[j], D[j-1] + 1)$$

3. 后向过程: for  $j$  from  $n-2$  down to 0，更新 $D[j]$

$$D[j] = \min(D[j], D[j+1] + 1)$$

# 倒角距离变换：如何做距离变换？

## □ 2-D距离变换：算法步骤与1-D情况类似

### ■ 初始化距离矩阵

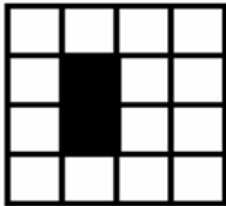
### ■ 前向过程：从上方和左方找离特征点最近的距离

$$D[i, j] = \min(D[i, j], D[i, j - 1] + 1, D[i - 1, j] + 1)$$

### ■ 后向过程：从下方和右方找离特征点最近的距离

$$D[i, j] = \min(D[i, j], D[i, j + 1] + 1, D[i + 1, j] + 1)$$

-	1
1	0
0	1
1	-



∞	∞	∞	∞
∞	0	∞	∞
∞	0	∞	∞
∞	∞	∞	∞

∞	∞	∞	∞
∞	0	1	∞
∞	0	∞	∞
∞	∞	∞	∞

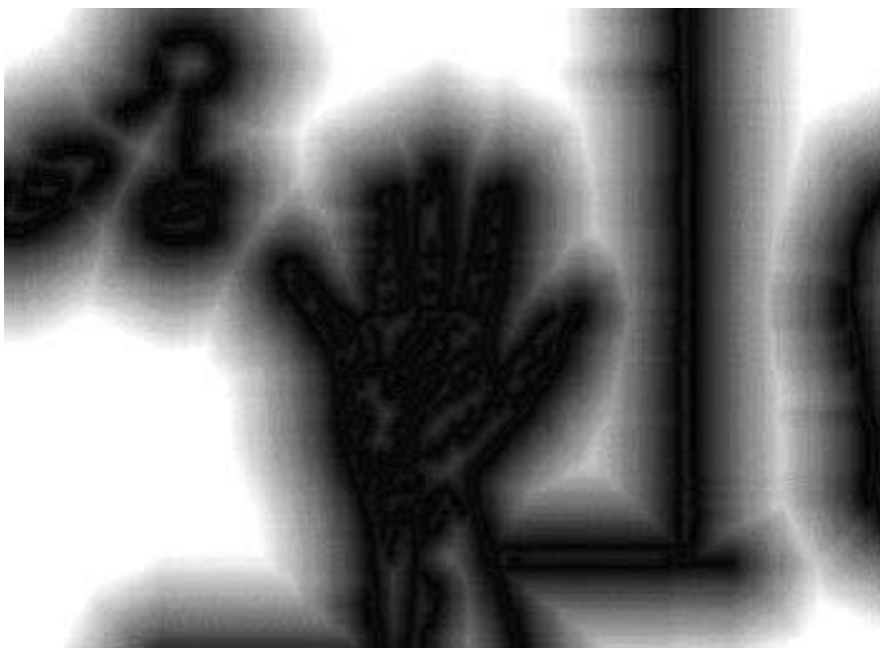
∞	∞	∞	∞
∞	0	1	2
∞	0	1	2
∞	1	2	3

2	1	2	3
1	0	1	2
1	0	1	2
2	1	2	3

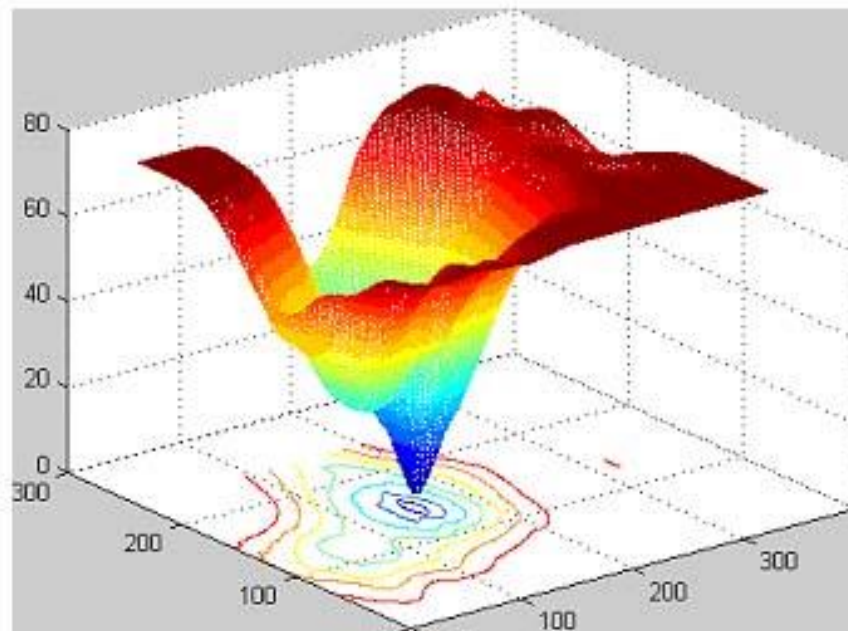
# 倒角距离变换：倒角匹配实例

## □ 手型匹配

边缘图的距离变换图



Chamfer Matching响应图





# 倒角距离变换 (Chamfer Distance Transform)

## □ 优点

- 对混乱背景干扰较为鲁棒
- 计算效率高

## □ 缺点

- 对缩放和旋转变换敏感
- 对形状的小的形变敏感
- 需要大量的模板形状，应应对目标形状的形变

## □ 改进方法

- 多尺度匹配
- 层级模型组织 (hierarchical model organization)





# 目标检测

---

- 简单形状的检测
- 人脸检测与识别
  - Viola & Jones Face Detector
- 一般目标检测



# Viola & Jones Detector

## □ 人脸检测算法面临的挑战

- 基本思想：将人脸检测问题形式化为滑窗分类问题
- 人脸可能出现在图中的任意区域，对于一张仅仅100万像素的图片，就需要分类超过100万个滑动窗口（sliding windows）
- 在每张图片中，相比较于所有可能的位置，仅仅有非常少数数量的人脸存在
- 在实际应用中我们希望虚警的检测能尽可能的少
- 为了检测速度尽可能的快，我们需要能快速的过滤掉大量不存在人脸的区域



# Viola & Jones Detector

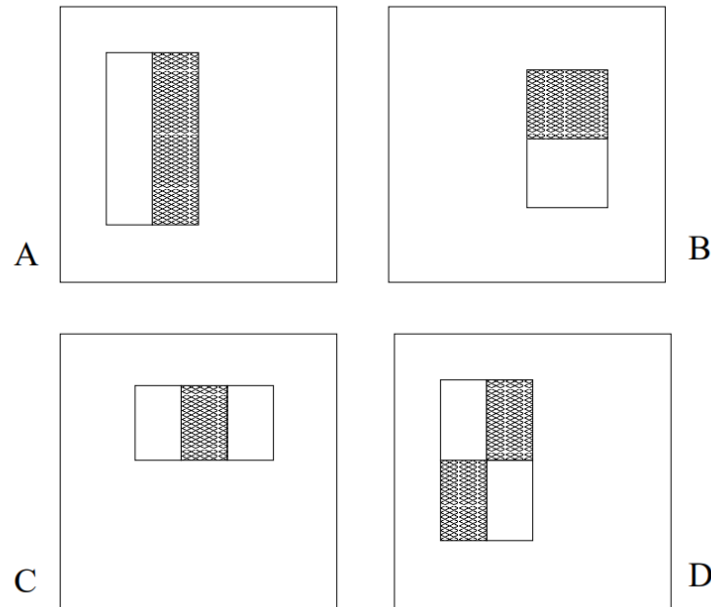
## □ Viola & Jones Detector: 第一个实时的人脸检测算法

- **弱可学习性等价于强可学习性**: 考虑一系列“基学习器”，让“后来者”重点关注“先行者”容易出错的部分，然后再将这些基学习器结合起来
- **速度提升**
  - ✓ 利用积分图像提取图像特征值，效率高
  - ✓ 利用adaboost分类器的特征筛选特性，保留最有用特征，减少了检测时的运算复杂度
- **准确率提升**
  - ✓ 将adaboost分类器进行改造，变成级联adaboost分类器，提高了人脸检测的准确率（降低漏检率和误检率）
- **检测精度评估**
  - ✓ 检测率：存在人脸并且被检测出的图像，在所有存在人脸图像中的比例。
  - ✓ 漏检率：存在人脸但是没有检测出的图像，在所有存在人脸图像中的比例。
  - ✓ 误检率：不存在人脸但是检测出存在人脸的图像，在所有不存在人脸图像中的比例。

# Viola & Jones Detector

## □ Haar-like features

- 与Haar小波类似的矩形状的特征，共四种基本形式



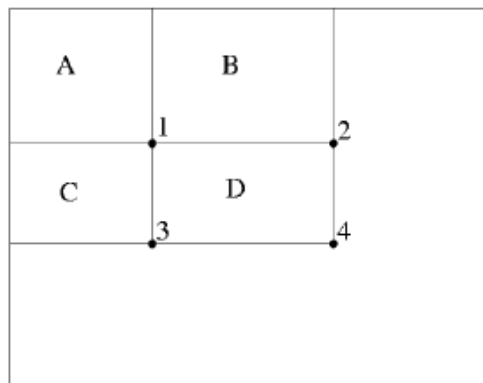
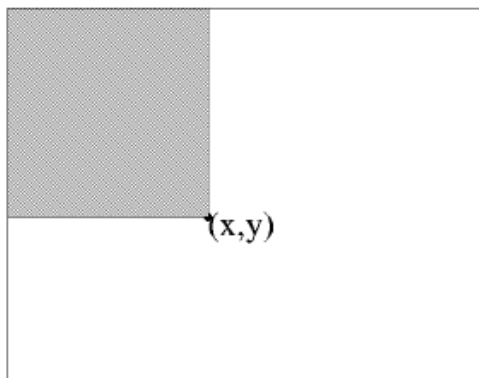
- 特征的计算:

✓  $\text{value} = \sum(\text{白色区域的像素值}) - \sum(\text{黑色区域的像素值})$

# Viola & Jones Detector

## □ 积分图 (Integral Images)

- 积分图的计算:  $I(x, y) = \sum_{i=0}^x \sum_{j=0}^y f(i, j)$



- 以右图中D区域为例

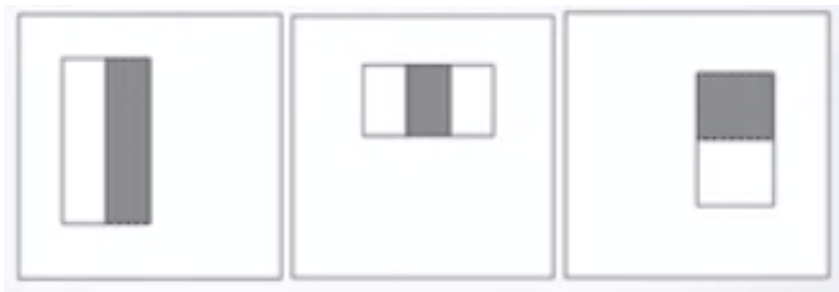
$$\sum_{(i,j) \in D} f(i, j) = I_1 + I_4 - (I_2 + I_3)$$

- 积分图的使用避免了重复的像素值求和计算, 每个像素点只需计算一次求得各点的积分图即可求得不同区域的Haar-like特征

# Viola & Jones Detector

## □ 面临的问题

- 当使用一个 $24 \times 24$ 大小的sliding window时，可能得到的特征有大约160,000个。



- 在测试时，我们不可能直接考虑整个特征集合进行检测。

## □ 通过AdaBoost选择有区分性的特征，并将对应的分类器级联组合起来

- 让每一级的分类器，都具有非常高的检测率（99.9%,接近100%），同时误检率也保持相当高（大概50%，即FPR）
- 例如，级联20个adaboost分类器，可实现检测率 $(0.999)^{20} = 0.98$ ，同时误检率为 $(0.5)^{20} = 9.5 \cdot 10^{-7}$



# Viola & Jones Detector

## □ 基于AdaBoost的特征选择和分类器训练

- 基于Haar-like特征，定义一系列弱分类器
- 对于Adaboost算法的每一个循环：
  1. 初始化正负样本的权重
  2. 在每一个样本上评价每一个特征
  3. 为每个特征选取最好的阈值（分类器）
  4. 选取最好的特征和分类器进行组合
  5. 对样本的权重进行重新分配
- 计算复杂度为 $O(TNK)$ 
  - ✓ 其中T为循环次数，N为样本个数，K为特征个数

# Viola & Jones Detector

## □ 详细步骤

- Given example images  $(x_1, y_1), \dots, (x_n, y_n)$  where  $y_i = 0, 1$  for negative and positive examples respectively.
- Initialize weights  $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$  for  $y_i = 0, 1$  respectively, where  $m$  and  $l$  are the number of negatives and positives respectively.
- For  $t = 1, \dots, T$ :

1. Normalize the weights,

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$$

so that  $w_t$  is a probability distribution.

2. For each feature,  $j$ , train a classifier  $h_j$  which is restricted to using a single feature. The error is evaluated with respect to  $w_t$ ,  $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$ .
3. Choose the classifier,  $h_t$ , with the lowest error  $\epsilon_t$ .
4. Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

where  $e_i = 0$  if example  $x_i$  is classified correctly,  $e_i = 1$  otherwise, and  $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$ .

- The final strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

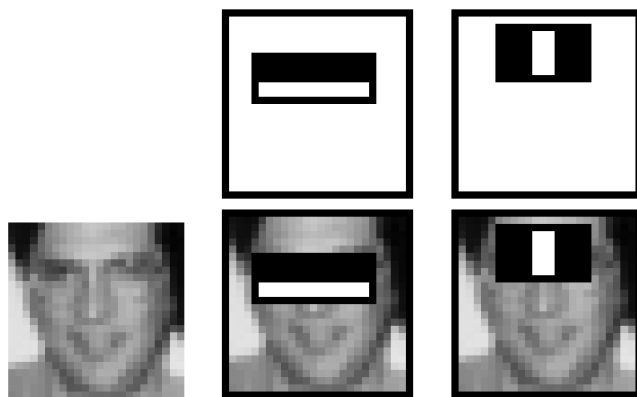
where  $\alpha_t = \log \frac{1}{\beta_t}$



# Viola & Jones Detector

## □ 特征选择的结果

- 可视化结果：下图是第一个和第二个被选中的特征

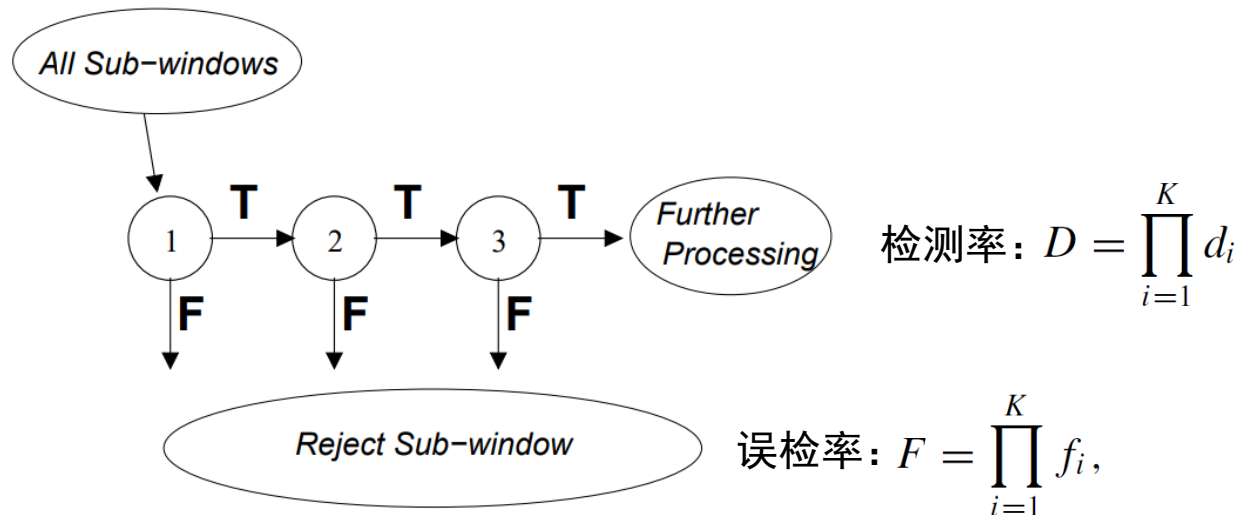


- 定量的结果：由200个特征构成的正面人脸检测器检出率为95%，而14084个样本中出现1次的虚警。

# Viola & Jones Detector

## □ Attentional cascade

- We start with simple classifiers which reject many of the negative sub-windows while detecting almost all positive sub-windows
- Positive response from the first classifier triggers the evaluation of a second (more complex) classifier, and so on
- A negative outcome at any point leads to the immediate rejection of the sub-window





# Viola & Jones Detector

---

## □ 效果

- 在测试时，平均每个window使用了10个特征
- 在一个700Mhz的奔腾3处理器上，这个算法检测一张 $384 \times 288$ 的照片中的人脸需要大概0.067秒（15fps），比之前性能相当的方法快了15倍

## □ 局限性：适用于正脸的检测，对侧脸检测性能不佳



# 目标检测

---

- 简单形状的检测
- 人脸识别与检测
- 一般目标检测
  - 问题定义与评价标准
  - 滑窗+HOG特征
  - 可形变部件模型



# 目标检测

---

## □ 问题定义

- 在图像中检测和定位来自不同类别的一般常见物体，如车辆，行人等

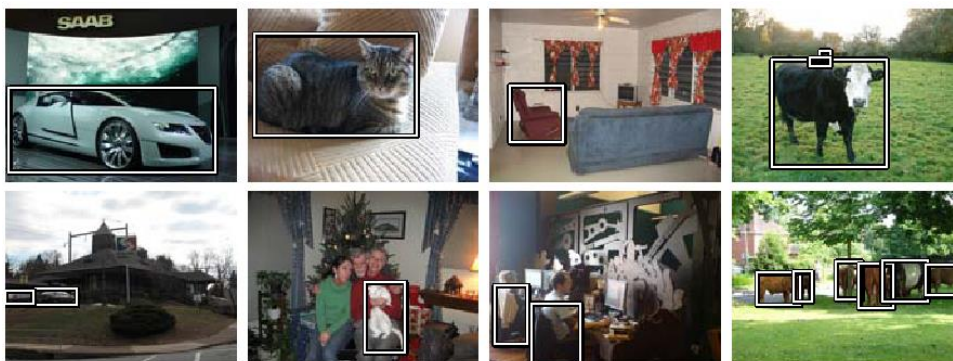
## □ 目标检测中存在的挑战

- 光照变化
- 视角变化
- 物体的可变形性
- 物体的类内差异
- 复杂背景干扰

# 目标检测

## □ 目标检测基准数据集

- PASCAL VOC Challenge: 20个类别



- ImageNet Large Scale Visual Recognition Challenge (ILSVRC): 检测目标包含200个类别
- Common Objects in Context (COCO): 80个类别



# 目标检测的评价标准

## □ 单个目标框正确与否的评判



— predictions  
— ground truth

### True positive:

- 分类正确，且与ground truth的IOU大于一定的阈值
- IOU: intersection over union

### False Negative (漏检):

- 有ground truth存在，但是没有对应的预测结果

### False positive (虚警):

- 分类错误，或者预测结果与ground truth的IOU小于一定的阈值

# 目标检测的评价标准

## □ Precision与Recall的定义

	<u>Predicted 1</u>	<u>Predicted 0</u>
<u>True 1</u>	true positive	false negative
<u>True 0</u>	false positive	true negative

	<u>Predicted 1</u>	<u>Predicted 0</u>
<u>True 1</u>	TP	FN
<u>True 0</u>	FP	TN

	<u>Predicted 1</u>	<u>Predicted 0</u>
<u>True 1</u>	hits	misses
<u>True 0</u>	false alarms	correct rejections

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$



# 目标检测的评价标准

- 在实际检测中，我们会得到大量的预测结果，这些目标框的置信度分数在0~1之间
  - 保留大量目标框会得到极高的recall，而precision非常低
  - 只保留预测分数很高的目标框会得到极高的precision，而recall非常低



— predictions  
— ground truth

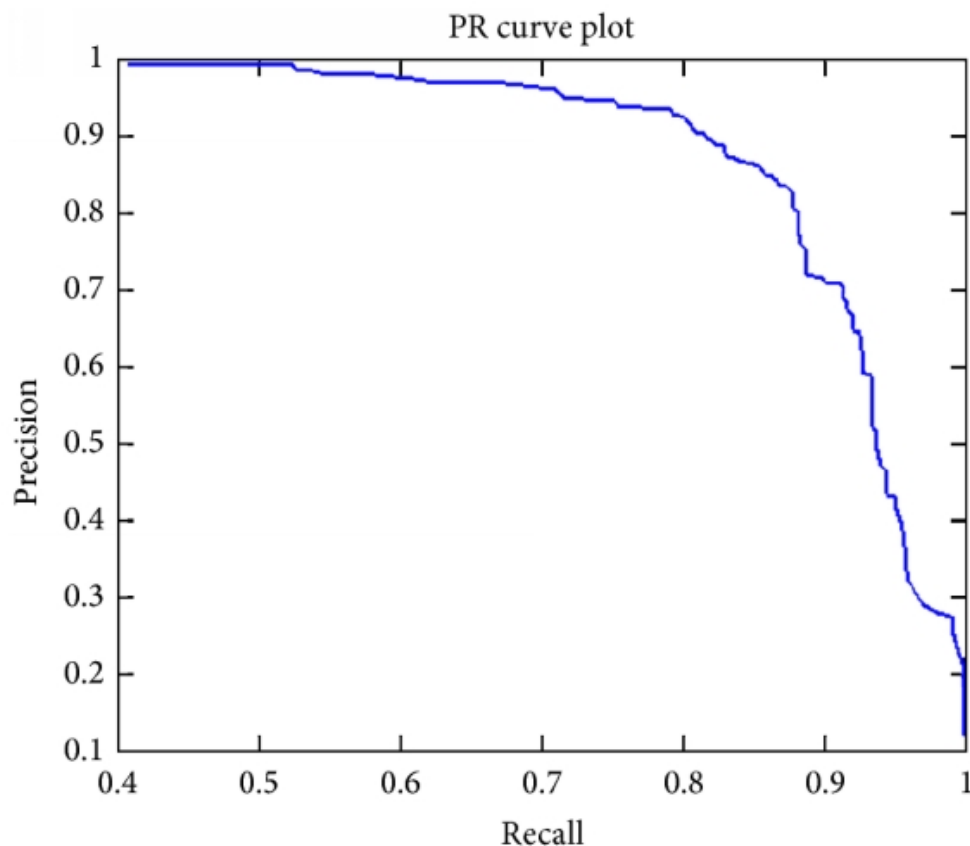
单独依靠precision或者recall的不足以评判检测器的优劣

我们使用**mean Average Precision (mAP)**作为评价标准

# 目标检测的评价标准

## □ 平均精度 (Average Precision)

### ■ Precision-recall 曲线

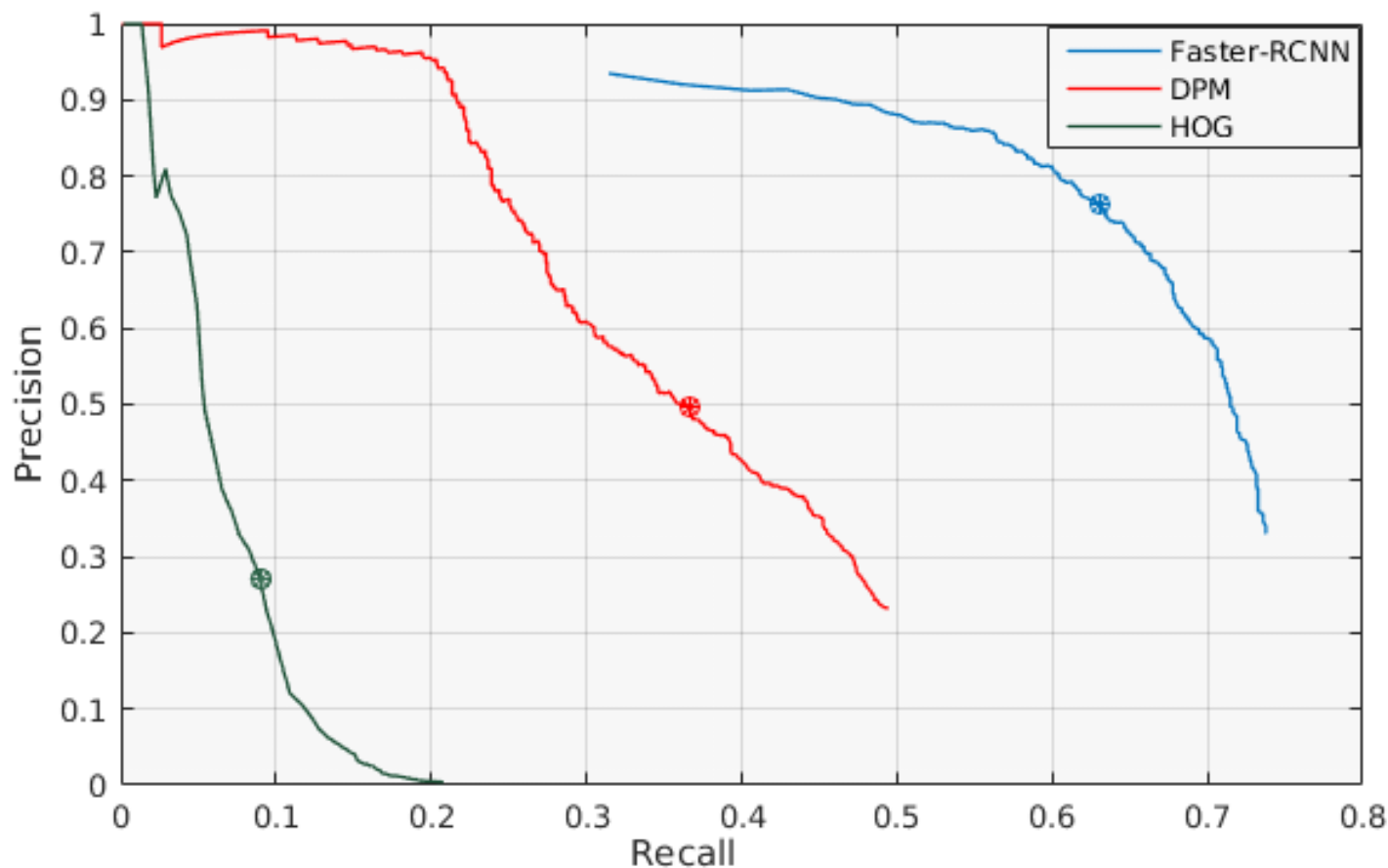


Precision-recall 曲线与坐标轴围成的面积，即为 **average precision (AP)**

多个类别的average precision 的平均值，即为 **mean average precision (mAP)**

# 目标检测的评价标准

- 从precision-recall曲线以及average precision, 能够很直观的比较检测方法的性能





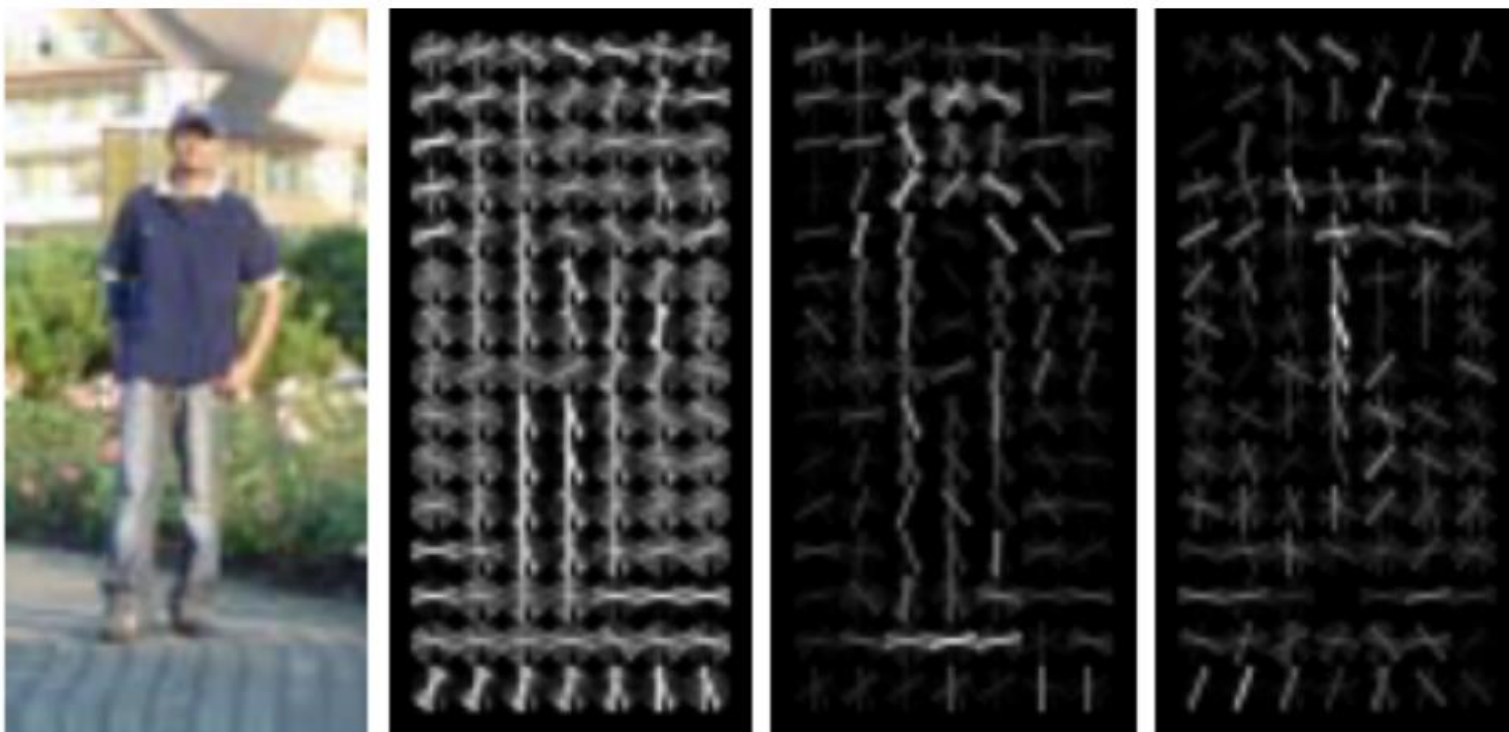
# 目标检测

---

- 简单形状的检测
- 人脸识别与检测
- 目标检测
  - 问题定义与评价标准
  - 滑窗+HOG特征
  - 可形变部件模型

# 基于HOG 特征的模板表达

- HOG: 梯度方向直方图
  - Histogram of Oriented Gradient
- 对待检测物体提取HOG template, 并且将之作为filter



# 滑窗 + HOG特征

- 用滑动窗口在整张图上搜索，判断图中是否有与HOG模板相匹配的位置



# 滑窗 + HOG特征

## □ 滑窗方法存在的问题

- 如何检测图中与人的尺度存在明显差异的公交车？

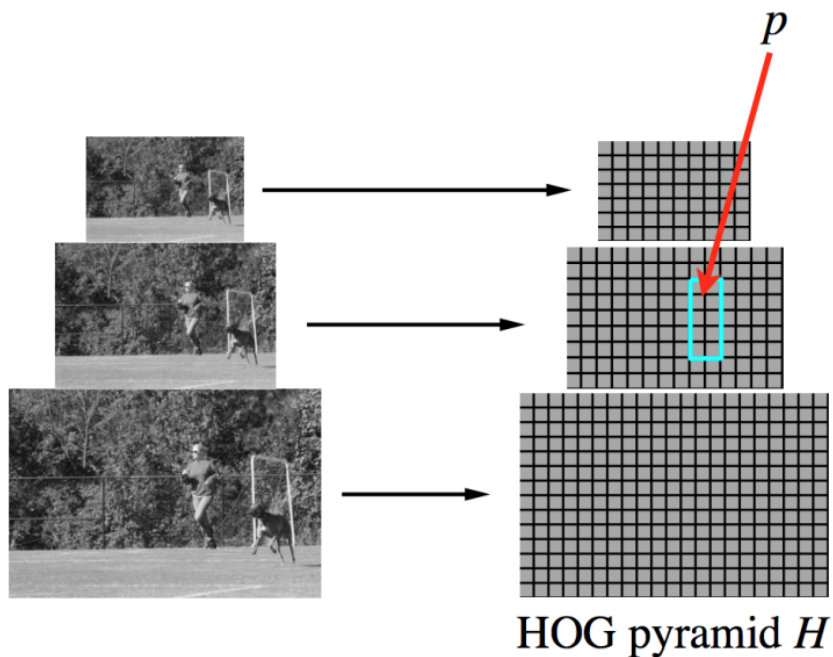


目标检测成功与否，与滑窗的大小选择息息相关



# 滑窗 + HOG特征

## □ 多尺度滑窗



Filter  $F$



Score of  $F$  at position  $p$  is  
 $F \cdot \phi(p, H)$

$\phi(p, H)$  = concatenation of  
HOG features from  
subwindow specified by  $p$





# 目标检测

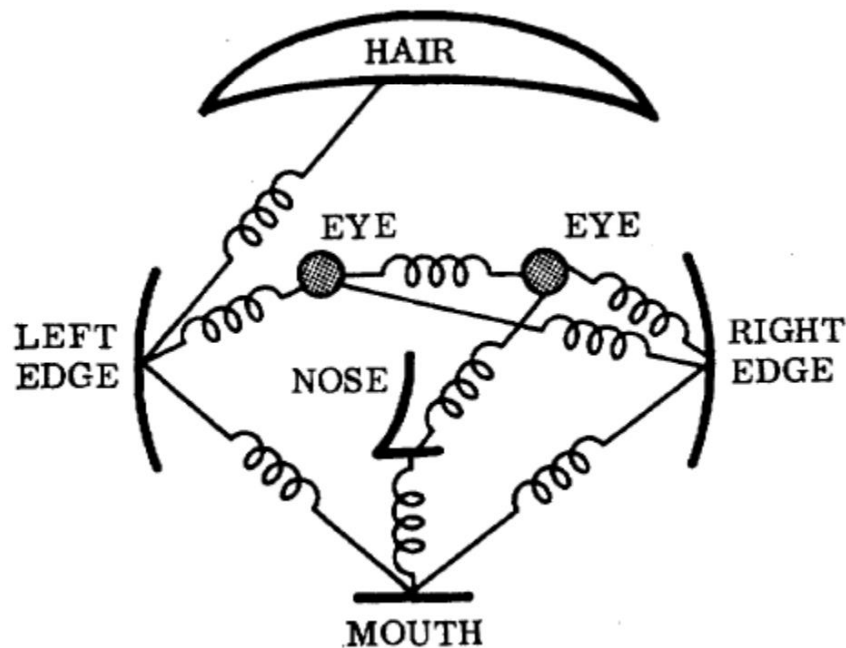
---

- 简单形状的检测
- 人脸识别与检测
- 目标检测
  - 问题定义与评价标准
  - 滑窗+HOG特征
  - 可形变部件模型

# 可变部件模型 (Deformable Parts Model)

## □ 弹簧形变模型

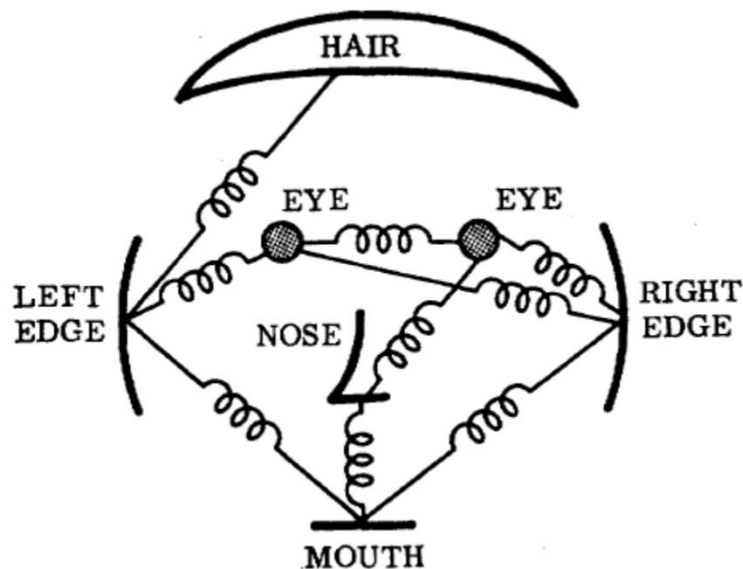
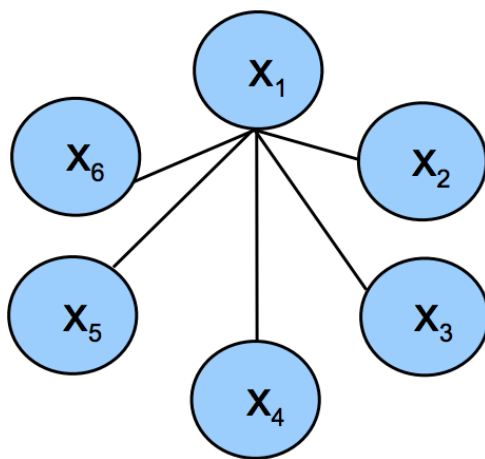
- 将物体表达为一系列可变形的部件的集合
- 每个部件表示物体的一个局部特征
- 部件与部件之间的呈弹簧状连接（可变形）



Fischler and Elschlager,  
Pictorial Structures,  
1973

# 可变部件模型 (Deformable Parts Model)

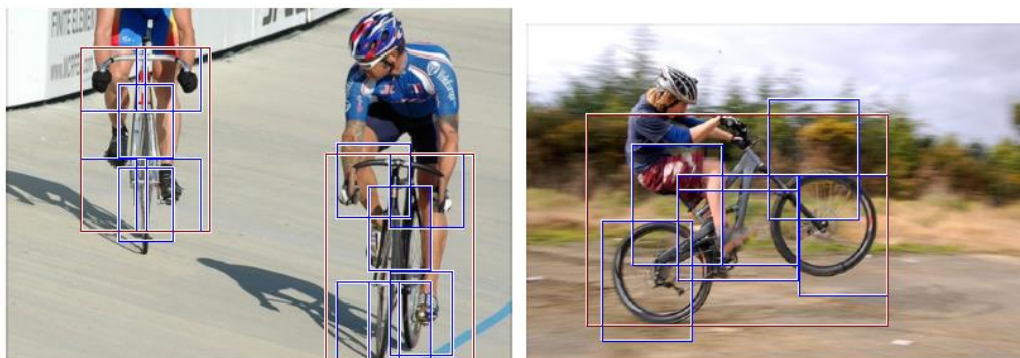
- 物体的DPM可以表示成一个star model, 由一个**整体模板**和一系列**局部模板**组成



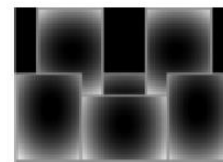
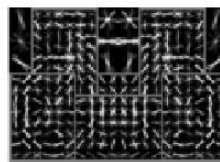
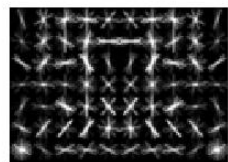
人脸模板可以表示为**整个人脸检测器**与**各个部件**（头发，左边耳朵，右边耳朵等等）的star model。人脸检测器作为**root**，每个部件都与root建立联系

# 可变部件模型 (Deformable Parts Model)

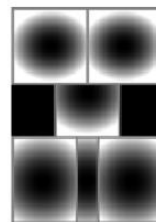
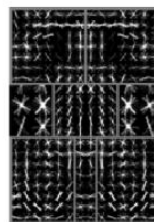
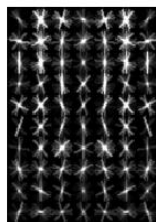
## □ 示例：Two-component bicycle model



“side”  
component



“frontal”  
component



当part filter偏离  
既定位置，给  
出对应的惩罚

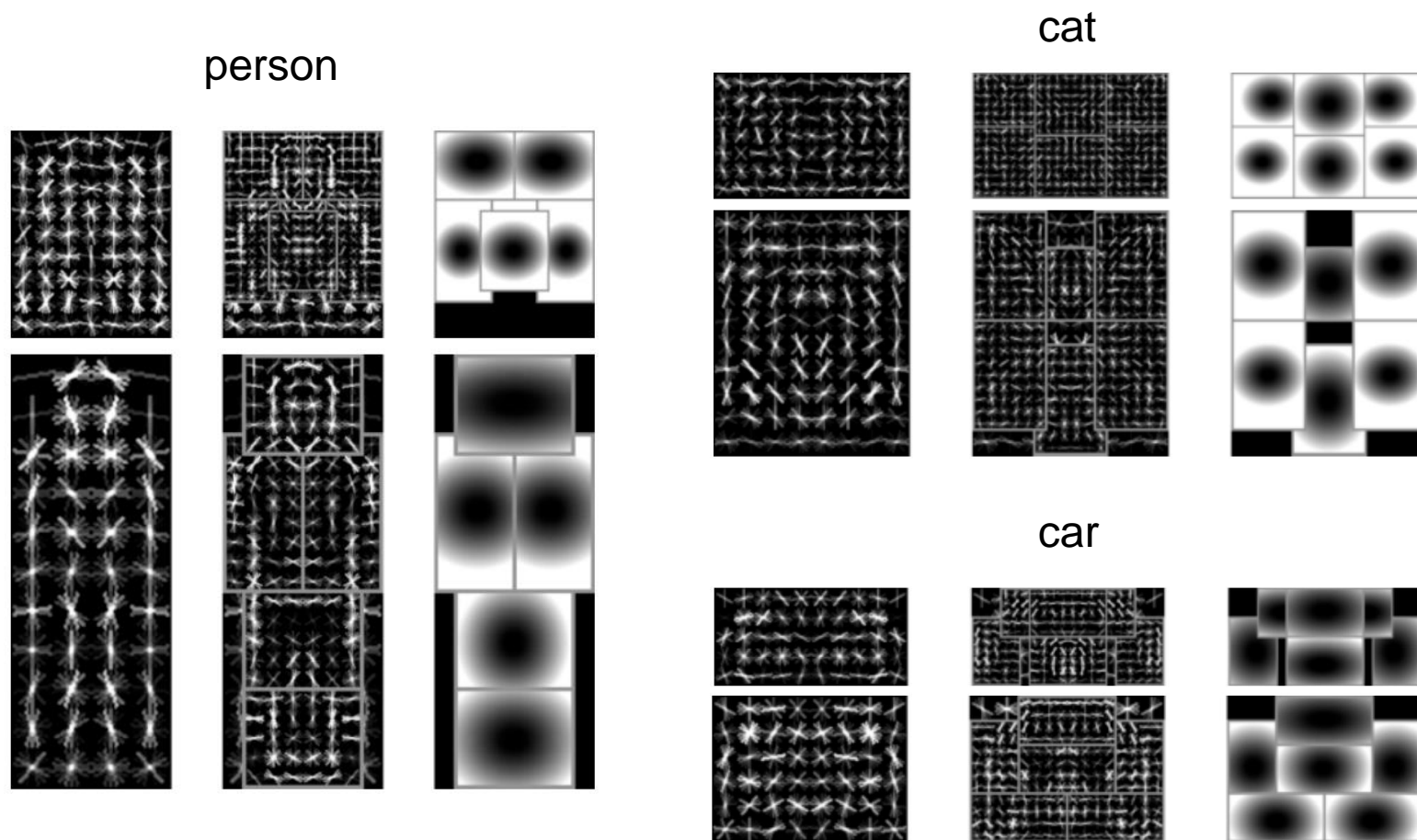
root filter

part filter

deformable cost

# 可变部件模型 (Deformable Parts Model)

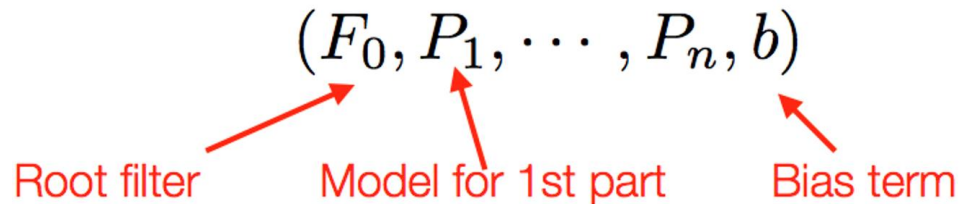
## □ 示例





# 可变部件模型 (Deformable Parts Model)

- 若把物体分为可变形的 $n$ 个部位，则他的DPM是一个 $(n + 2)$  元组



- 每个局部模型可定义为  $(F_i, v_i, d_i)$ 
  - $F_i$  为第 $i$ 个部位的filter
  - $v_i$  为第 $i$ 个部位相对于root filter的锚点位置
  - $d_i$  为第 $i$ 个部位偏移到各个位置的deformable cost

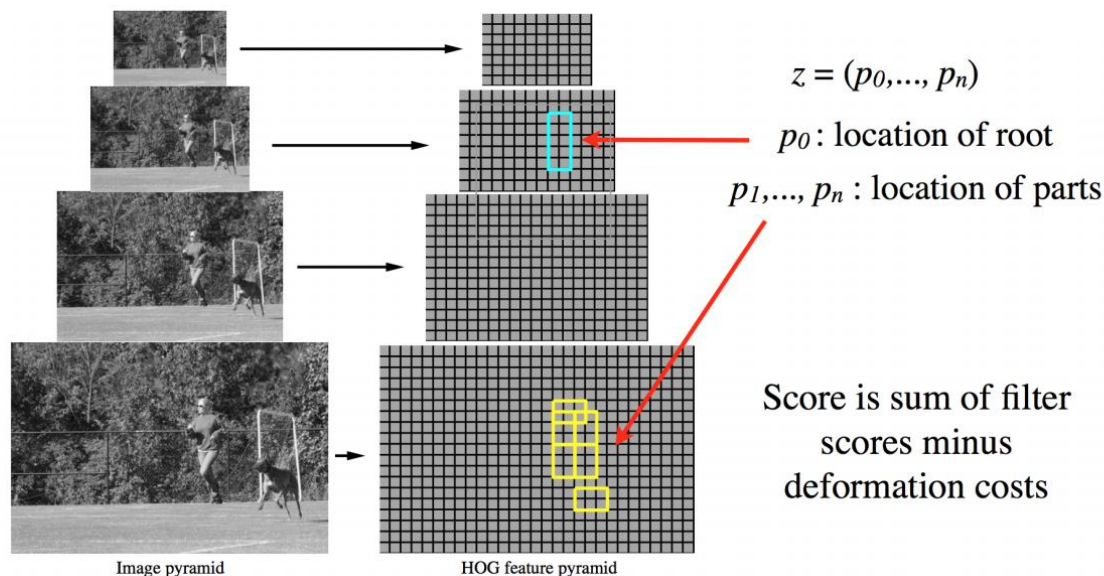


# 可变部件模型 (Deformable Parts Model)

- 当使用DPM时，检测分数定义为：检测器得到的分数减去各个部位的deformable cost

$$S = \prod_{i=0}^n F_i \phi(p_i, H) - \sum_{i=1}^n d_i (dxi, dyi, dxi^2, dyi^2)$$

$p_i = (x_i, y_i, l_i)$  specifies the level and position of the  $i$ -th filter

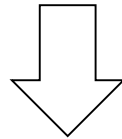




# 可变部件模型 (Deformable Parts Model)

## □ 基于sliding window的检测流程

首先使用sliding window和global feature大致定位物体的位置



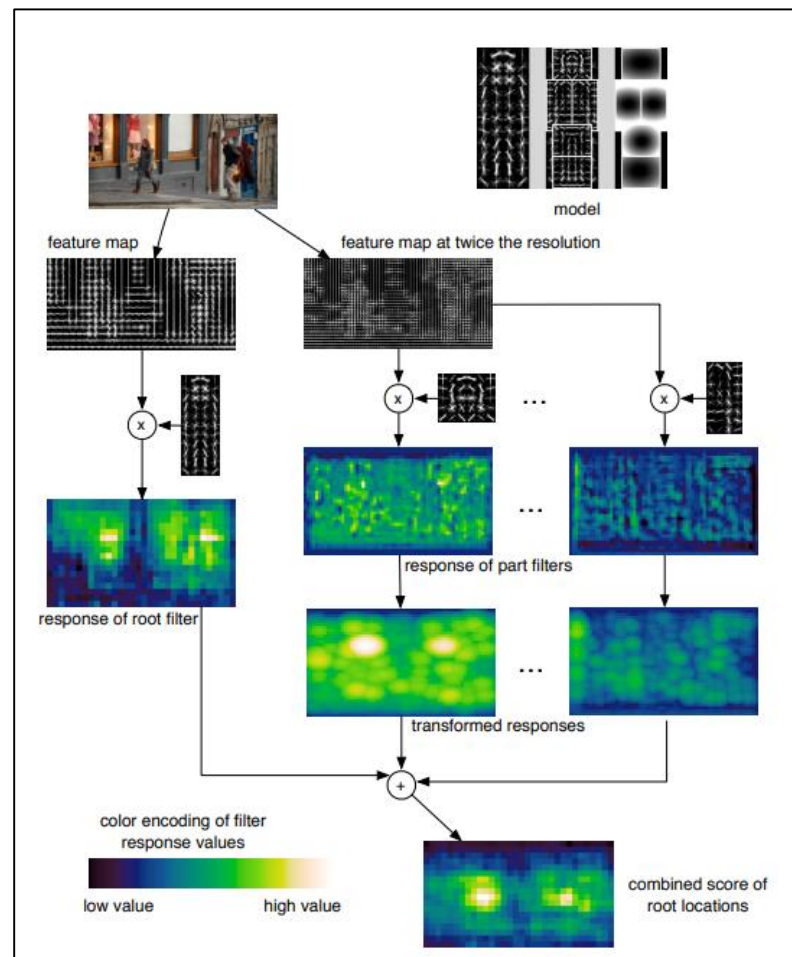
再使用part filters作用到物体上，与 global filter 的结果相结合，得到检测的分数



# 可变部件模型 (Deformable Parts Model)

## □ 算法步骤

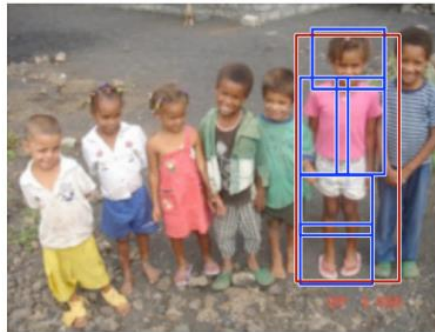
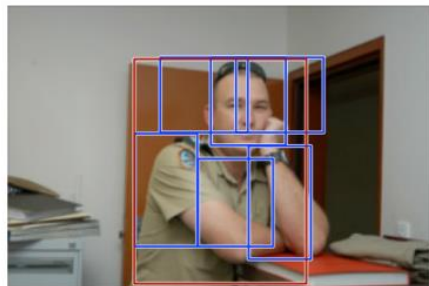
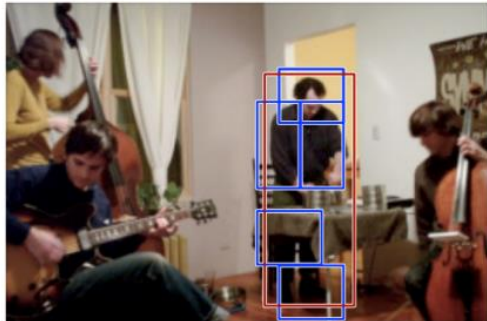
1. 产生多个基于HOG特征的模板，包括整体模板与一系列局部模板。
2. 用这些模板去对输入图像的HOG特征做卷积，得到响应图。
3. 将局部模板的卷积响应图加上**变形惩罚**，并且和整体模板的响应图相结合，得到融合的响应图



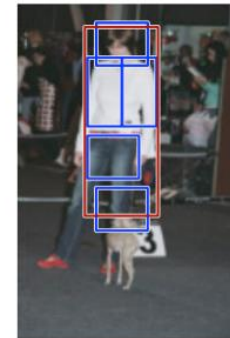
# DPM检测结果

## □ Person

high scoring true positives



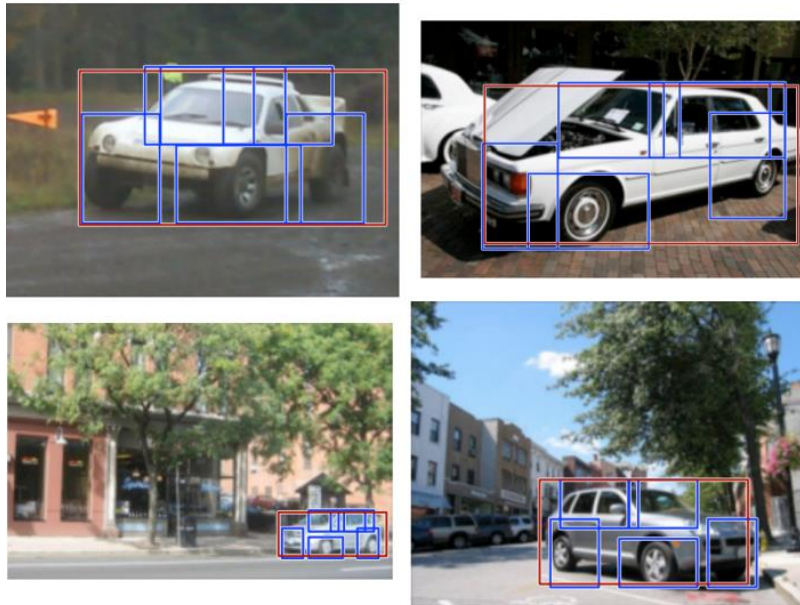
high scoring false positives  
(not enough overlap)



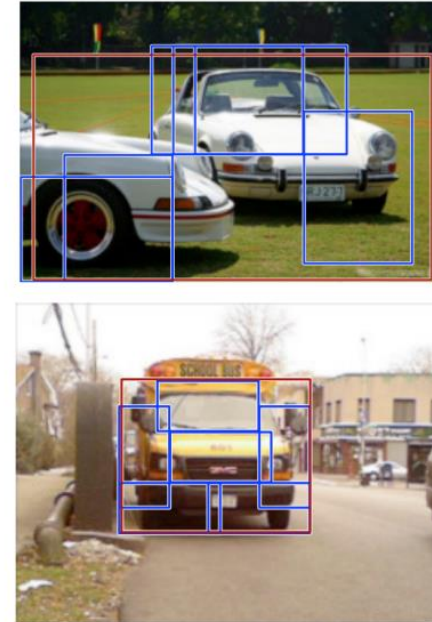
# DPM检测结果

## □ Car

high scoring true positives



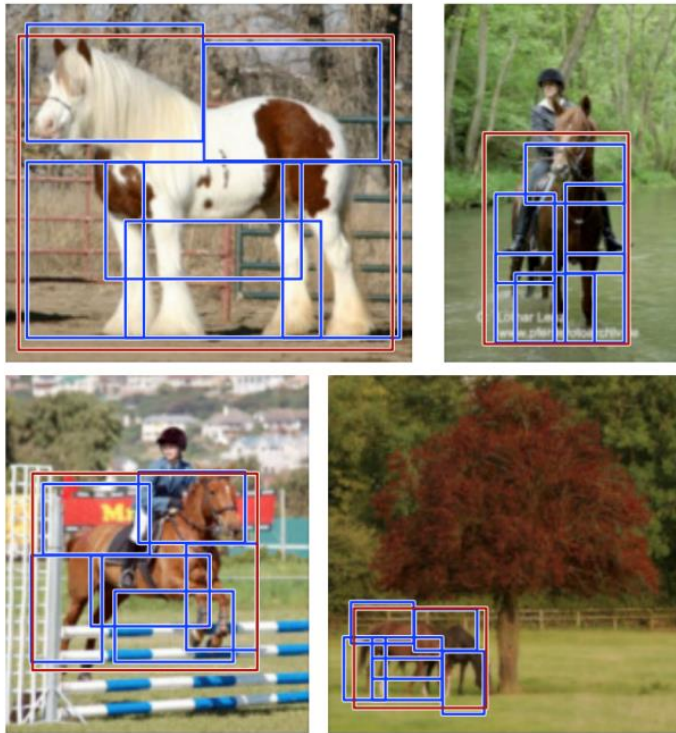
high scoring false positives



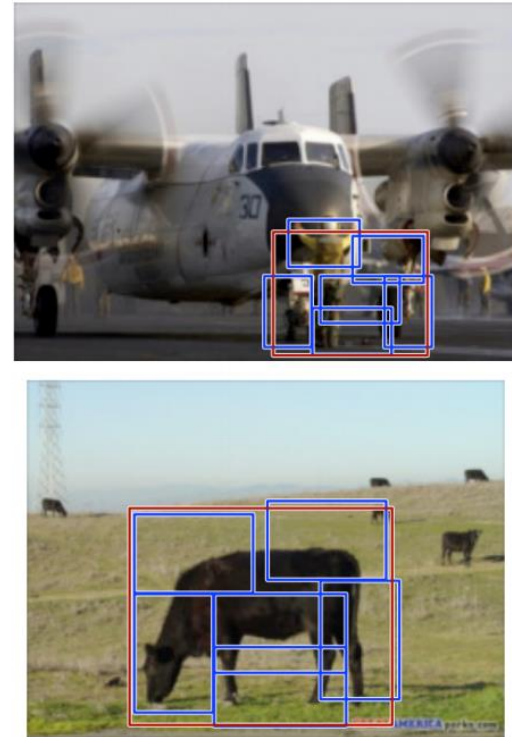
# DPM检测结果

## □ Horse

high scoring true positives



high scoring false positives







# DPM的优缺点

## □ 关键思想

- 手工选择部件集合：为每个部件(part)专门训练的检测子
- 在部件响应上，训练空间模型
- 评估部件响应的联合似然

## □ 优点

- 可解释性：部件具有直观的物理意义
- 对每个部件，可以使用表征的检测方法
- 适用于特定类别的物体检测

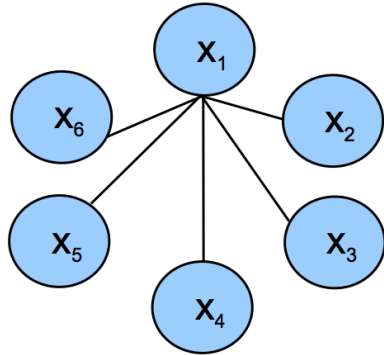
## □ 缺点

- 手工选择部件的泛化性不好
- 一些语义层面的部件可能无法用简单的表观分布 (appearance distribution) 表达
- 无法保证重要的部件不被遗漏

# DPM的拓展

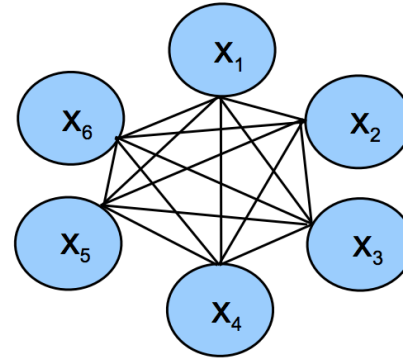
## □ 星状模型 (star model) 与星座模型 (constellation model)

“Star” shape model



- ▶ e.g. ISM (Implicit Shape Model)
- ▶ Parts mutually independent
- ▶ Recognition complexity:  $O(NP)$
- ▶ Method: Generalized Hough Transform

Fully connected shape model



- ▶ e.g. Constellation Model
- ▶ Parts fully connected
- ▶ Recognition complexity:  $O(N^P)$
- ▶ Method: Exhaustive search