

# SURF: Speeded-Up Robust Features

## ECCV2006, CVIU2008

Herbert Bay<sup>a</sup>, Andreas Ess<sup>a</sup>, Tinne Tuytelaars<sup>b</sup>,  
Luc Van Gool<sup>a,b</sup>

<sup>a</sup>ETH Zurich  
( 苏黎世理工学院 )  
Switzerland

<sup>b</sup>Katholieke Unviersiteit Leuven  
( 鲁汶大学 )  
Belgium

# 提纲

- SURF特点
- SURF检测
- SURF描述
- 实验及实验结果
- 结论

# SURF实现的主要方法

## ■ Detector:

- ❑ 积分图像(Integral Images)
- ❑ 基于Hessian Matrix的兴趣点检测
- ❑ 尺度空间表示
- ❑ 兴趣点定位

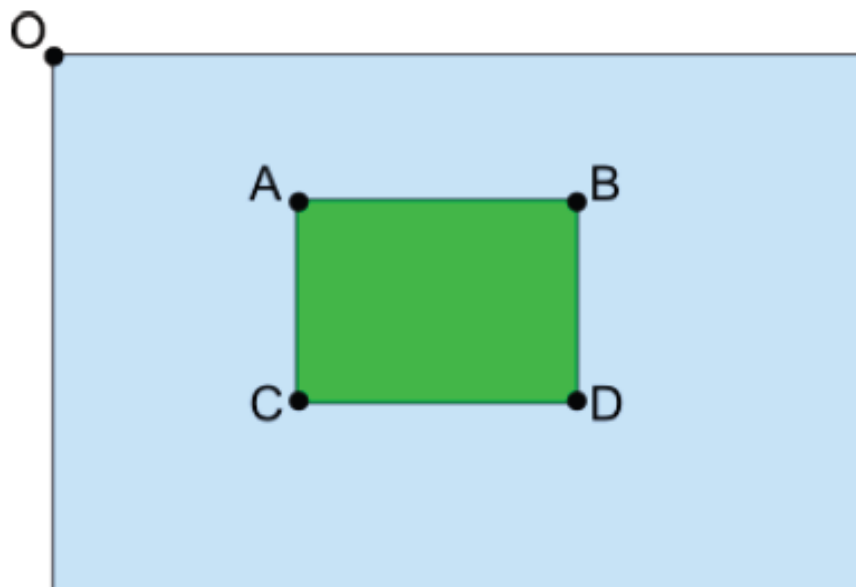
## ■ Descriptor:

- ❑ 指定方向
- ❑ 基于Haar小波响应和的描述子计算
- ❑ 描述子的快速索引匹配

Detector: 积分图像、Fast-Hessian、尺度空间、兴趣点定位

Descriptor: 主方向、描述子计算、快速索引匹配

# Detection\_积分图像(Integral Images)



$$I_{\Sigma}(x, y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(x, y)$$

$$S = A - B - C + D$$

Second order derivative and Haar-wavelet response

Detector: 积分图像、Fast-Hessian、尺度空间、兴趣点定位

Descriptor: 主方向、描述子计算、快速索引匹配

# Detection\_ Fast-Hessian Detector

Hessian矩阵: 
$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix}$$

$L_{xx}(x, \sigma)$  是图像 $I$ 和二阶高斯差分 $\frac{\partial^2}{\partial x^2} g(\sigma)$ 的卷积，即图像的

多尺度Gaussian Laplacian结果。

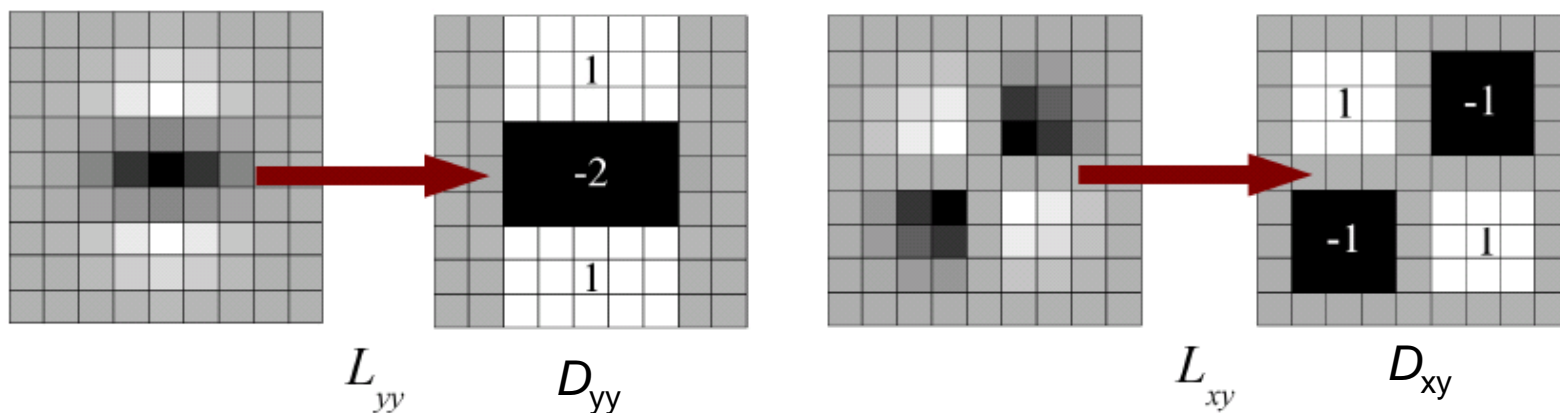
本文不认为Lindeberg高斯尺度空间理论在任何情况下都完全正确，并认为其可以进行简化，本文采用了box filters简化代替LOG即 $D_{xx}$ 近似 $L_{xx}$

$9 \times 9$ 的box filters近似Gaussian在 $\sigma = 1.2$ 的二阶差分，并作为最小尺度（最高分辨率）

Detector: 积分图像、Fast-Hessian、尺度空间、兴趣点定位

Descriptor: 主方向、描述子计算、快速索引匹配

# Detection\_ Fast-Hessian Detector



能量转换:

$$\frac{|L_{xy}(1.2)|_F \times |D_{xx}(9)|_F}{|L_{xx}(1.2)|_F \times |D_{xy}(9)|_F} = 0.912... \approx 0.9$$

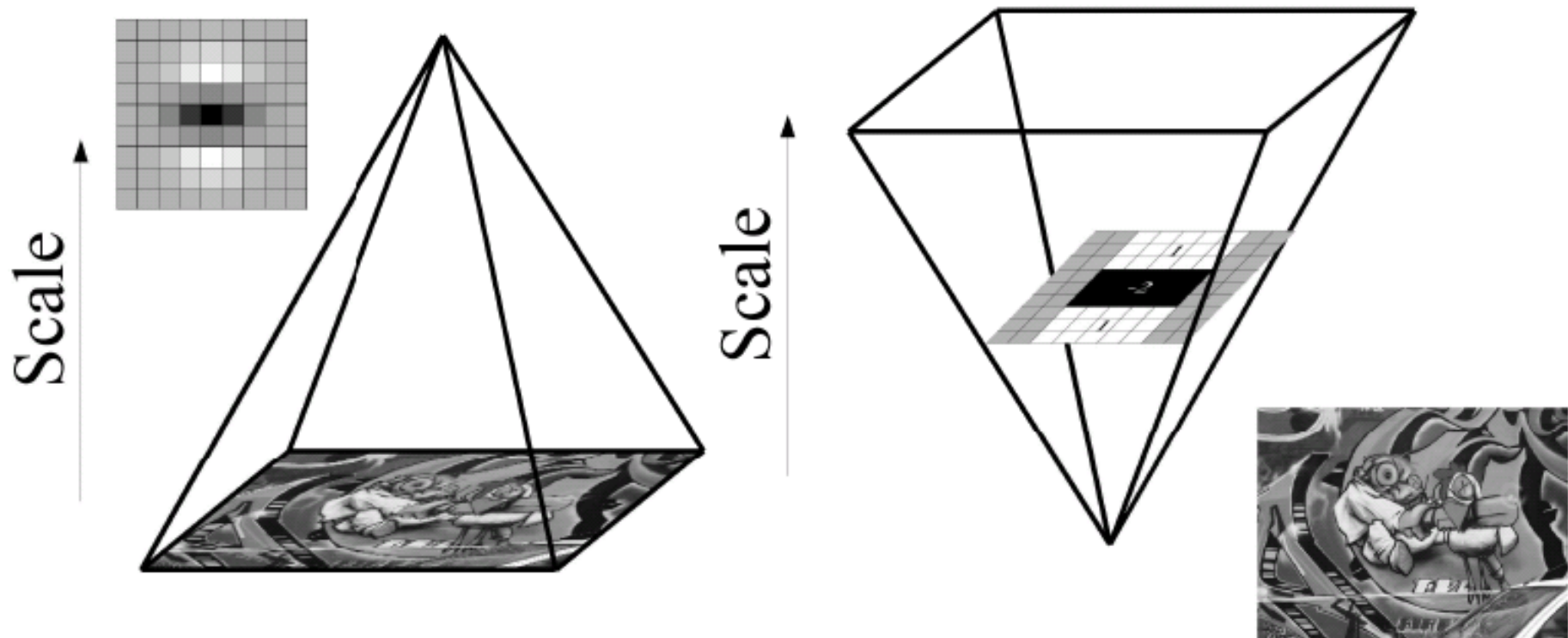
Blob响应

$$\det(H_{approx}) = D_{xx} D_{yy} - (0.9 D_{xy})^2$$

Detector: 积分图像、Fast-Hessian、尺度空间、兴趣点定位

Descriptor: 主方向、描述子计算、快速索引匹配

# Detection\_ 尺度空间表示



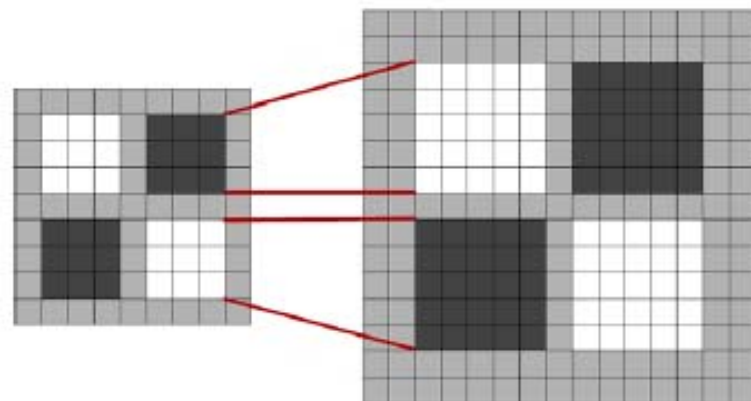
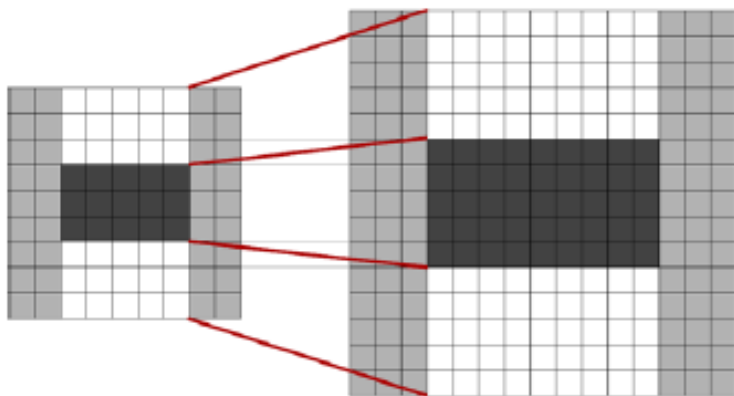
传统描述子（SIFT）尺度空间保持filter不变，依赖上层结果，改变图像大小

SURF保持图像大小不变，改变filter的尺度大小，提高速度和精度

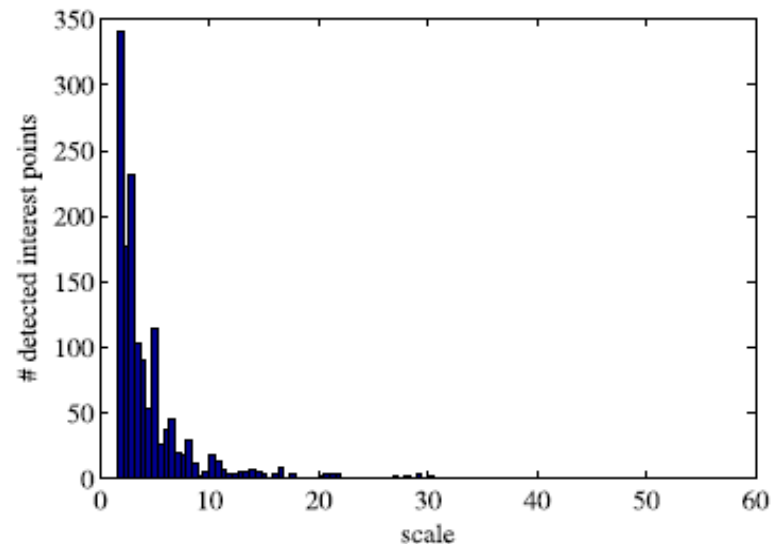
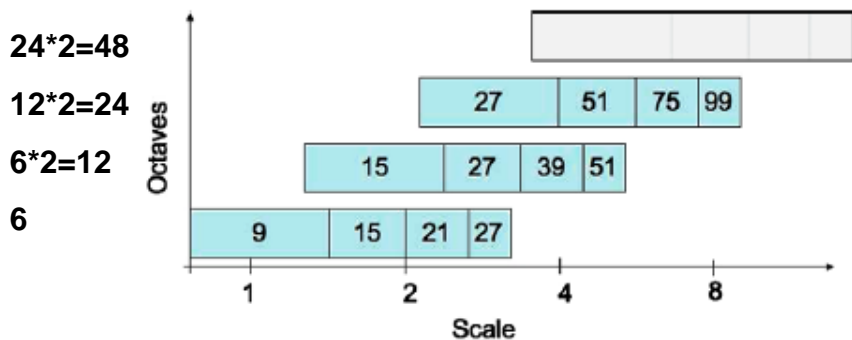
Detector: 积分图像、Fast-Hessian、尺度空间、兴趣点定位

Descriptor: 主方向、描述子计算、快速索引匹配

# Detection\_ 尺度空间表示



$$\begin{aligned}\sigma_{approx} &= \text{Current Filter Size} \cdot \frac{\text{Base Filter Scale}}{\text{Base Filter Size}} \\ &= \text{Current Filter Size} \cdot \frac{1.2}{9}\end{aligned}$$



Detector: 积分图像、Fast-Hessian、尺度空间、兴趣点定位

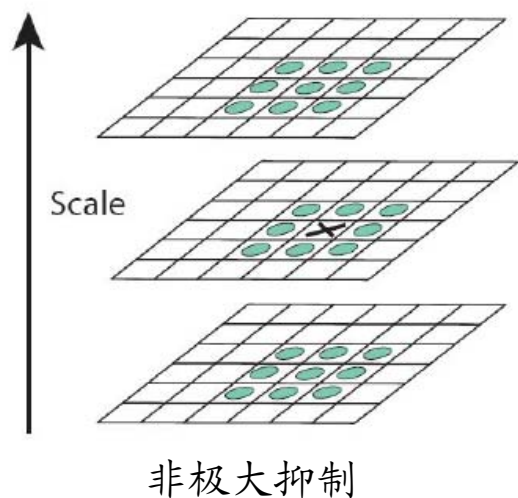
Descriptor: 主方向、描述子计算、快速索引匹配



# Detection\_ 兴趣点定位

Steps:

1. 设置阈值，确定候选兴趣点
2. 非极大抑制
3. 内插子像素精确定位



拟合3D二次曲线内插得到 $x, y$ 和  $\sigma$

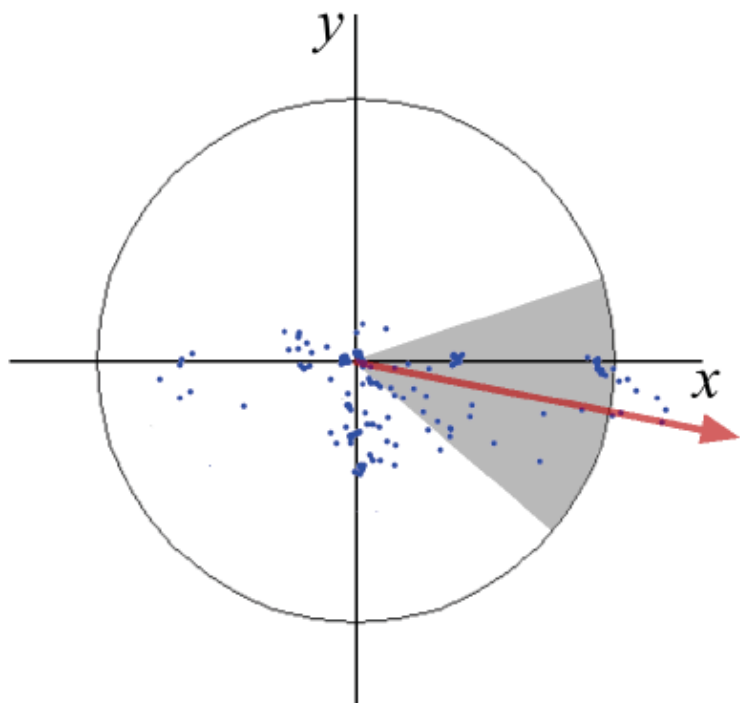
$$H(\mathbf{x}) = H + \frac{\partial H^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 H}{\partial \mathbf{x}^2} \mathbf{x}$$
$$\hat{x} = -\frac{\partial^2 H^{-1}}{\partial \mathbf{x}^2} \frac{\partial H}{\partial \mathbf{x}}$$

子像素精确定位

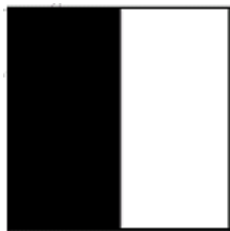
Detector: 积分图像、Fast-Hessian、尺度空间、兴趣点定位

Descriptor: 主方向、描述子计算、快速索引匹配

# Description\_指定方向



x response

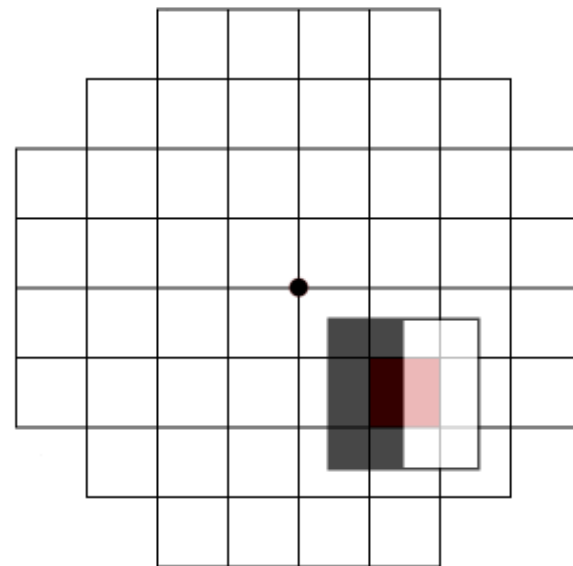


y response



Side length =  $4s$

采用积分图像，6次模版滤波操作可完成响应的计算（利用积分图像进行加减运算）



兴趣点周围的 $6s$ 半径圆范围的采样点  
( $s$  = the scale at which the point was detected)

Detector: 积分图像、Fast-Hessian、尺度空间、兴趣点定位

Descriptor: 主方向、描述子计算、快速索引匹配

# Description\_指定方向

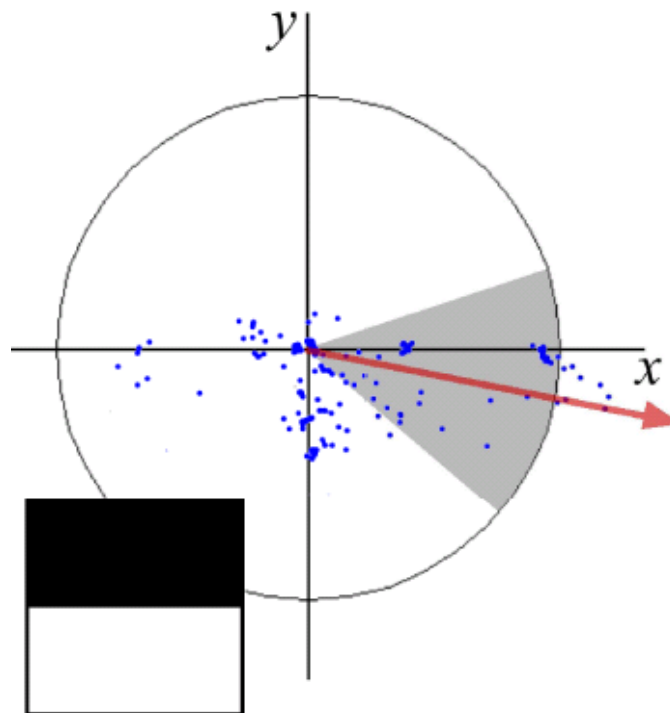
响应以兴趣点为中心进行高斯加权(2.5s)

Haar小波响应以矢量形式表示

在  $\pi/3$  范围的滑动窗口内计算响应的和

水平方向和垂直方向分别求和，两个响应和产生一个新的矢量

最长矢量对应的扇形方向作为兴趣点的主方向



Detector: 积分图像、Fast-Hessian、尺度空间、兴趣点定位

Descriptor: 主方向、描述子计算、快速索引匹配

# Description\_描述子计算

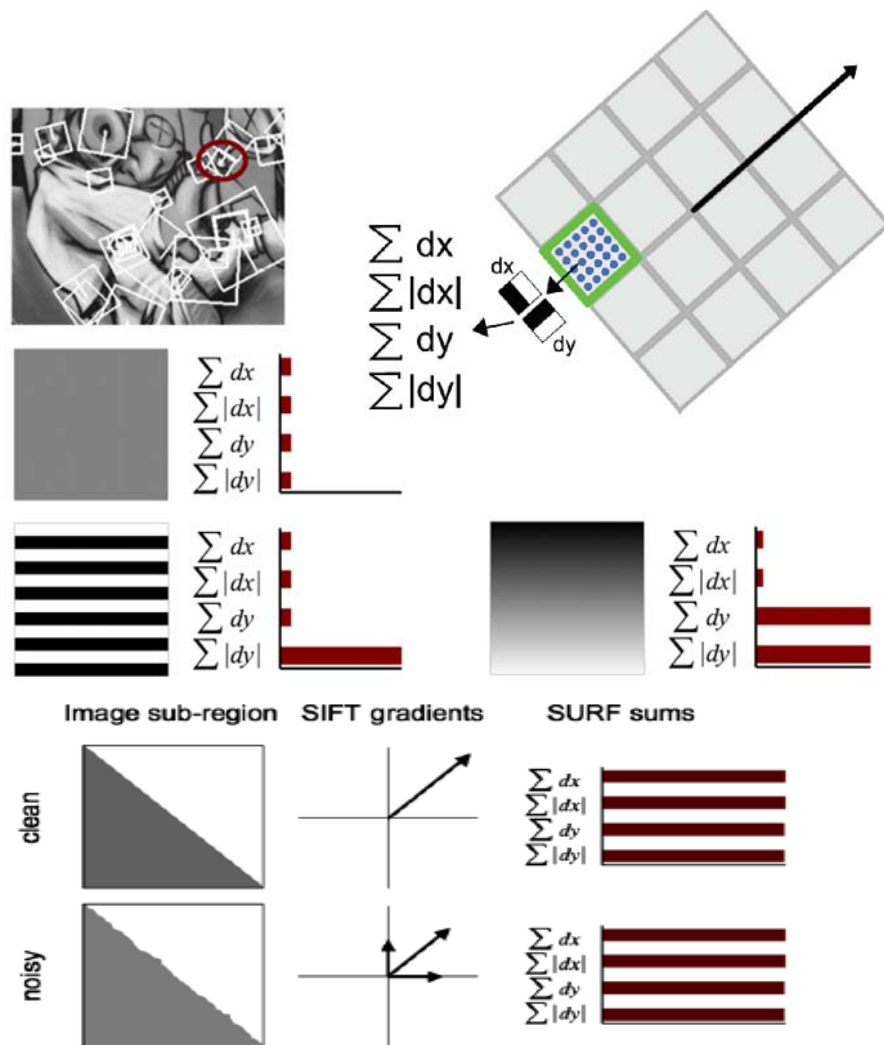
1.确定兴趣区域(20s x 20s)，并切分为4×4的方形区域

2.由指定的主方向，在5×5采样点上计算Harr小波响应dx和dy

3.对响应dx和dy进行高斯加权(以兴趣点为中心， $\sigma = 3.3s$ )

4.分别对子区域的dx,dy,|dx|,|dy|响应求和

5.归一化为单位向量



Detector: 积分图像、Fast-Hessian、尺度空间、兴趣点定位

Descriptor: 主方向、描述子计算、快速索引匹配

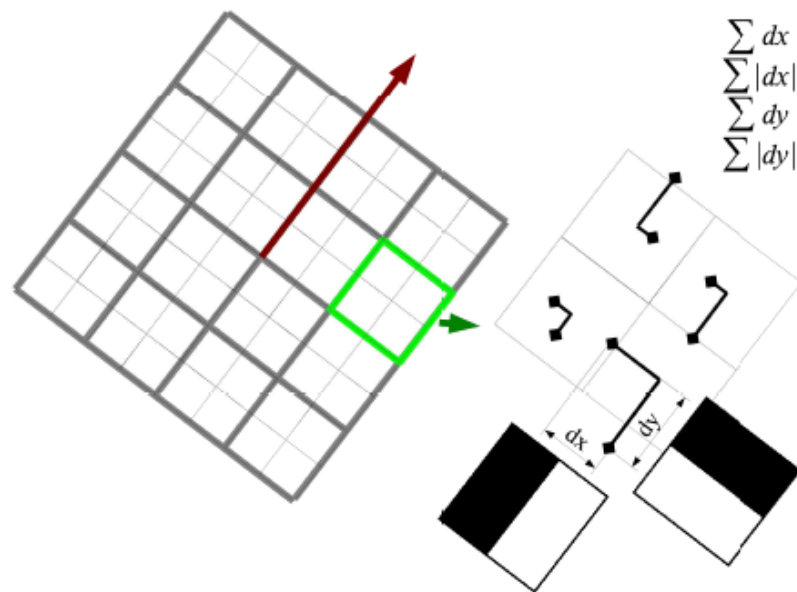
# Description\_描述子计算

向量长度:  $4 \times 4 \times 4 = 64$ , 以该向量作为SURF描述子, 对旋转, 尺度, 亮度和对比度变化具有不变性

SURF-36:  $3 \times 3$ 子区域, distinctive差, 匹配速度快

SURF-128: 在 $dy > 0$ 和 $dy < 0$ 情况下分别计算 $\sum dx$ 、 $\sum |dx|$ 。 $\sum dy$ 、 $\sum |dy|$ 同样分两种情况计算, 这样得到128维矢量, distinctive强, 匹配速度慢

$$v_{subregion} = \left[ \sum dx, \sum dy, \sum |dx|, \sum |dy| \right]$$



Detector: 积分图像、Fast-Hessian、尺度空间、兴趣点定位

Descriptor: 主方向、描述子计算、快速索引匹配

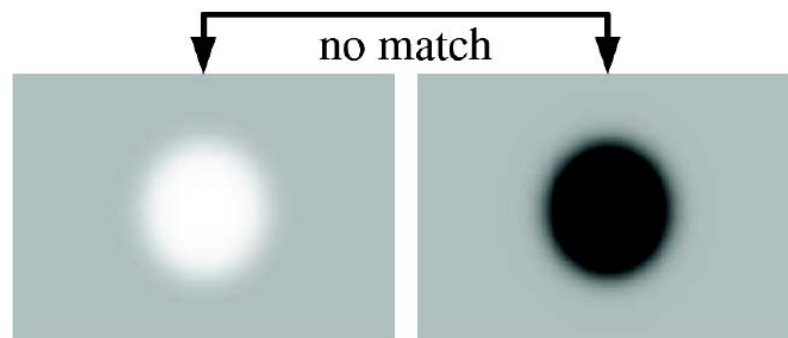
# Description\_快速索引匹配

利用Laplacian的符号信息，可以进行快速索引匹配

利用Hessian矩阵trace（反映了对比类型）的符号

$$\text{Trace} = L_{xx} + L_{yy}$$

匹配相同trace符号的描述子，提高速度



# 实验及实验结果(时间比较)

数据: <http://www.robots.ox.ac.uk/~vgg/research/affine/>

**Table 1.** Thresholds, number of detected points and calculation time for the detectors in our comparison. (First image of Graffiti scene,  $800 \times 640$ ).

detector	threshold	nb of points	comp. time (msec)
Fast-Hessian	600	1418	120
Hessian-Laplace	1000	1979	650
Harris-Laplace	2500	1664	1800
DoG	default	1520	400



**Table 2.** Computation times for the joint detector - descriptor implementations, tested on the first image of the Graffiti sequence. The thresholds are adapted in order to detect the same number of interest points for all methods. These relative speeds are also representative for other images.

	U-SURF	SURF	SURF-128	SIFT
time (ms):	255	354	391	1036

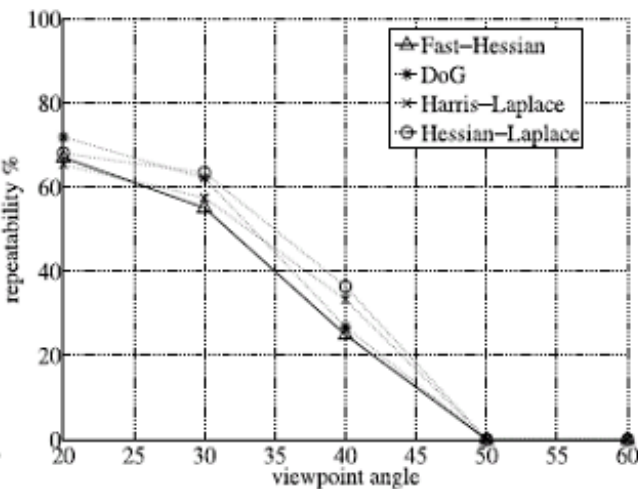
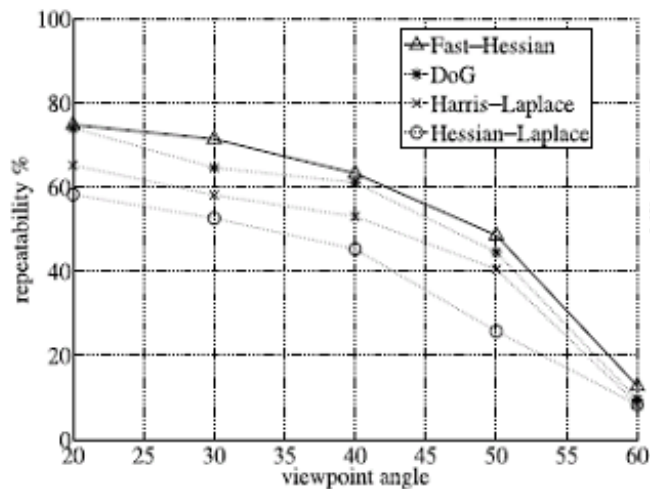
# 实验及实验结果(Detector性能比较)



Walls, viewpoint change



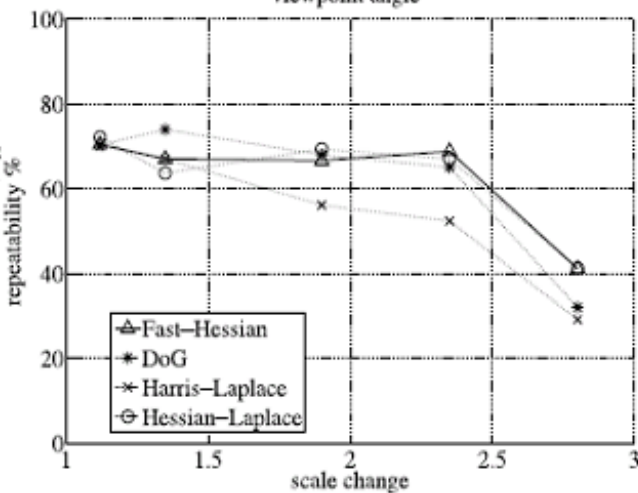
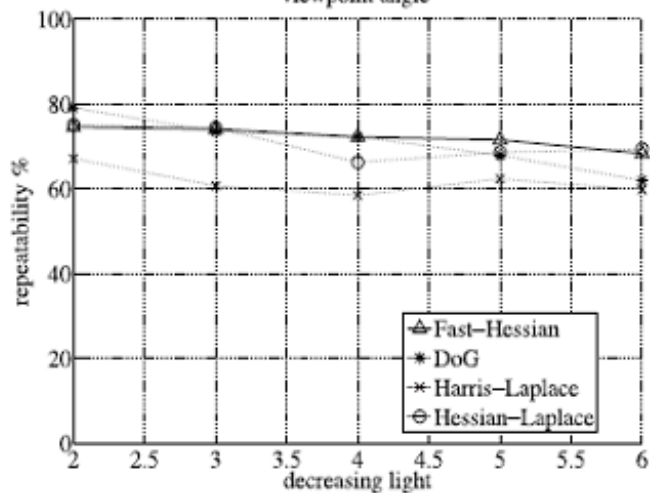
Graffiti, viewpoint change



Leuven, brightness change



Boat, zoom and rotation





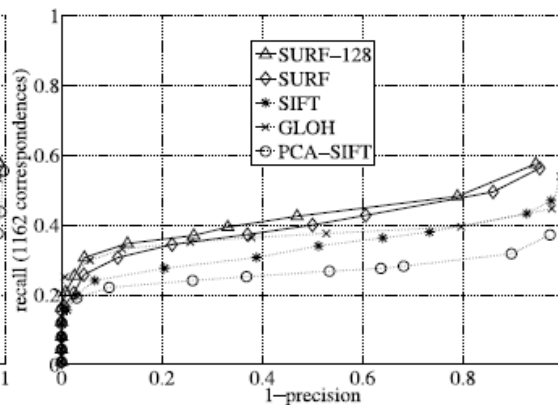
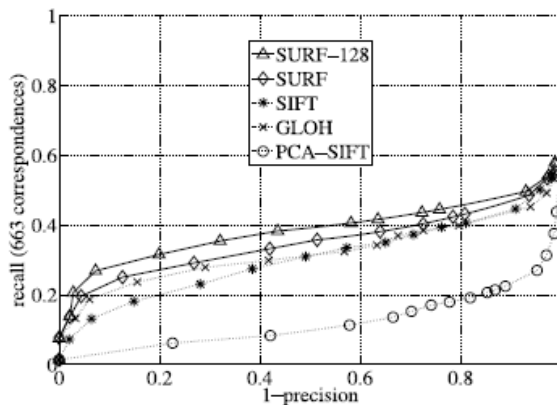
# 实验及实验结果(descriptor性能比较)



Walls, viewpoint change of 50 degrees



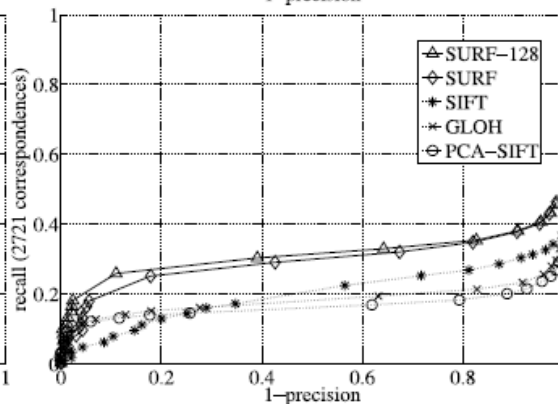
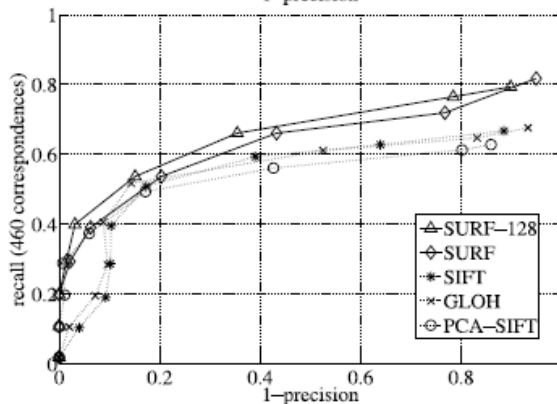
Boat, scale factor 2



Bikes, image blur



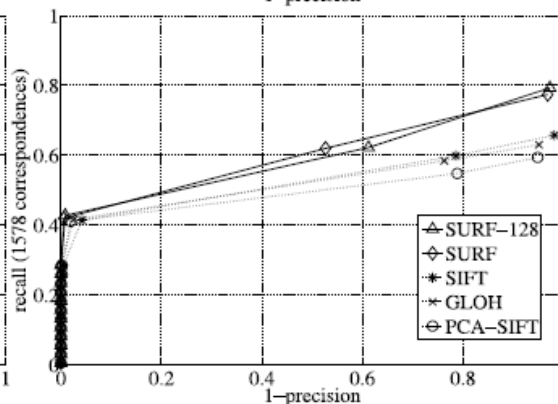
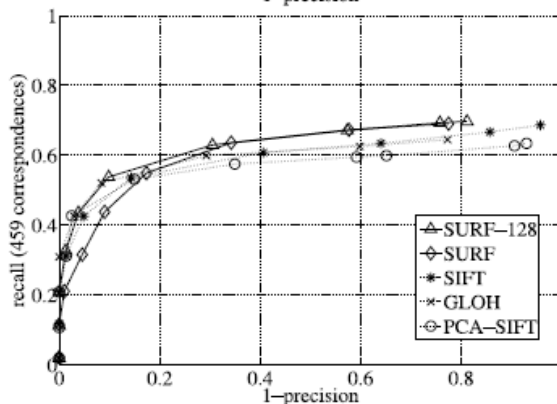
Trees, image blur



Leuven, brightness change



Ubc, JPEG compression



# 结论

- **SURF**速度快，较**SIFT**速度检测和匹配，有**3**倍速度提高，性能和**SIFT**大体相当
- **SURF**在图像模糊和旋转不变上优于**SIFT**
- **SURF**在图像视点变化和光照变化上略差于**SIFT**