第四章

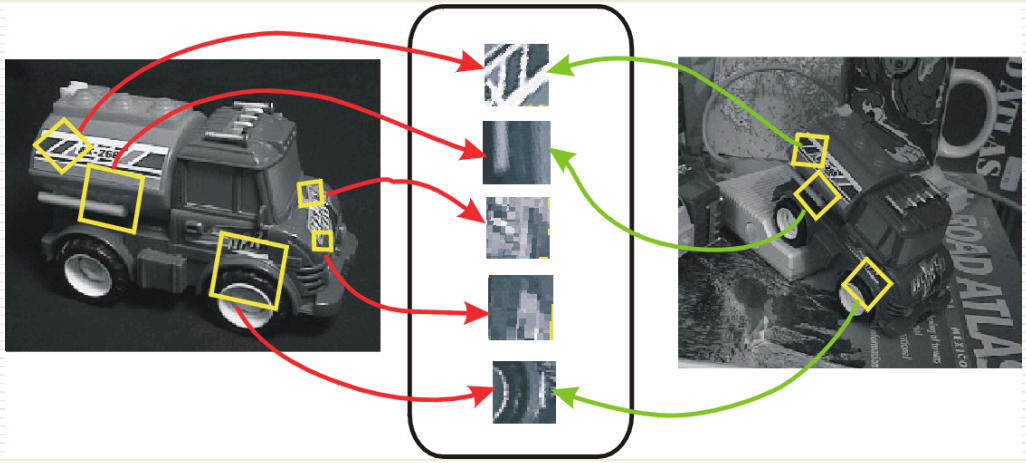
目标检测

李静

三、基于特征点匹配的目标检测

- 目标存在旋转或尺度变化——基于特征点的目标检测

基于特征点的匹配方法主要有SIFT(Scale Invariant Features Transform)、SURF(Speeded Up Robust Features)、ORB（Oriented Robust Brief）、BRISK等。



- 将一幅图像映射（变换）为一个局部特征向量集；
- 特征向量具有平移、缩放、旋转不变性，同时对光照变化、仿射及投影变换也有一定不变性。

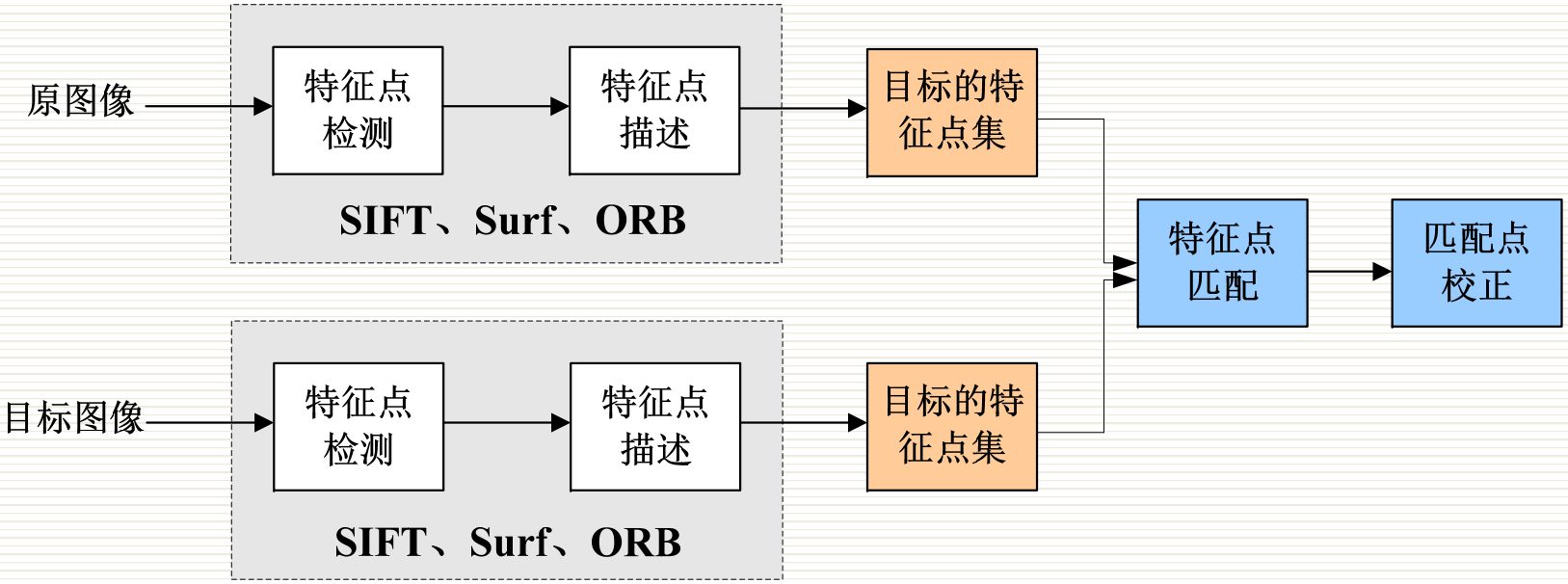
三、基于特征点匹配的目标检测

特征点匹配方法主要有三大步骤：

1、提取关键点；

2、对关键点附加描述信息（特征向量）；

3、通过两方特征点（附带上特征向量的关键点）的两两比较找出**相互匹配**的若干对特征点，也就建立了图像间的对应关系。



三、基于特征点匹配的目标检测

（1）SIFT 特征向量的生成算法

总共包括四步：

- 检测尺度空间极值点，初步确定关键点位置和所在尺度。
- 精确确定关键点的位置和尺度，同时去除低对比度的关键点和不稳定的边缘响应点，以增强匹配稳定性、提高抗噪声能力。
- 为每个关键点指定方向参数，使算子具备旋转不变性。
- 关键点描述子的生成，即生成SIFT特征向量。

三、基于特征点匹配的目标检测

1）构建尺度空间

$$G(x,y,\sigma)=ce^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

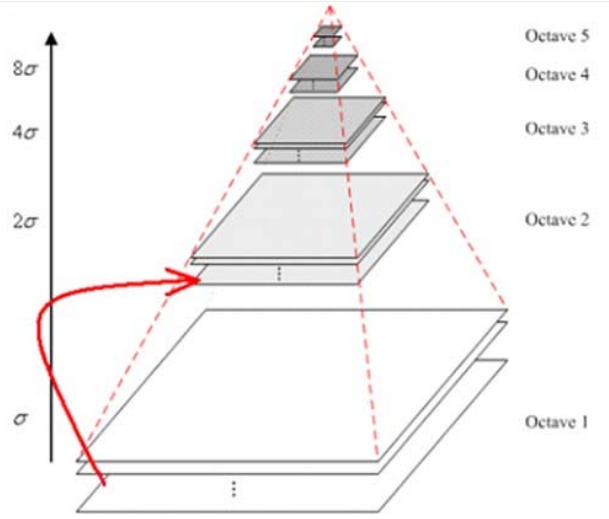
σ是标准差，其值越大，图像越模糊(平滑)。

根据σ的值和模板大小，计算出高斯模板矩阵的值，与原图像做卷积，即可获得原图像的平滑(高斯模糊)图像。

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

输入图像经过s次卷积后可以得到高斯卷积函数为G(x, y, 2σ)的输出图像，所有从 σ到2 σ的图像构成一个组，每个组固定有N个平面。每一层

I_s= G(x, y, k^s σ)*I， s=1, 2, …, N. 而k^N=2。



高斯金字塔

三、基于特征点匹配的目标检测

□ 高斯金字塔

在同一阶中相邻两层的尺度因子比例系数是k，则第1阶第2层的尺度因子是$k\sigma$，然后其它层以此类推；

第2阶的第1层由第一阶的中间层尺度图像进行降采样获得，其尺度因子是$k^2\sigma$，然后第2阶的第2层的尺度因子是第1层的k倍即$k^3\sigma$；

第3阶的第1层由第2阶的中间层尺度图像进行降采样获得。其它阶的构成以此类推。

高斯图像金字塔每层的尺度：

$$\sigma(s)=\sigma_0\cdot 2^{\frac{s}{N}}$$

s子层坐标、σ_0初始尺度、N每组层数（一般为3～5）。

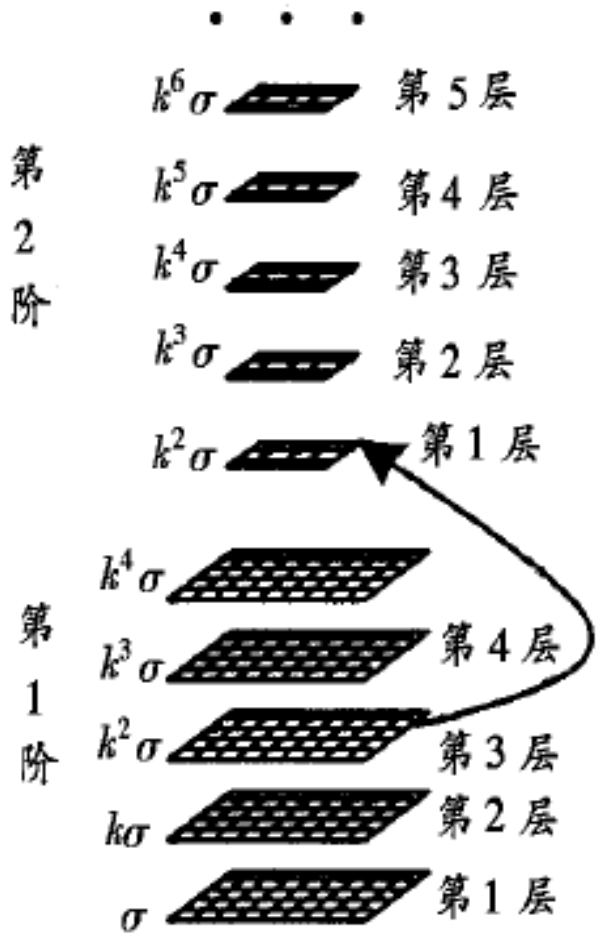


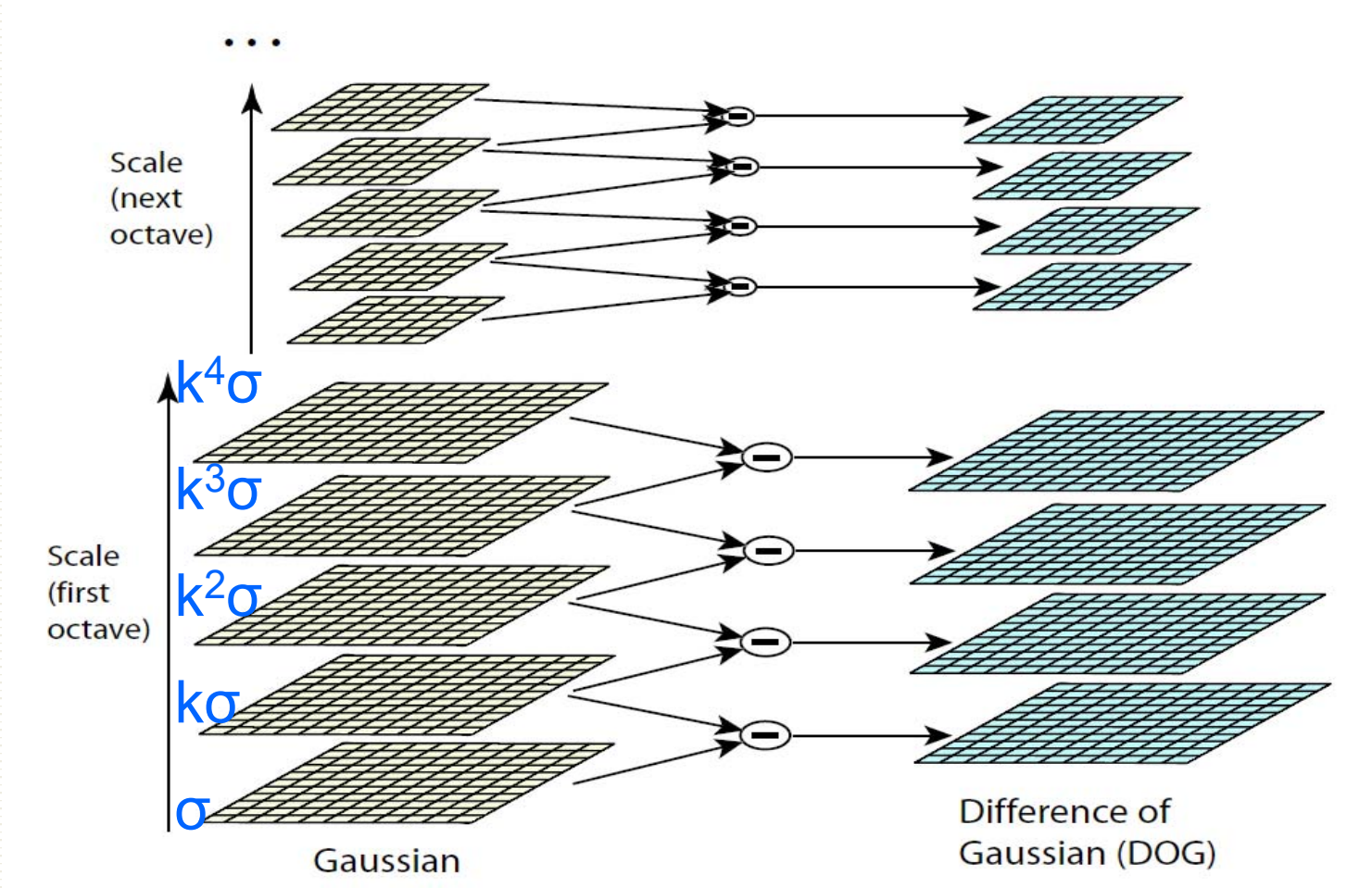
图1 高斯金字塔

Fig1 Gaussion pyramid

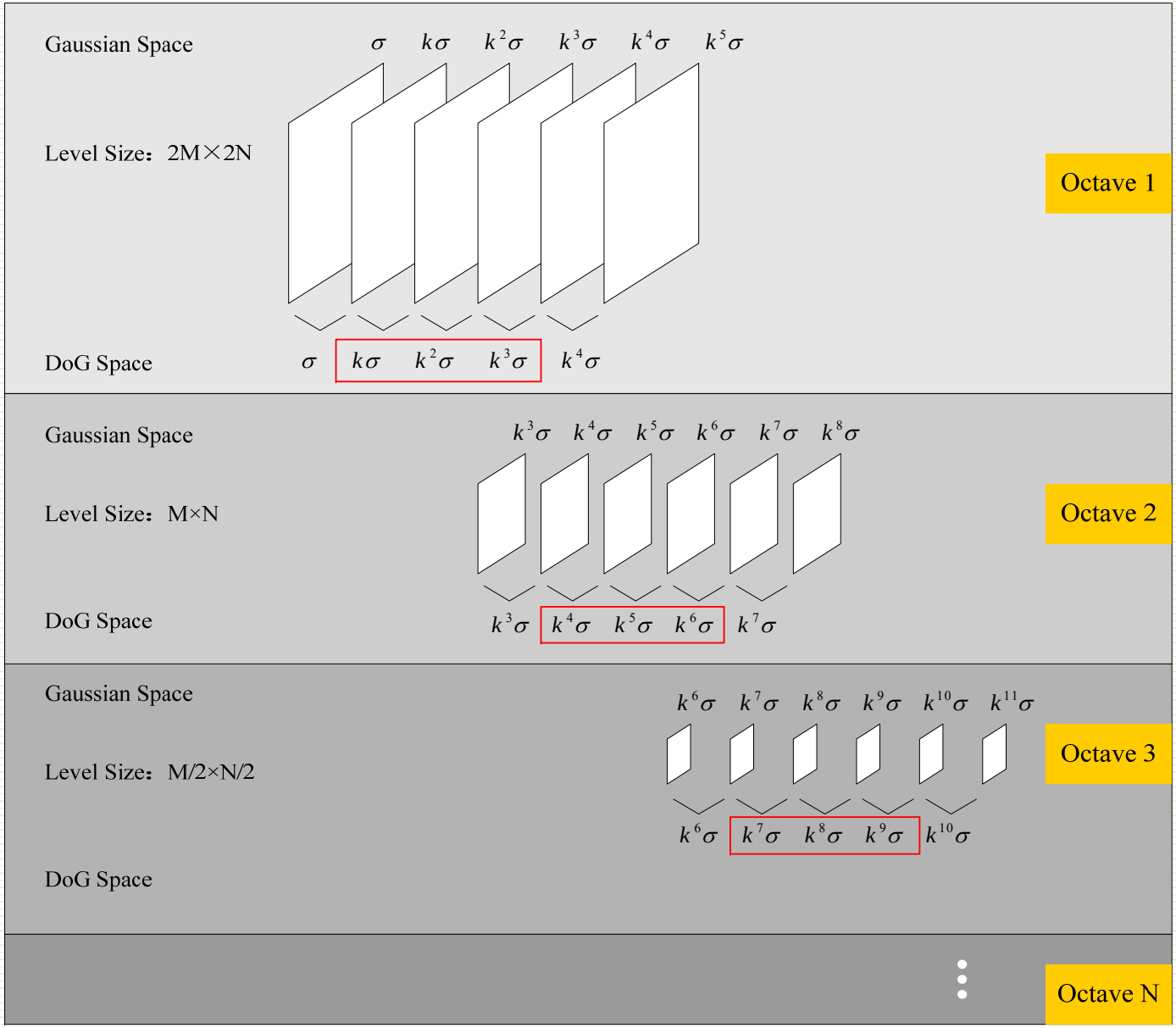
三、基于特征点匹配的目标检测

高斯差分金字塔　通过高斯金字塔中相邻尺度空间函数相减获得

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



三、基于特征点匹配的目标检测

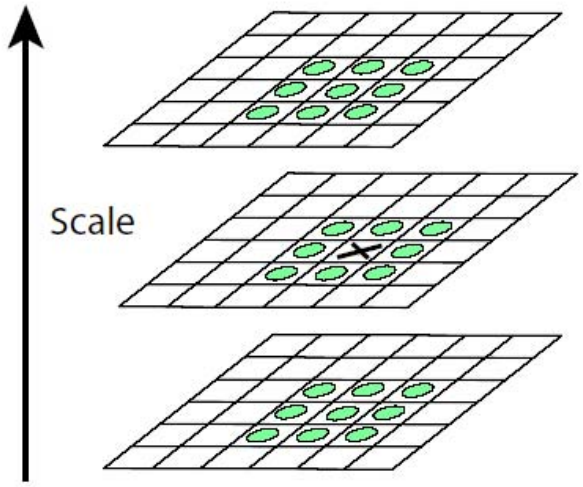


三、基于特征点匹配的目标检测

空间极值点检测

在DOG尺度空间金字塔中，为了检测到DOG空间的最大值和最小值， DOG尺度空间中中间层（最底层和最顶层除外）的每个像素点需要跟同一层的相邻8个像素点以及它上一层和下一层的9个相邻像素点总共26个相邻像素点进行比较，以确保在尺度空间和二维图像空间都检测到局部极值。

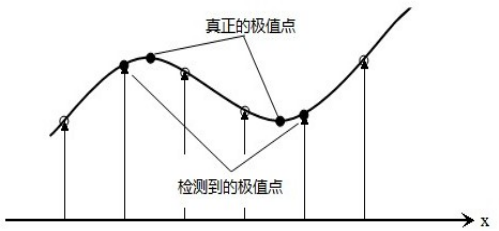
标记为叉号的像素若比相邻26个像素的DOG值**都大或都小**，则该点将作为一个**局部极值点**，记下它的位置和对应尺度。



三、基于特征点匹配的目标检测

2）关键点定位

检测到的极值点是离散空间的极值点，



- 通过拟合三维二次函数来精确确定关键点的位置和尺度；
- 由于DOG算子会产生较强的边缘响应还需去除低对比度的关键点和不稳定的边缘响应点，以增强匹配稳定性、提高抗噪声能力。

关键点尺度：$\sigma_{oct} = \sigma_0 \cdot 2^{\frac{s}{N}}$

式中：σ_{oct}为关键点的尺度，

s 为关键点在高斯差分金字塔中所处于的层数，

N 为每组的层数。

获得关键点的位置和尺度

三、基于特征点匹配的目标检测

3）关键点方向分配

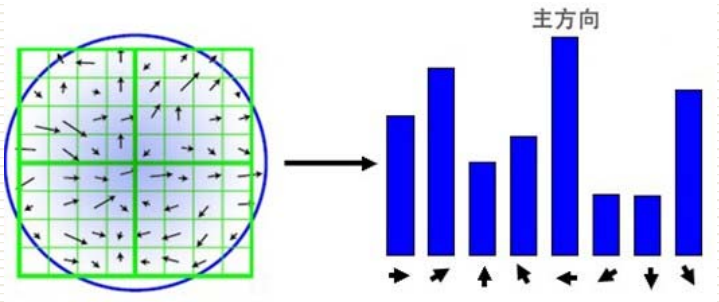
为了使描述符具有旋转不变性，需要利用图像的局部特征为给每一个关键点分配一个基准方向。对于在DOG金字塔中检测出的关键点，采集其所在高斯金字塔图像邻域窗口内像素的梯度和方向分布特征。梯度的幅值和方向如下：

$$gradI(x,y)=\left(\frac{\partial I}{\partial x},\frac{\partial I}{\partial y}\right)$$

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

$$\theta(x,y)=\arctan((I(x+1,y)-I(x-1,y))/(I(x,y+1)-I(x,y-1)))$$

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。梯度直方图将0～360度的方向范围分为8个柱(bins)，其中每柱45度。



关键点方向直方图

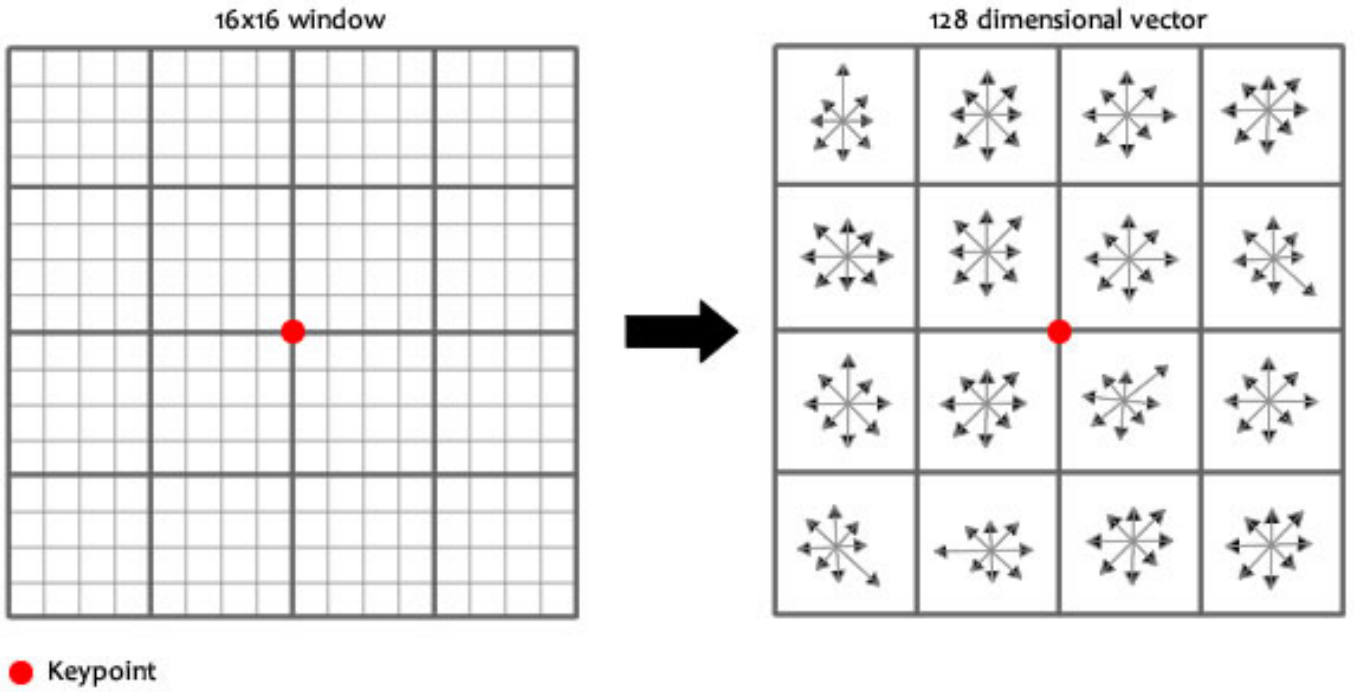
直方图的峰值方向代表了关键点的主方向

每个关键点有三个信息：位置、尺度、方向。

三、基于特征点匹配的目标检测

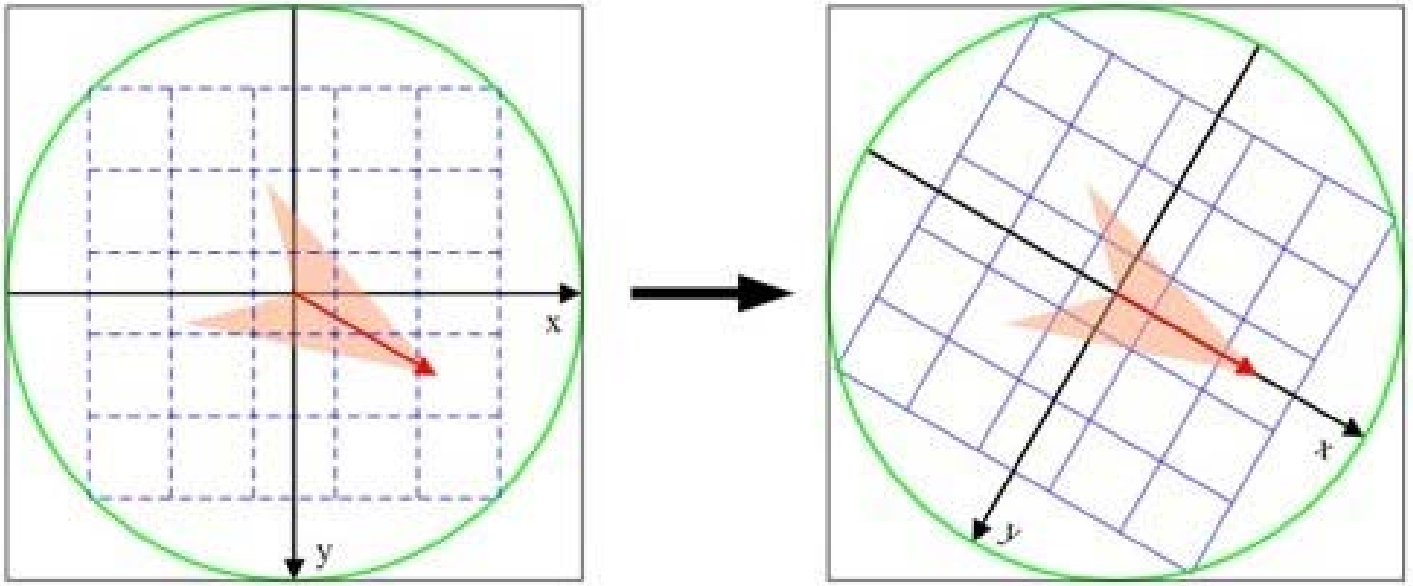
4）关键点特征描述

① 计算关键点周围的16*16邻域中每一个像素的梯度，描述子使用在关键点尺度空间内4*4的窗口（4*4个像素）中计算8个方向的梯度信息，共4*4*8=128维向量表征。



三、基于特征点匹配的目标检测

② 将坐标轴旋转为关键点的方向，以确保旋转不变性



旋转后邻域内采样点的新坐标为：

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (x, y \in [-radius, radius])$$

获得旋转后的关键点描述向量。

三、基于特征点匹配的目标检测

③ 描述子向量元素阈值化及规范化

描述子向量元素**规范化**：

$W=(w_1,w_2,\cdots,w_{128})$ 为得到的128描述子向量，

$L=(l_1,l_2,\cdots,l_{128})$ 为规范化后的向量

$$l_j = w_j / \sqrt{\sum_{i=1}^{128} w_i} \qquad j=1,2,\cdots 128$$

关键点描述子向量的规范化可去除满足此模型的光照影响。对于图像灰度值整体漂移 ，图像各点的梯度是邻域像素相减得到，所以也能去除。

三、基于特征点匹配的目标检测

5）关键点匹配

- 首先进行相似性度量。

相似性度量是指用什么来确定待匹配特征之间的相似性，它通常是某种代价函数或距离函数的形式，如直方图匹配方法、欧式距离、马氏距离等；

- 其次消除错配。

无论采用何种特征描述符和相似性判定度量，错配难以避免。这一步主要做的就是根据几何或光度的约束信息去除候选匹配点中的错配。

三、基于特征点匹配的目标检测

关键点匹配方法

（1）欧式距离： $\mathrm{Dis}_{ij}=\left[\sum_{k=0}^{k=n}\left(X_{ik}-X_{jk}\right)^2\right]^{1/2}$

式中，X_{ik}和X_{jk}分别表示待配准图和参考图中第i个和第j个特征描述子的第k个元素，n表示特征向量的维数。

Brute Force匹配和FLANN匹配是opencv二维特征点匹配常见的两种办法，分别对应BFMatcher（BruteForceMatcher）和FlannBasedMatcher。

（2）BFMatcher::BFMatcher(int normType=NORM_L2, bool crossCheck=false)

normType：**NORM_L1, NORM_L2, NORM_HAMMING, NORM_HAMMING2.**
L1 and L2 norms are preferable choices for SIFT and SURF descriptors,
NORM_HAMMING should be used with ORB, BRISK and BRIEF,
NORM_HAMMING2 should be used with ORB。
crossCheck：If it is false, this is will be default BFMatcher behaviour when it finds the k nearest neighbors for each query descriptor. If crossCheck==true, then the BFMather will only return consistent pairs.

三、基于特征点匹配的目标检测

(3) FlannBasedMatcher（Fast Library forApproximate Nearest Neighbors，最近邻近似匹配），FlannBasedMatcher的构造函数如下：

```
class FlannBasedMatcher : public DescriptorMatcher
{
    public: FlannBasedMatcher(
        const Ptr<flann::IndexParams>& indexParams=new flann::KDTreeIndexParams(),
        const Ptr<flann::SearchParams>& searchParams=new flann::SearchParams() );
    virtual void add( const vector<Mat>& descriptors );
    virtual void clear(); virtual void train();
    virtual bool isMaskSupported() const;
    virtual Ptr<DescriptorMatcher> clone( bool emptyTrainData=false ) const;
    protected: ...
};
```

```
// matching
FlannBasedMatcher matcher;
vector<DMatch> matches;
matcher.match(descriptor_dst, descriptor_src, matches);
```

三、基于特征点匹配的目标检测

● SIFT算法特点

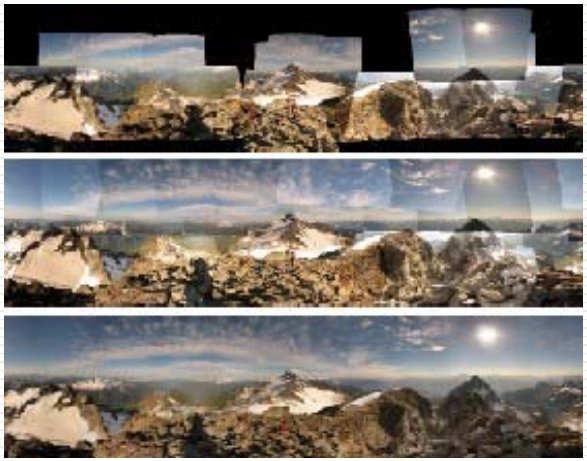
SIFT特征是图像的局部特征，其对旋转、尺度缩放、亮度变化保持不变性，对视角变化、仿射变换、噪声也保持一定程度的稳定性

● SIFT算法的应用

- 物体识别
- 图像拼接
- 三维建模
- 手势识别
- 视频跟踪
- 笔记鉴定
- 指纹与人脸识别
- 犯罪现场特征提取

物体识别

图像拼接



三、基于特征点匹配的目标检测

- 来自网友的创意——周正龙的老虎

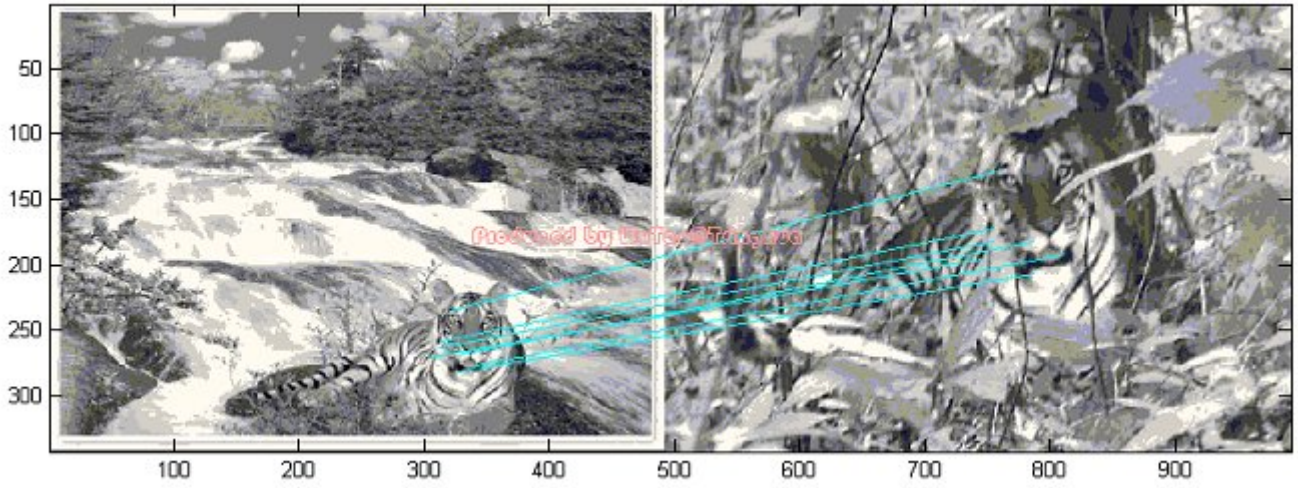
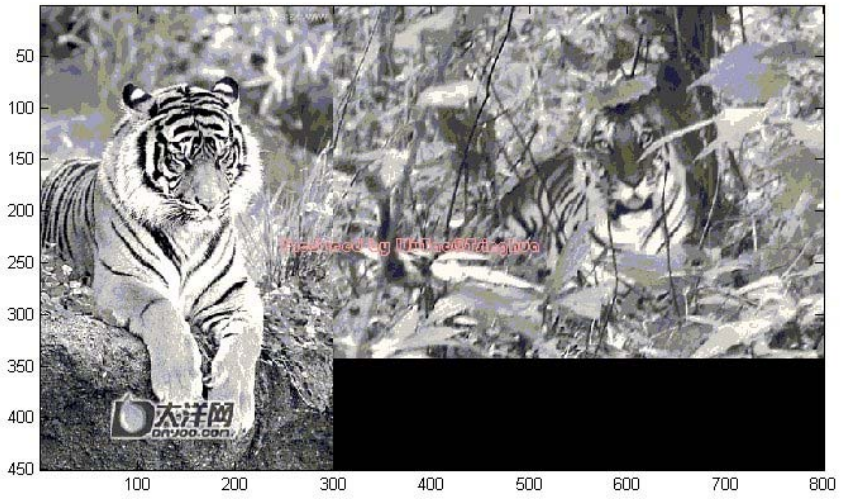


图1周正龙的华南虎照片与年画上的华南虎照片12点匹配

图2周正龙的华南虎照片与真实的华南虎照片0点匹配



三、基于特征点匹配的目标检测

（2）BRIEF描述子原理：

1）以特征点P为中心，取一个S×S大小的Patch邻域；

2）在邻域内随机取N对点，然后对这2N个点分别做高斯平滑。定义τ测试，比较N对像素点的灰度值的大小，N=128、256、512；

$$\tau(\mathbf{p};\mathbf{x},\mathbf{y}) := \begin{cases} 1 & \text{if } \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 & \text{otherwise} \end{cases}$$

3）最后把N个二进制码串组成一个N维向量即可；

$$f_{n_d}(\mathbf{p}) := \sum_{1 \le i \le n_d} 2^{i-1}\, \tau(\mathbf{p};\mathbf{x}_i,\mathbf{y}_i)$$

三、基于特征点匹配的目标检测

BRIEF描述子随机取N对点的方法：

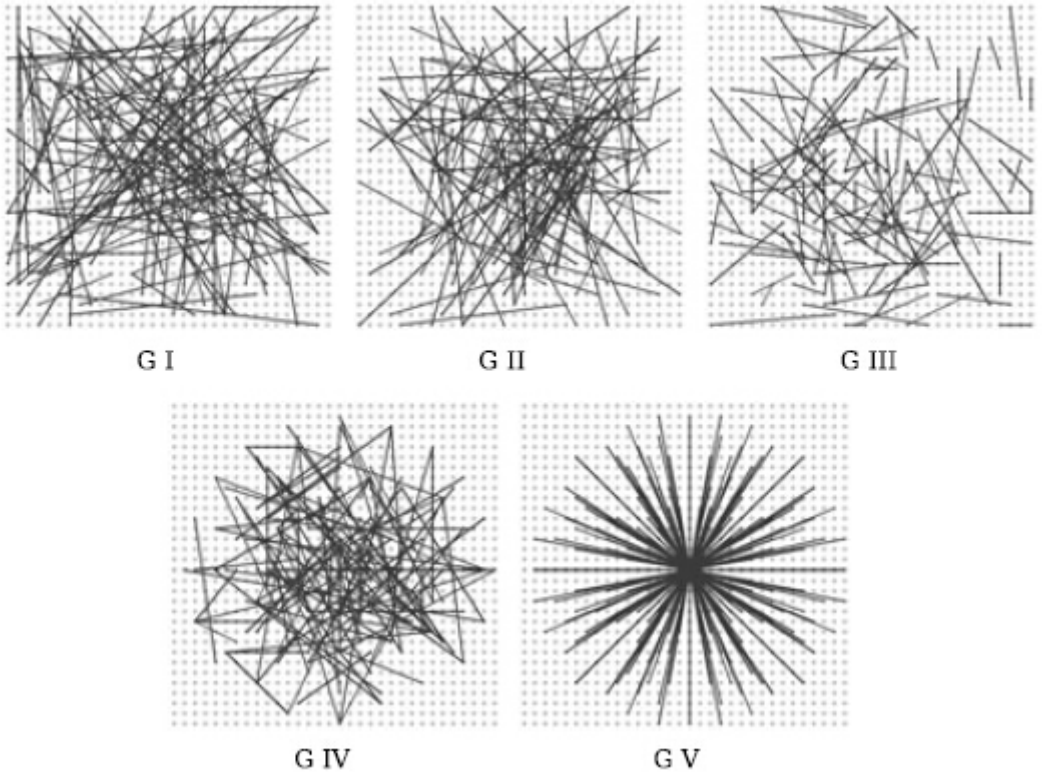
GI:X,Y即均匀分布；

GII: X,Y均服从高斯分布；

GIII: 先随机取X点，再以X点为中心，取Y点；

GIV: 在空间量化极坐标系下，随机取2N个点；

GV: X固定在中心，在Patch内，Y在极坐标系中尽可能取所有可能的值；



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。

三、基于特征点匹配的目标检测

总结：

1、优点：

- BRIEF抛弃了传统的用梯度直方图描述区域的方法，改用检测随机响应，大大加快了描述子建立速度；
- 生成的二进制描述子便于高速匹配（计算汉明距离只需通过异或操作加上统计二进制编码中“1”的个数，通过底层运算即可实现），且便于在硬件上实现。

2、缺点：不具备旋转不变性，不具备尺度不变性，容易受噪声影响 。

三、基于特征点匹配的目标检测

(3) BRISK描述子

BRISK(Binary Robust Invariant Scalable Keypoints)是BRIEF算法的一种改进，也是一种基于二进制编码的特征描述子，而且对噪声鲁棒，具有尺度不变性和旋转不变性。

1)特征点检测

BRISK主要利用FAST算法进行特征点检测，为了满足尺度不变性，BRISK构造图像金字塔在多尺度空间检测特征点。

2)特征点描述

给定一组特征点，BRISK通过比较邻域Patch内像素点对的灰度值，并进行二进制编码得到特征描述子。为了满足旋转不变性，需要选取合适的特征点主方向。

3)特征点匹配

BRISK特征匹配和BRIEF一样，都是通过计算特征描述子的汉明距离来实现。

BRISK算法具有较好的尺度不变性、旋转不变性，以及抗噪性能。计算速度优于SIFT、SURF，而次于ORB。

三、基于特征点匹配的目标检测

	SIFT	SURF	BRIEF	ORB	BRISK
提点方法	DoG的最值点位置再通过二次拟合确定位置	Hessian矩阵的行列式最值	无	使用FAST提点，使用Harris Corner去除非角点	使用FAST或AGAST提点
确定方向	特征邻域梯度直方图的最值方向	特征邻域对Haar wavelet的最大响应方向	无	使用Intensity centroid方法来确定方向	使用邻域随机抽样点对，对远点对做梯度确定方向
确定尺度	通过建立确定尺度空间，尺度空间中DoG最值所在尺度为特征尺度	尺度空间中Hessian矩阵行列式最值所在尺度	无	无	尺度空间中FAST提点最显著的尺度
描述方法	在特征周围取一个region，分成4*4的sub-region，对每个sub-region使用八方向的梯度表示，总共128维	在特征周围取一个region，分成4*4的sub-region，对每个sub-region计算haar响应，分别取x方向响应和，x方向响应绝对值之和，y方向响应和，y方向绝对值之和四个值描述，总共64维	在特征点周围随机抽取随机点对，比较两个点的像素强度，根据结果大小记为1或0，取256组组成256位的二进制字符串	通过贪心方法抽取符合正态分布的随机点对，其他同BRIEF	使用短距离点对进行强度匹配，组成512位的二进制字符串

三、基于特征点匹配的目标检测

（4）SURF（Speeded Up Robust Features）算法

SURF是一种高鲁棒性的局部特征点检测器。由Herbert Bay 等人在2006年提出。该算法可以用于计算机视觉领域例如物体识别或者三维重建。根据作者描述该算法比SIFT更快更具有鲁棒性。

该算法中采用积分图、近似的Hessian矩阵和Haar小波变换运算来提高时间效率，采用Haar小波变换增加鲁棒性。

步骤：兴趣点检测、兴趣点的描述与匹配。

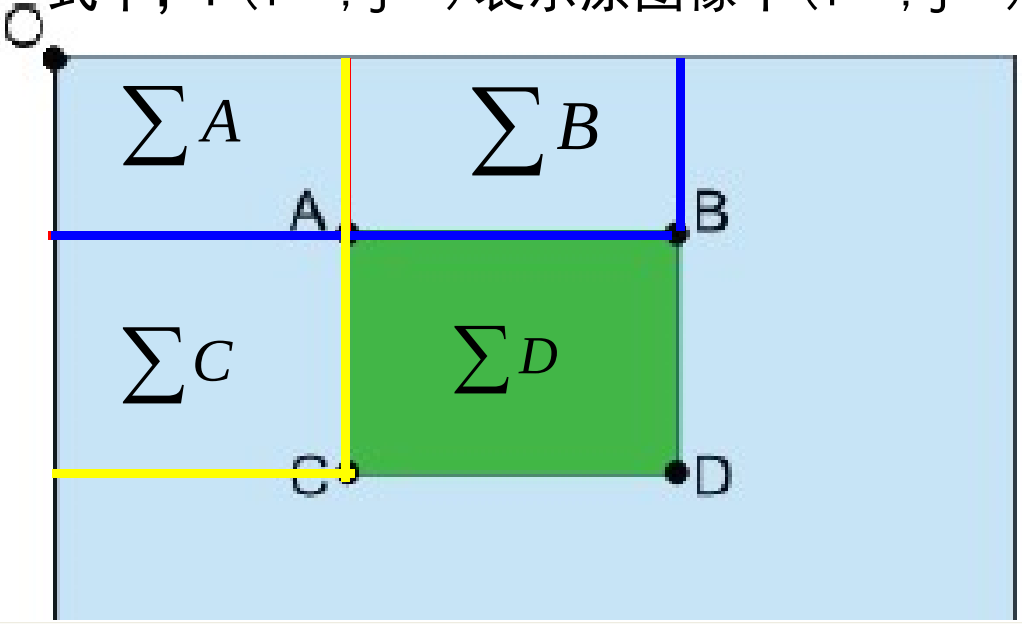
四、基于特征点匹配的目标检测

积分图像

积分图像中任意一点I(i, j)的值n(i, j)，为原图像左上角到该任意点I(i, j)相应的对角线区域灰度值的总和，即：

$$n(i,j)=\sum_{i'<i,j'<j} p(i',j')$$

式中，P(i’，j’)表示原图像中(i’，j’)的灰度值



S_{ABCD}=D−C−B+A

四、基于特征点匹配的目标检测

1）构建尺度空间

图像滤波

二维高斯对应的x阶导数公式：

$$G(x,y,\sigma)=\frac{1}{2\pi\sigma^2}e^{-(x^2+y^2)}/2\sigma^2$$

$$G_x(x,y,\sigma)=-\frac{x}{\sigma^2}G(x,y,\sigma)$$

$$G_{xx}(x,y,\sigma)=\frac{x-\sigma^2}{\sigma^4}G(x,y,\sigma)$$

$$G_{xy}(x,y,\sigma)=\frac{xy}{\sigma^4}G(x,y,\sigma)$$

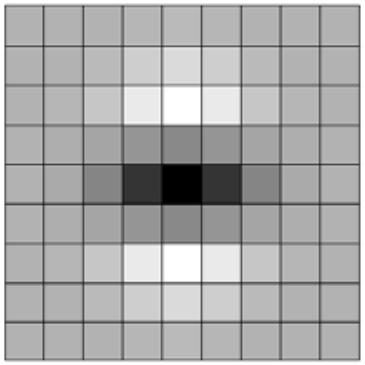
$L_{xx}(X,\sigma)$ 是Gaussian二阶偏导数 $\frac{\partial^2 f}{\partial x^2}$ 在x处与图像f的卷积。

$$L_{xx}(X,\sigma)=I(x,y)\cdot G_{xx}(x,y,\sigma)$$

$$L_{yy}(X,\sigma)=I(x,y)\cdot G_{yy}(x,y,\sigma)$$

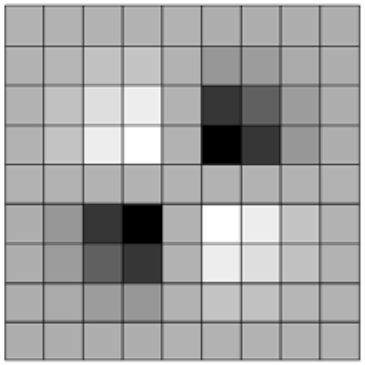
$$L_{xy}(X,\sigma)=I(x,y)\cdot G_{xy}(x,y,\sigma)$$

四、基于特征点匹配的目标检测



$L_{yy}(X,\sigma)$

模板9×9



$L_{xy}(X,\sigma)$

模板9×9

为了加速卷积，采用盒子型滤波器对上面高斯滤波器进行近似。

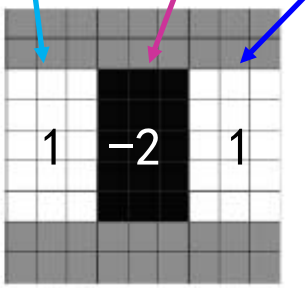
HTemplate[]={

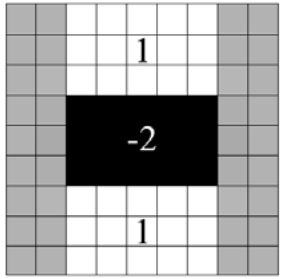
3,0,2,2,6,15,1, 3,2,5,6,15,-2, 6,2,8,6,15,1, 0,0,0,0,0,0, ……}

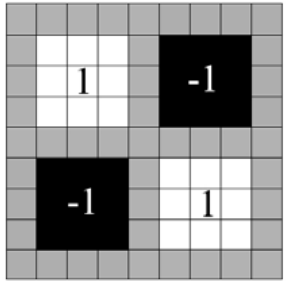
第1位：滤波器盒子数量　　第2、3位：第1个盒子左上角坐标

第4、5位：第1个盒子右下角坐标　　第6位：盒子面积

第7位：盒子填充值　　第8位：第二个盒子







$$D_{xx} = I \cdot H_{xx}$$

$$D_{yy} = I \cdot H_{yy}$$

$$D_{xy} = I \cdot H_{xy}$$

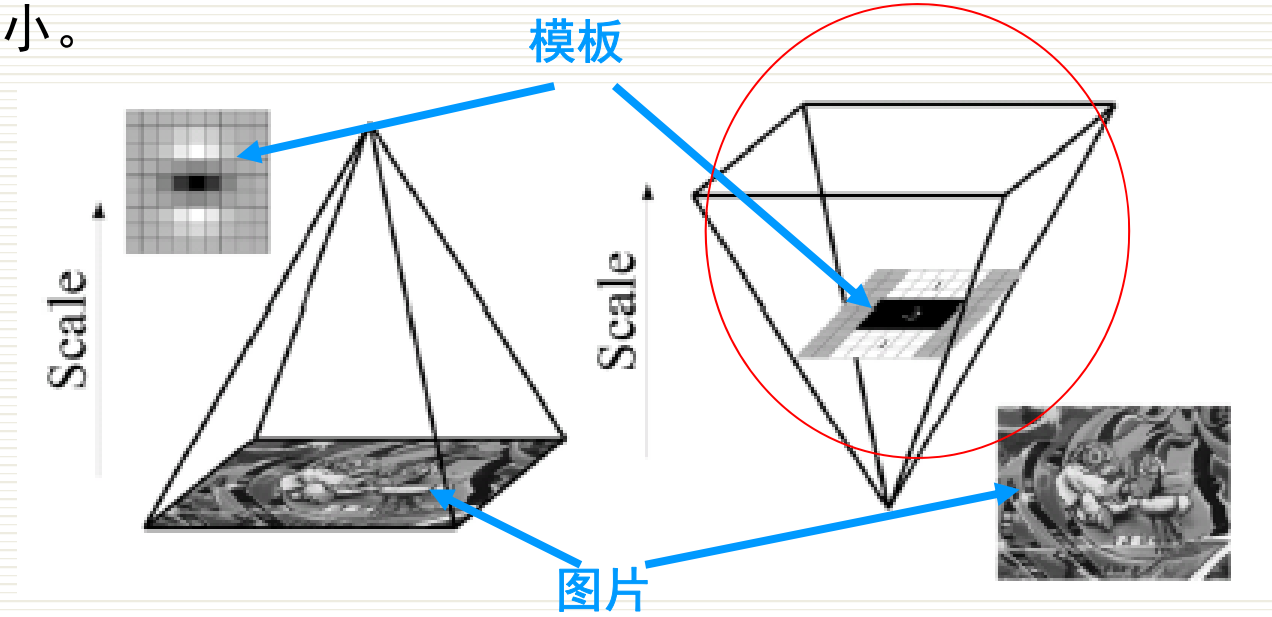
四、基于特征点匹配的目标检测

尺度空间表示

算法的尺度不变性主要在不同尺度下寻找感兴趣点。

SIFT构造尺度空间：对原图像不断的进行Gauss平滑和降采样，得到金字塔图像后，通过差分进一步得到DOG图，通过DOG图得到点在原图的位置。

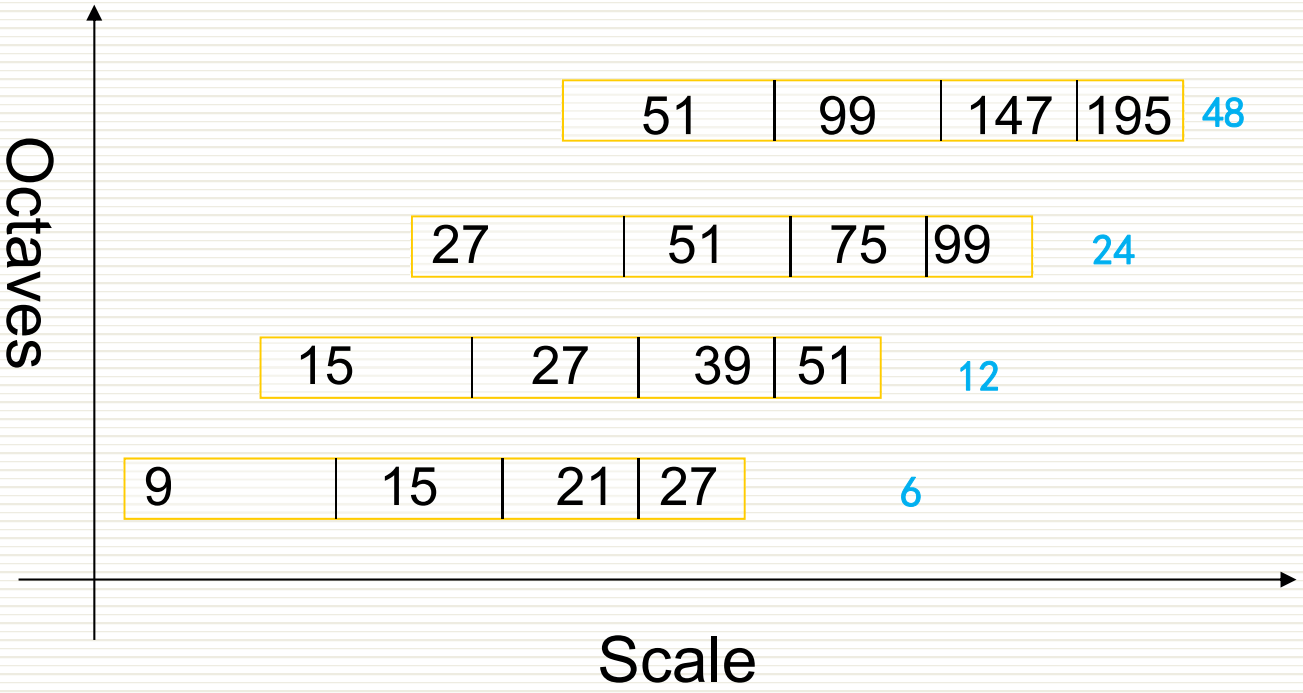
SURF构造尺度空间：SIFT算法在构造金字塔图层时Gauss滤波器大小不变，改变的是图像的大小，而SURF则相反，图像大小保持不变，改变的是滤波器的大小。

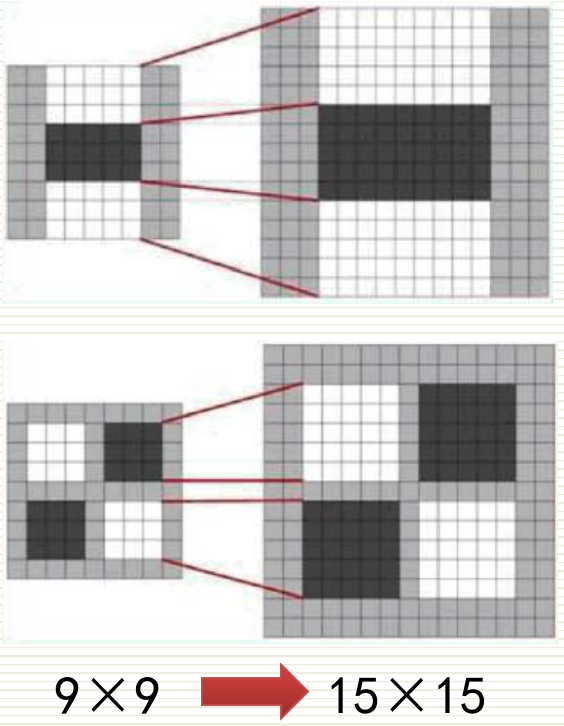


SIFT构建尺度空间　　SURF构建尺度空间

四、基于特征点匹配的目标检测

与SIFT相类似，SURF也将尺度空间划分成若干组(Octaves)。一个组代表了逐步放大的滤波模板对同一个输入图像进行滤波的一系列响应图像。每一组又有若干固定的层组成。





尺度空间的第一组采用大小为9×9的filter，接下来的filter大小依次是15×15, 21×21, 27×27。

四、基于特征点匹配的目标检测

2）兴趣点检测与定位

用于检测特征点的Hessian矩阵

在SURF中，采用近似的Hessian矩阵行列式的局部最大值来定位感兴趣点的位置。当Hessian行列式的局部值最大的时候，所检测出来的就是感兴趣点。感兴趣点的特征为比周围邻域更亮或者更暗一些。

给定图像f(x,y)中一个点（x,y），其Hessian矩阵H(x,σ)定义如下：

$$H(f(x,y))=\begin{bmatrix}\frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x\partial y}\\ \frac{\partial^2 f}{\partial x\partial y} & \frac{\partial^2 f}{\partial y^2}\end{bmatrix}$$

$$H(X,\sigma)=\begin{bmatrix}L_{xx}(X,\sigma) & L_{xy}(X,\sigma)\\ L_{yx}(X,\sigma) & L_{yy}(X,\sigma)\end{bmatrix}$$

位置　　尺度

$L_{xx}(X,\sigma)$ 是Gaussian二阶偏导数 $\frac{\partial^2 f}{\partial x^2}$ 在x处与图像f的卷积。

四、基于特征点匹配的目标检测

$$det(H) = \frac{\partial^2 f}{\partial x^2}\frac{\partial^2 f}{\partial y^2} - \left(\frac{\partial^2 f}{\partial x \partial y}\right)^2$$

$$\det(H_{approx}) = D_{xx}\,D_{yy} - (wD_{xy})^2$$

加权系数

Lxx是高斯模板与图像卷积

Dxx是盒子模板与图像卷积

用Dxx近似代替Lxx

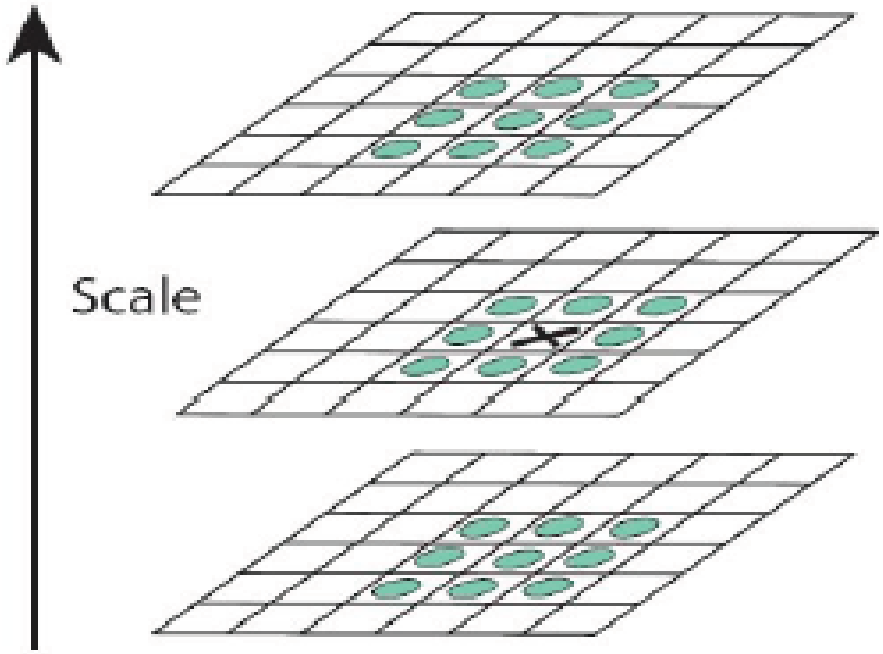
W选为0.9

如果行列式的符号为负，则特征值有不同的符号，则不是局部极值点。

如果行列式的符号为正，则该行列式的两个特征值同为正或负，所以该点可以归类为极值点。

四、基于特征点匹配的目标检测

为了在目标影像上确定SURF特征点，使用了3*3*3的模板在3维尺度空间进行非最大化抑制，根据预设的Hessian阈值H，当h大于H，而且比临近的26个点的响应值都大的点才被选为兴趣点。**最后再进行插值精确**。



四、基于特征点匹配的目标检测

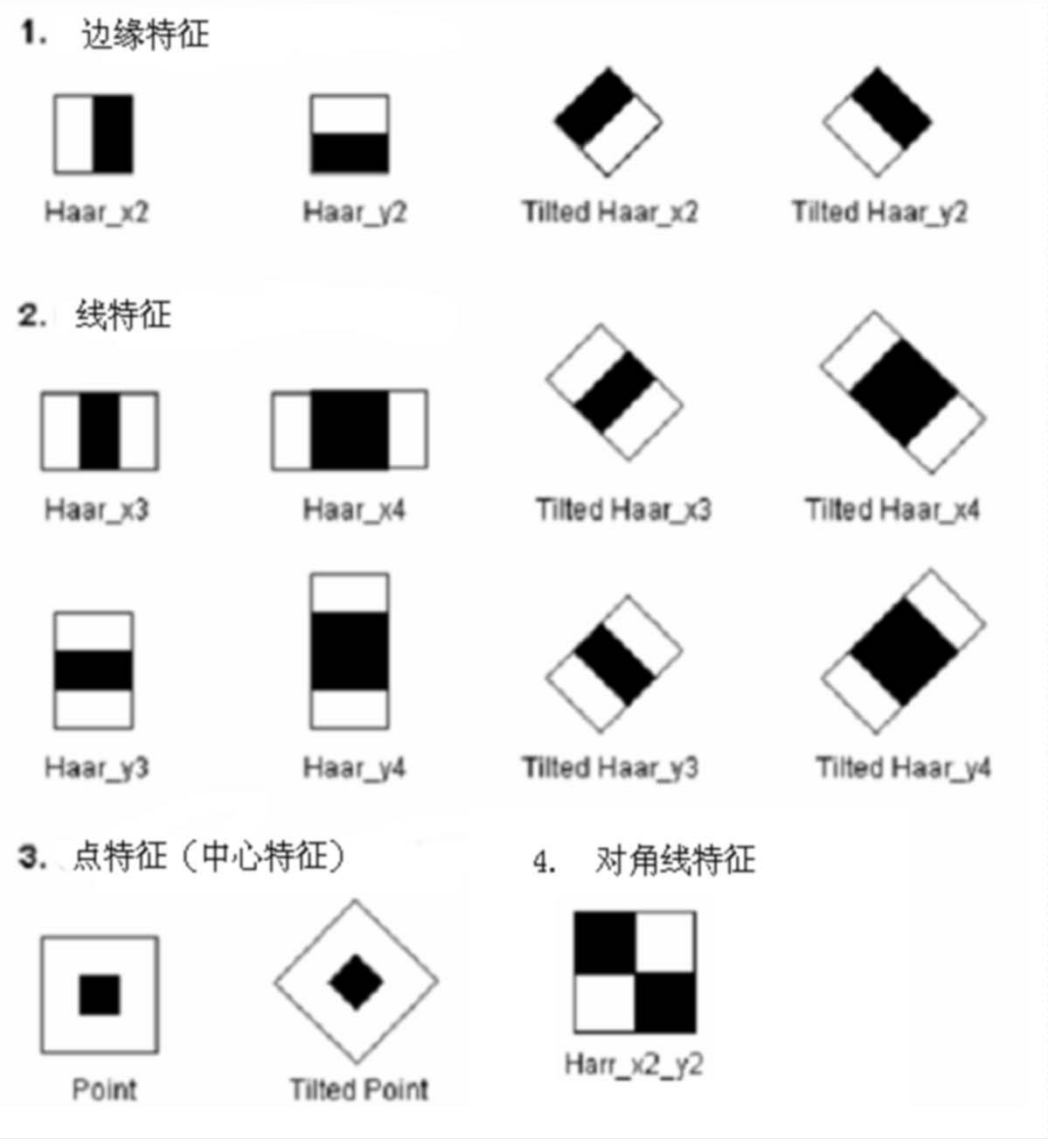
3）兴趣点的描述

Haar特征

常用的Haar-like特征：

- 边缘特征有4种：x方向，y方向，x倾斜方向，y倾斜方向；
- 线特征有8种；
- 点特征有2种；
- 对角线特征有1种。

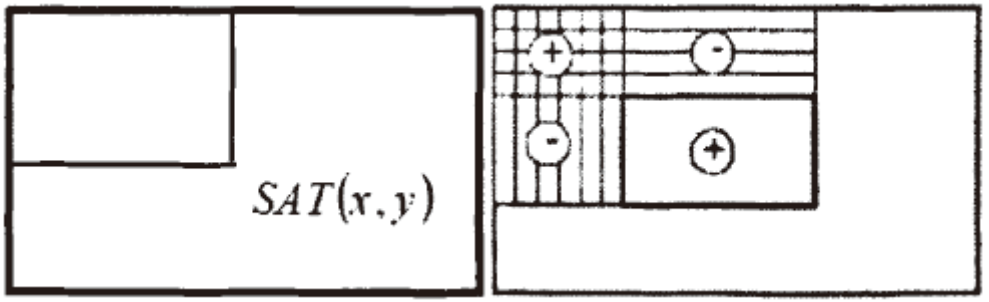
每一种特征的计算都是由黑色填充区域的像素值之和与白色填充区域的像素值之和的差值。而计算出来的这个差值就是所谓的Haar-like特征的特征值。



四、基于特征点匹配的目标检测

利用积分图像法快速计算Haar特征

- **构造积分图像**



积分图像中，每个点存储是其左上方所有像素之和：

$$SAT(x,y)=\sum_{x'\le x,y'\le y} I(x',y')$$

积分图像可以采用增量的方式计算：

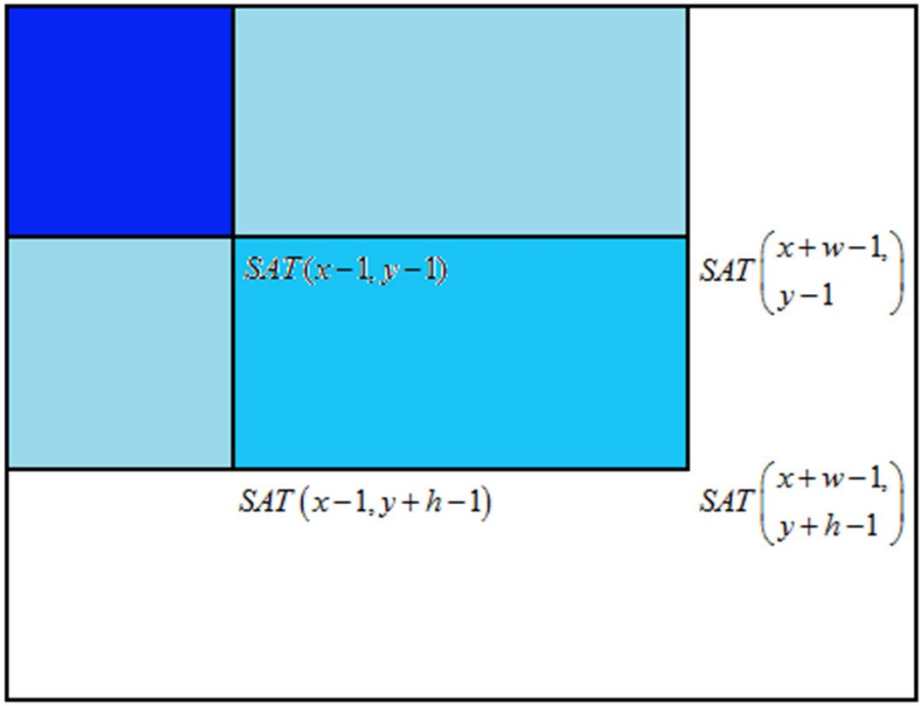
$$SAT(x,y)=SAT(x,y-1)+SAT(x-1,y)+I(x,y)-SAT(x-1,y-1)$$

初始边界：SAT(-1,y)=SAT(x,-1)=SAT(-1,-1)=0

所以，只需要对整张图像遍历一次就可以求得这张图的积分图像。

四、基于特征点匹配的目标检测

- 计算Haar矩形特征



利用积分图计算可计算矩形区域内像素和：

$$RecSum(r) = SAT(x-1, y-1) + SAT(x+w-1, y+h-1) - SAT(x-1, y+h-1) - SAT(x+w-1, y-1)$$

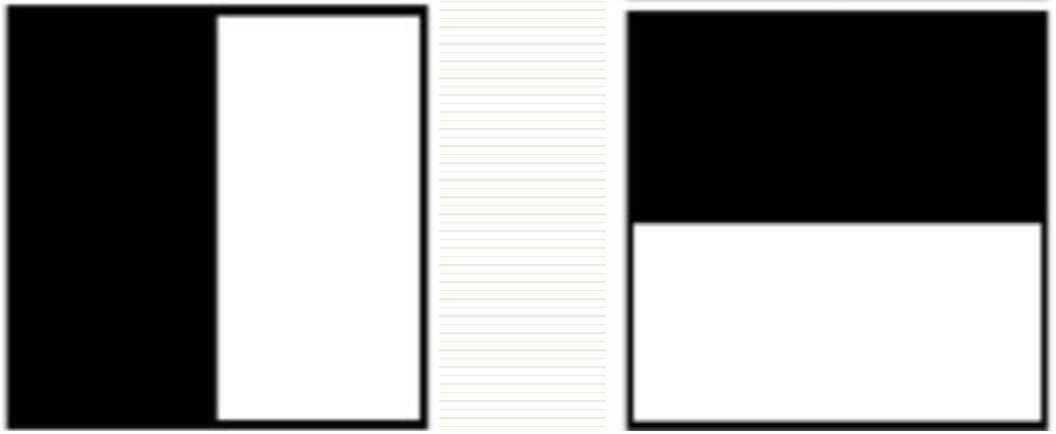
所以，无论矩形的尺寸大小，只需查找积分图像4次就可以求得任意矩形内像素值的和。

四、基于特征点匹配的目标检测

方向分配

为了保证特征矢量具有旋转不变形，需要对每一个特征点分配一个主要方向。方法为：

- 以特征点为中心，以6s（s为特征点所处尺度）为半径的圆形区域内计算在x、y方向的haar小波响应。
- 采样步长设为s，小波的size设为2s

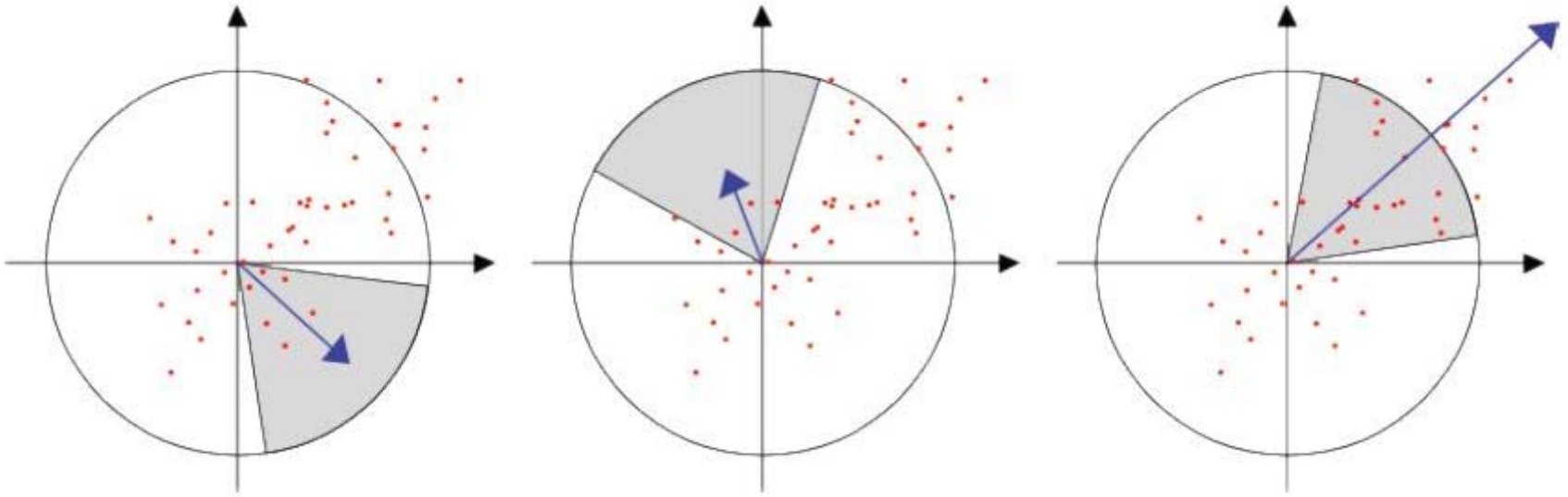


用来计算x、y方向上响应的haar小波滤波器

四、基于特征点匹配的目标检测

小波响应计算完毕后还要以兴趣点为中心进行高斯加权（σ=2s），即兴趣点的描述可用一个点坐标（x，y）表示，其中x表示x方向上的响应，y表示y方向上的响应。

为了求取主方向值，在兴趣点的邻域(如半径为6s的圆内，s为该点所在的尺度)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，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兴趣点的主方向。

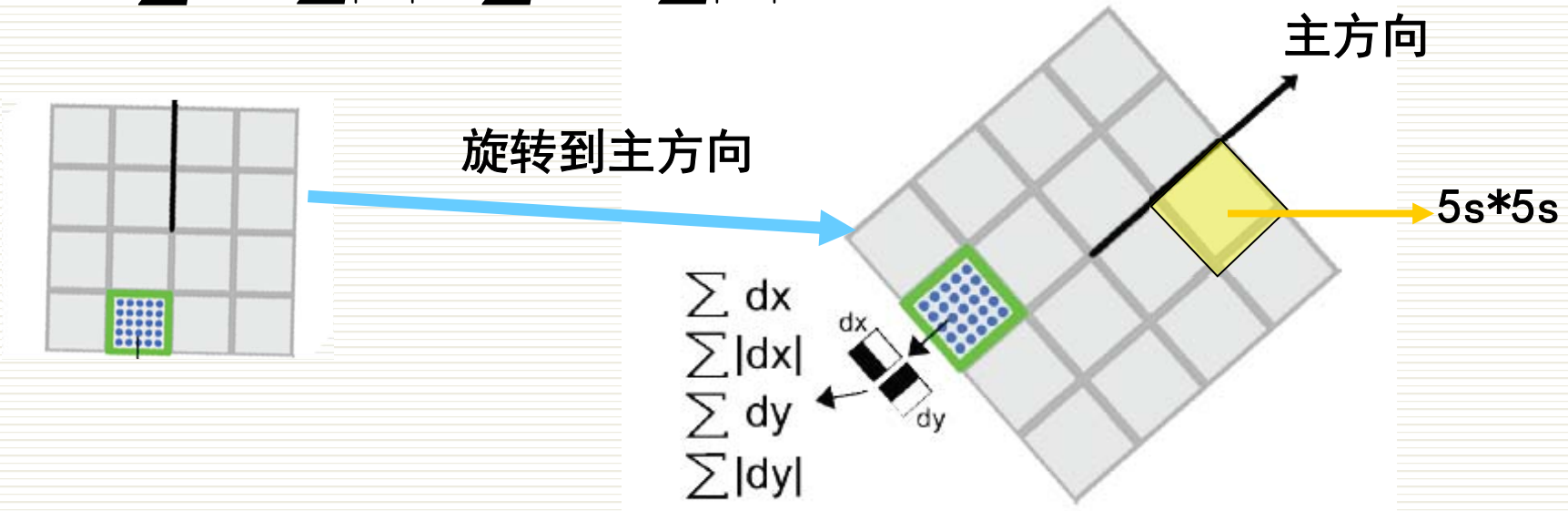


四、基于特征点匹配的目标检测

基于Haar小波响应的Descriptor

对于Descriptor的提取，第一步是构造一个中心点在兴趣点附近，带方向（方向即前面估计的方向）的方框，方框的大小设为20s。

在矩形区域内，以特征点为中心，沿主方向将20s*20s的图像划分成4*4个子块，对每个子域计算25（5*5）个空间归一化的采样点利用尺寸2s的Haar小波模板进行响应计算。Dx表示水平方向上的Haar小波响应，dy表示垂直方向上的，然后对响应值进行统计 $\sum dx$ ，$\sum |dx|$ ，$\sum dy$ ，$\sum |dy|$ 形成的特征矢量 。



每个特征点采用16*4=64维的向量，相比sift而已少了一半。

四、基于特征点匹配的目标检测

4）兴趣点的匹配

对于两个特征点描述子的相似性度量，采用欧式距离进行计算。

对于待配准图上的特征点，计算它到参考图像上所有特征点的欧氏距离，得到一个距离集合。通过对距离集合进行比较运算得到小欧氏距离和次最小欧式距离。设定一个阈值，一般为 0.8，当最小欧氏距离和次最小欧式距离的比值小于该阈值时，认为特征点与对应最小欧氏距离的特征点是匹配的，否则没有点与该特征点相匹配。

阈值越小，匹配越稳定，但极值点越少。

三、基于特征点匹配的目标检测

SIFT使用方法：

```
Ptr<SIFT> siftdetector = SIFT::create();
vector<KeyPoint> kp1, kp2;
siftdetector->detect(src1, kp1);
siftdetector->detect(src2, kp2);

Mat des1, des2;
siftdetector->compute(src1, kp1, des1);
siftdetector->compute(src2, kp2, des2);

BFMatcher matcher(NORM_L2,true);
vector<DMatch> matches;
matcher.match(des1,des2,matches);
```

三、基于特征点匹配的目标检测

ORB使用方法：

```
vector<KeyPoint> keypoints1, keypoints2;
Mat descriptors1, descriptors2;
Ptr<ORB> orb = ORB::create();

orb->detect(img1, keypoints1);
orb->detect(img2, keypoints2);

orb->compute(img1, keypoints1, descriptors1);
orb->compute(img2, keypoints2, descriptors2);

BFMatcher matcher(NORM_HAMMING,true);
vector<DMatch> matches;
matcher.match(obj_descriptors, scene_descriptors, matches);
```

三、基于特征点匹配的目标检测

BRISK使用方法：

```
Ptr<BRISK> detector = BRISK::create();
vector<KeyPoint> kp1, kp2;

detector->detect(src1, kp1);
detector->detect(src2, kp2);

Mat des1, des2;
detector->compute(src1, kp1, des1);
detector->compute(src2, kp2, des2);

BFMatcher matcher(NORM_HAMMING,true);
vector<DMatch> matches;
matcher.match(obj_descriptors, scene_descriptors, matches);
```

实验课一一编写程序

1、编写一种特征点匹配方法实现零件图像瑕疵检测

零件图像ORB特征提取结果：