计算机视觉

图像特征

中国传媒大学

COMMUNICATION UNIVERSITY OF CHINA

作业 2

2115530096 计算机视觉

作业 2：边缘检测

最后期限：2023 年 11 月 1 日 23:59

（占期末成绩 20%）

此次作业是为了确保学生掌握图像滤波、梯度计算、边缘检测等算法。作业必须独立完成。不要分享你的代码或者使用网上的代码，我们将会使用 MOSS 系统检查抄袭，违反者（不论抄袭还是被抄袭）将会得到 0 分。在操作图像时，确保使用合适的类型转换（即 float32 和 uint8 等）。

请将所有图像、程序打包到“你的姓名_学号_a2.zip”文件，在最后期限前通过邮件发送到 lifang8902@cuc.edu.cn，每迟交 1 天扣 3 分。要求可以调用 a2_script 输出全部结果。

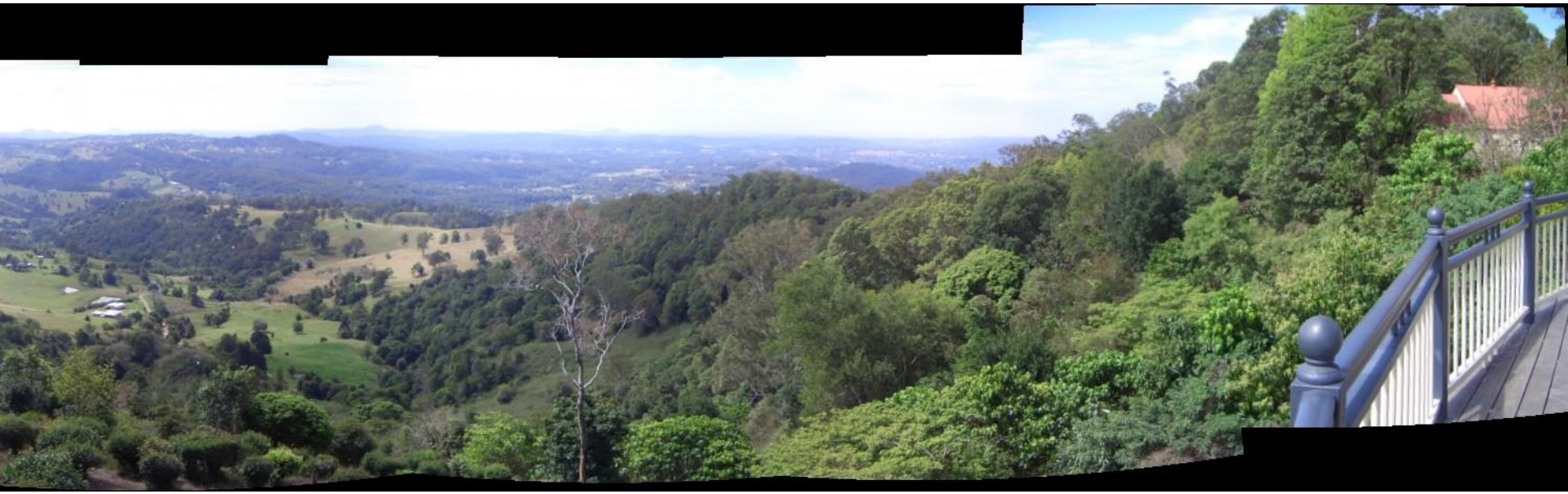
1. 输入（2 分）：
   1) 在 http://sipi.usc.edu/database/database.php?volume=misc 选择一张彩色图像，尺寸不大于512 × 512，下载到 Python 工作目录；
   2) 创建 Python 文件，并命名为“a2_script.py”；
   3) 使用 cv2.imread 读取图像，转换成灰度图并存储在变量 im 中，并使用 cv2.imshow 显示；
2. 计算图像梯度（4 分）：
   1) 使用 sobel 滤波器计算图像 im 的梯度 im_dx、im_dy；
   2) 计算梯度的幅度 grad_mag 和方向 grad_dir；
   3) 将 grad_dir 中的梯度方向量化到 0，$\pm\frac{\pi}{4}$，$\pm\frac{\pi}{2}$，$\pm\frac{3\pi}{4}$，π（即指向周围 8 个像素中的一个）；
   4) 使用 cv2.imshow 分别显示 grad_mag 和 grad_dir；
3. 执行非极大值抑制（6 分）

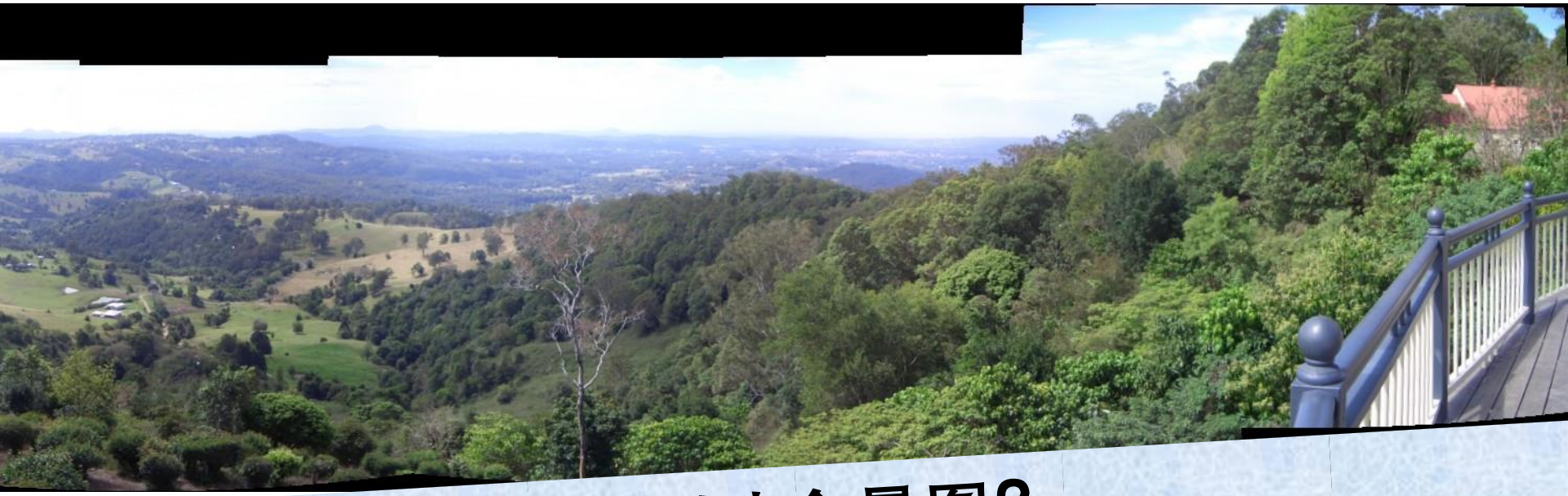
   对梯度幅度 grad_mag 中每个像素，比较它与周围 8 个像素的梯度幅度值，对 grad_mag 执行非极大值抑制，并使用 cv2.imread 显示结果；
4. 阈值化并连接（8 分）
   1) 设置大阈值 thresh_high 和小阈值 thresh_low；
   2) 对 grad_mag 分别使用大小两个阈值，获得二值图像（注意转成 0 或 1），将结果分别存储在 im_thresh_high 和 im_thresh_low；
   3) 令 BW=(im_thresh_high+im_thresh_low)/2，其中值为 1 的像素称为强边缘，值为 0.5 的像素称为弱边缘；
   4) 依次从图像的左上角、右上角、左下角、右下角出发，遍历 BW 中所有的弱边缘像素，如果其相邻 8 个像素中有强边缘，则将该弱边缘改为强边缘；
   5) 将 BW 中的弱边缘置 0，调整合适的大小阈值，并用 cv2.imshow 显示。

上周


上周
边缘可以看作是图像强度表面上的峭壁

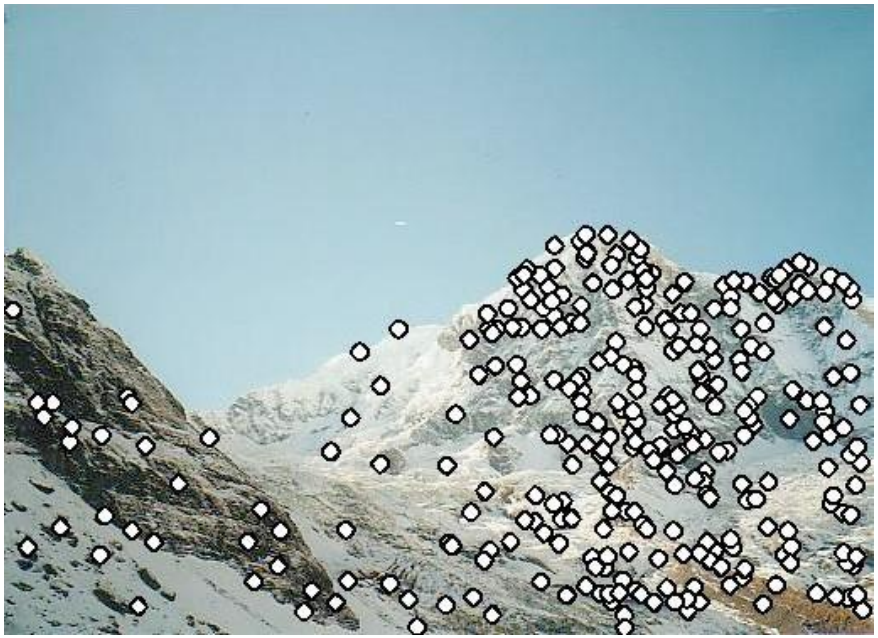
今天

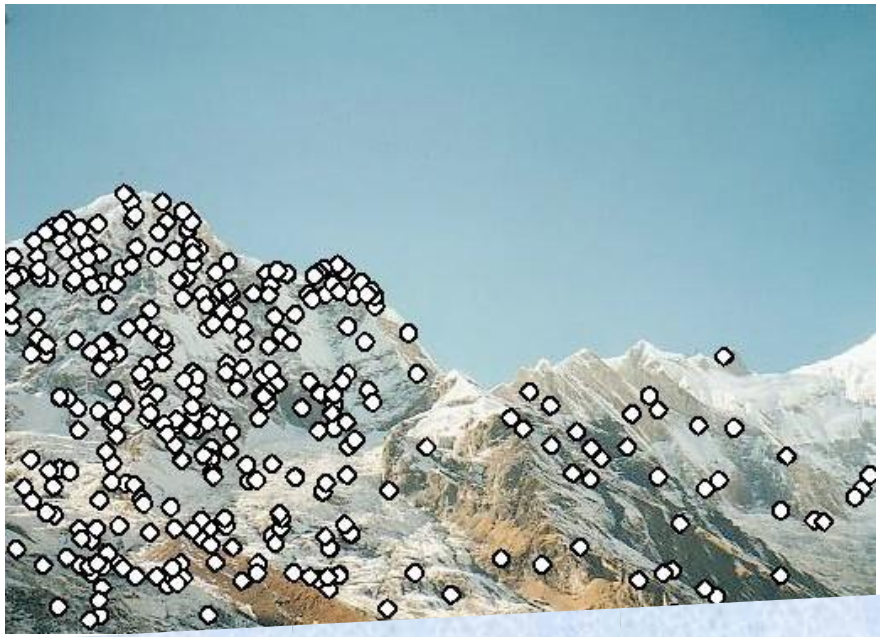




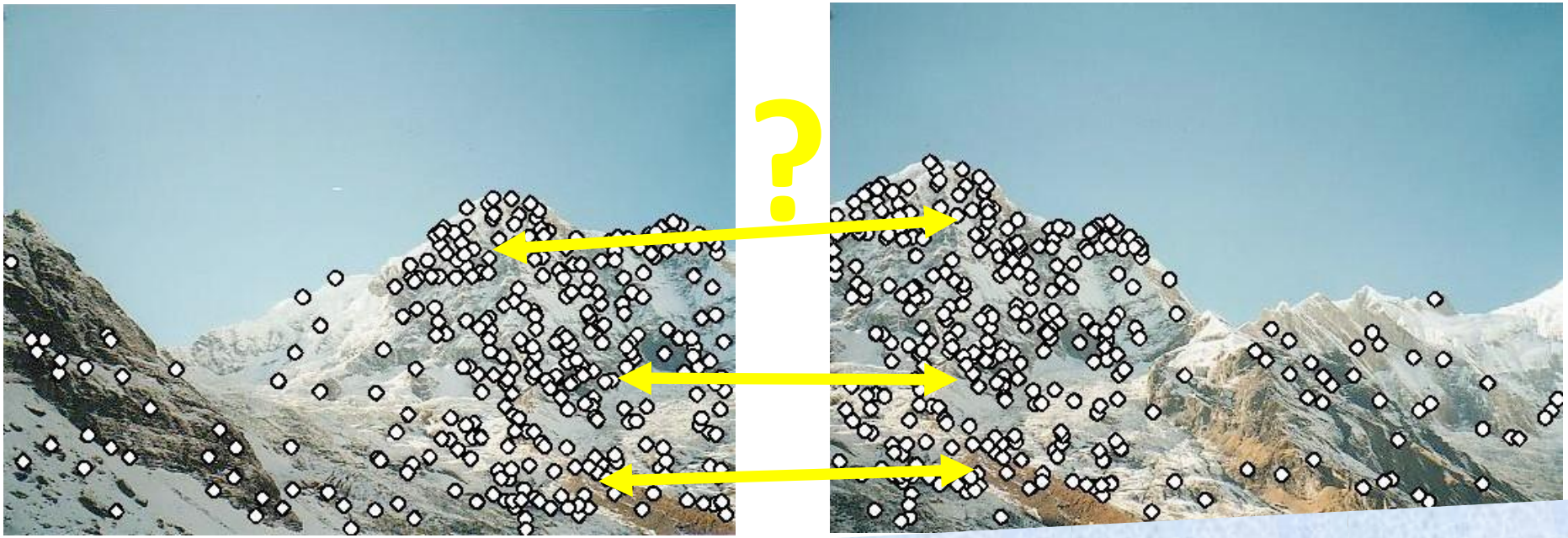
如何创建全景图?

左图中的哪些内容与右图内容匹配?

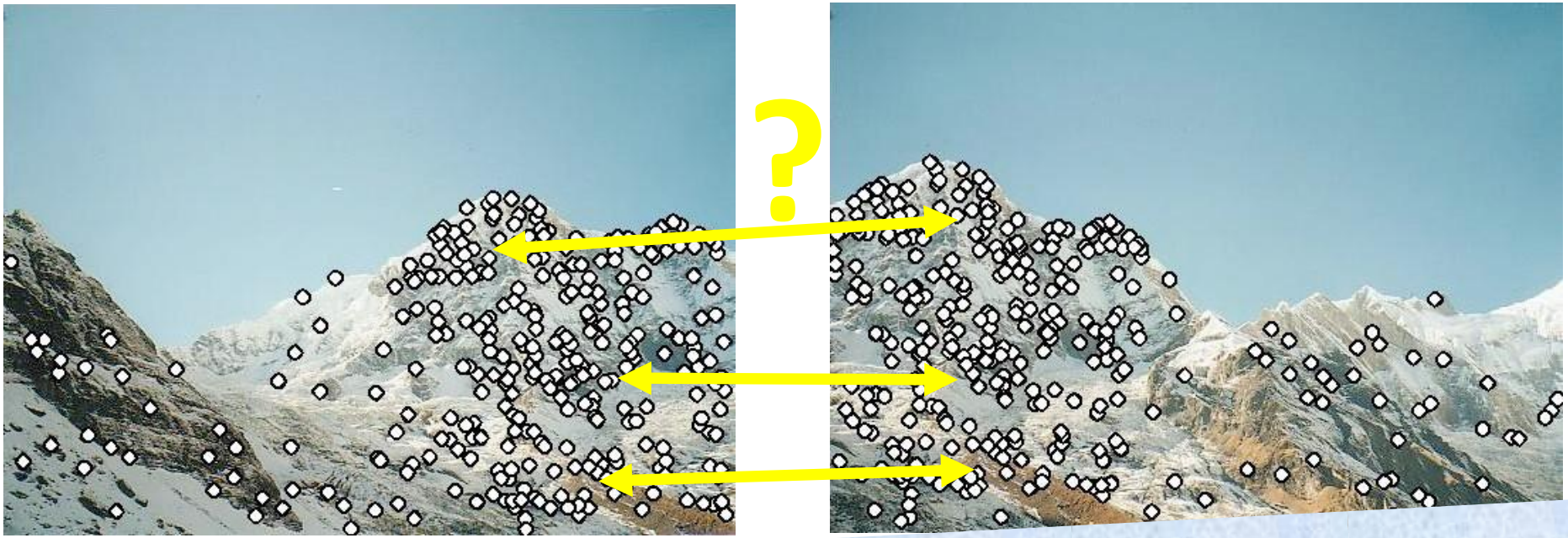




左图中的哪些内容与右图内容匹配?



左图中的哪些内容与右图内容匹配?



左图中的哪些内容与右图内容匹配?

找出匹配对并对齐

Structure-from-Motion Revisited

Johannes L. Schönberger, Jan-Michael Frahm

CVPR 2016

Code available at:
https://github.com/colmap/colmap

“好”特征是怎样炼成的？

可重复性

尽管经过几何/光度映射，依然可以在其他图像中找到相同的特征


输入图像

光度变换

$$I_{\text{new}}(x,y) = \alpha I(x,y) + \beta$$


输入图像

几何变换

可重复性

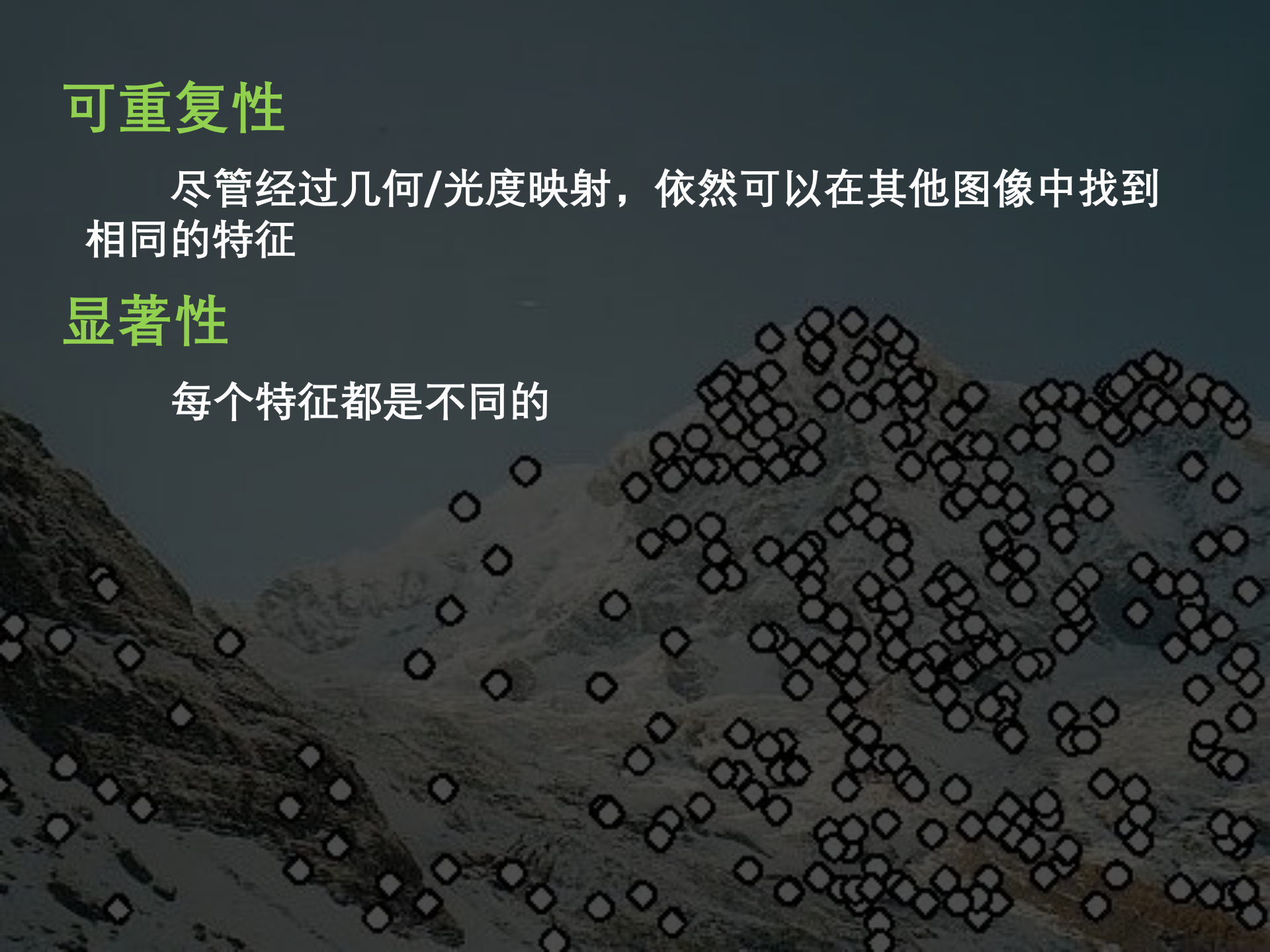
尽管经过几何/光度映射，依然可以在其他图像中找到相同的特征

可重复性

尽管经过几何/光度映射，依然可以在其他图像中找到相同的特征

显著性

每个特征都是不同的



可重复性

尽管经过几何/光度映射，依然可以在其他图像中找到相同的特征

显著性

每个特征都是不同的

紧凑性和效率

特征要比图像像素少很多

可重复性

尽管经过几何/光度映射，依然可以在其他图像中找到相同的特征

显著性

每个特征都是不同的

紧凑性和效率

特征要比图像像素少很多

局部性

特征占据图像的一个相对较小的区域

可重复性

尽管经过几何/光度映射，依然可以在其他图像中找到相同的特征

显著性

每个特征都是不同的

紧凑性和效率

特征要比图像像素少很多

局部性

特征占据图像的一个相对较小的区域

为什么局部性很重要？

Hessian

FAST

Salient Regions

Harris-/Hessian-Affine

Moravec

Harris/Forstner Corners

EBR and IBR

Laplacian

DoG

MSER

Harris-/Hessian-Laplace

Hessian

FAST

Salient Regions

Harris-/Hessian-Affine

Moravec

Harris/Forstner Corners

EBR and IBR

Laplacian

DoG

MSER

Harris-/Hessian-Laplace

Hessian

FAST

Salient Regions

Harris-/Hessian-Affine

Moravec

Harris/Forstner Corners

EBR and IBR

Laplacian

DoG

MSER

Harris-/Hessian-Laplace

Hessian

FAST

Salient Regions

Harris-/Hessian-Affine

Moravec

Harris/Forstner Corners

EBR and IBR

Laplacian **DoG**

MSER

Harris-/Hessian-Laplace

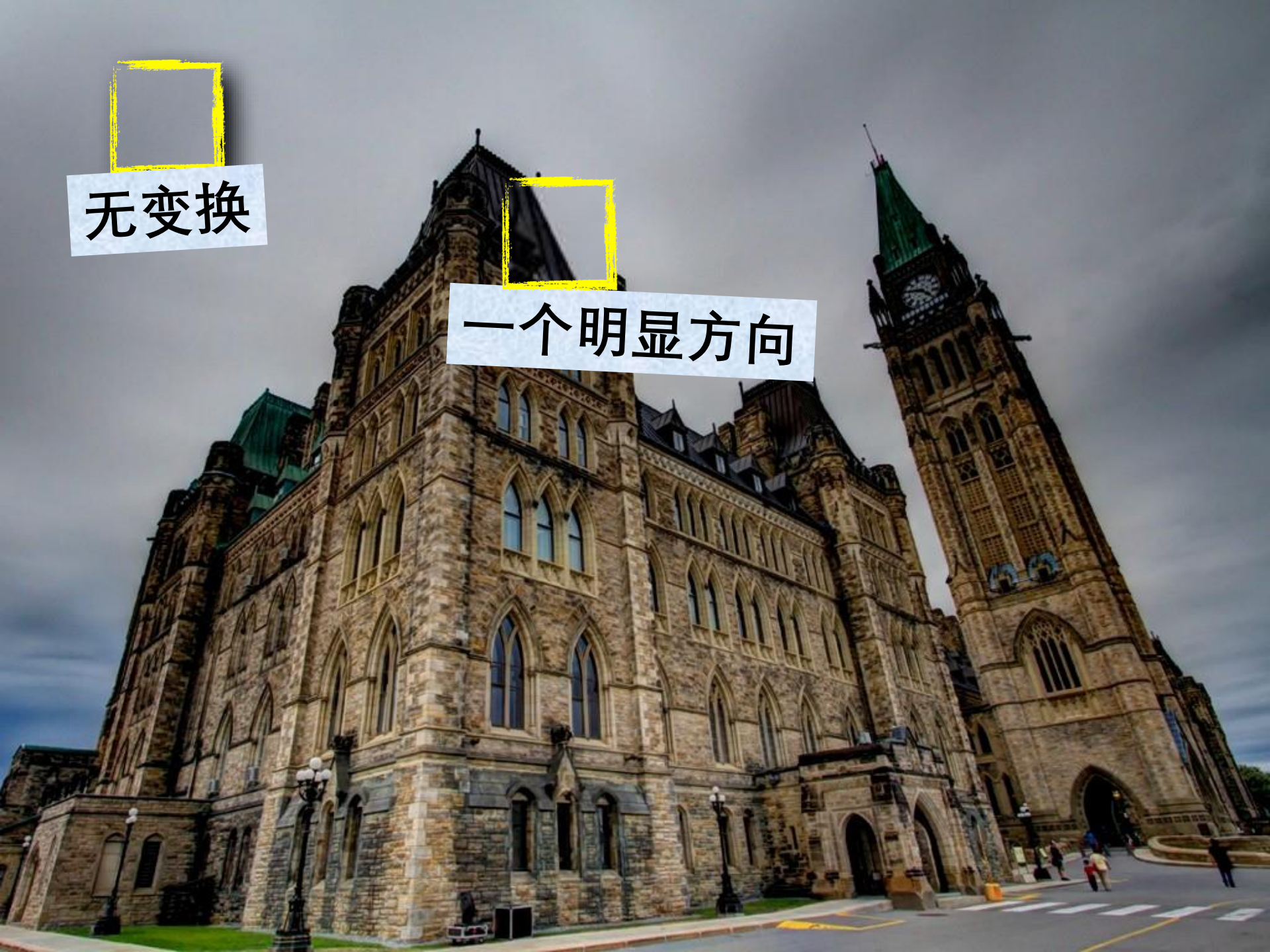
什么是角点？

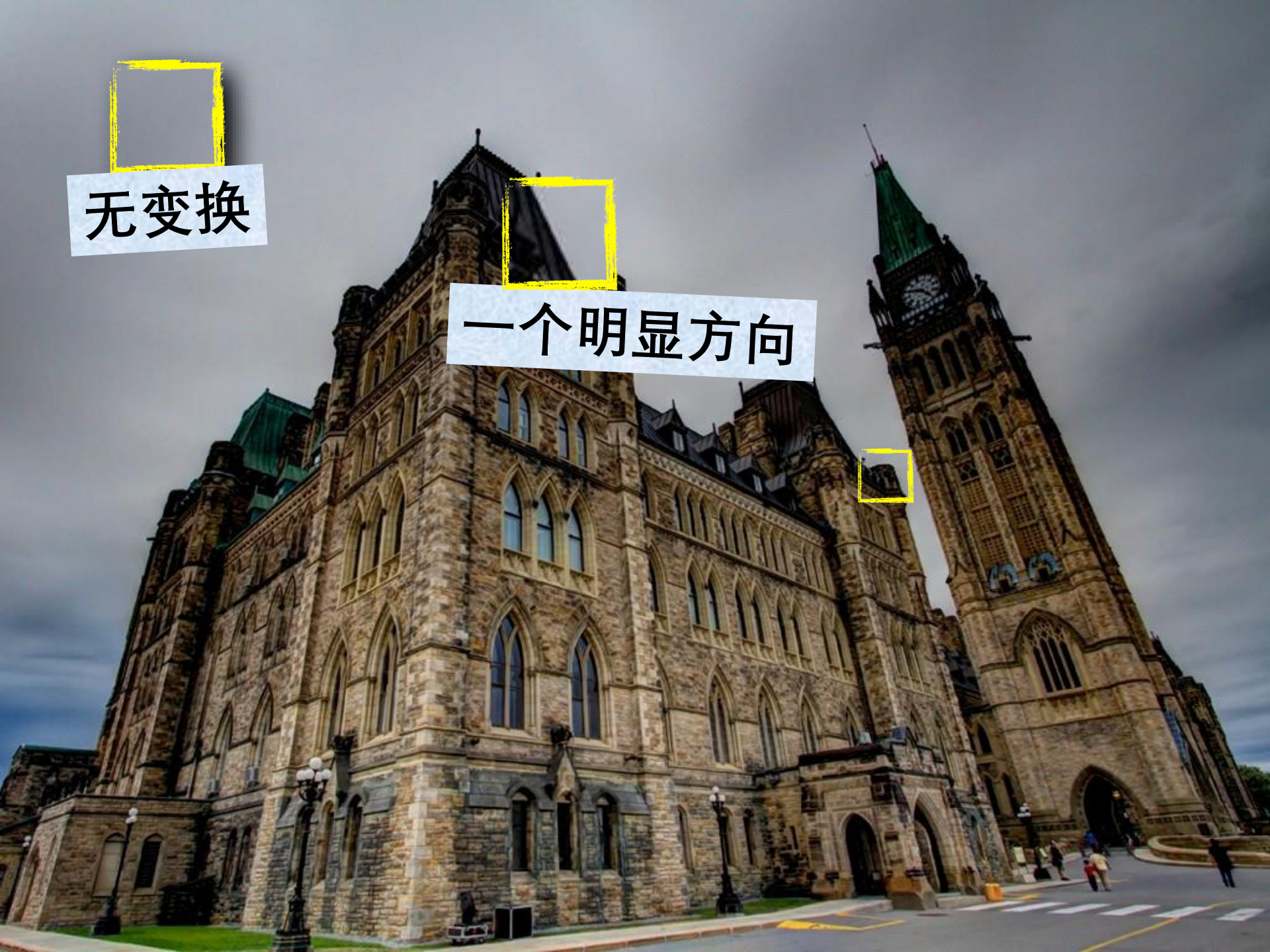
什么是角点？

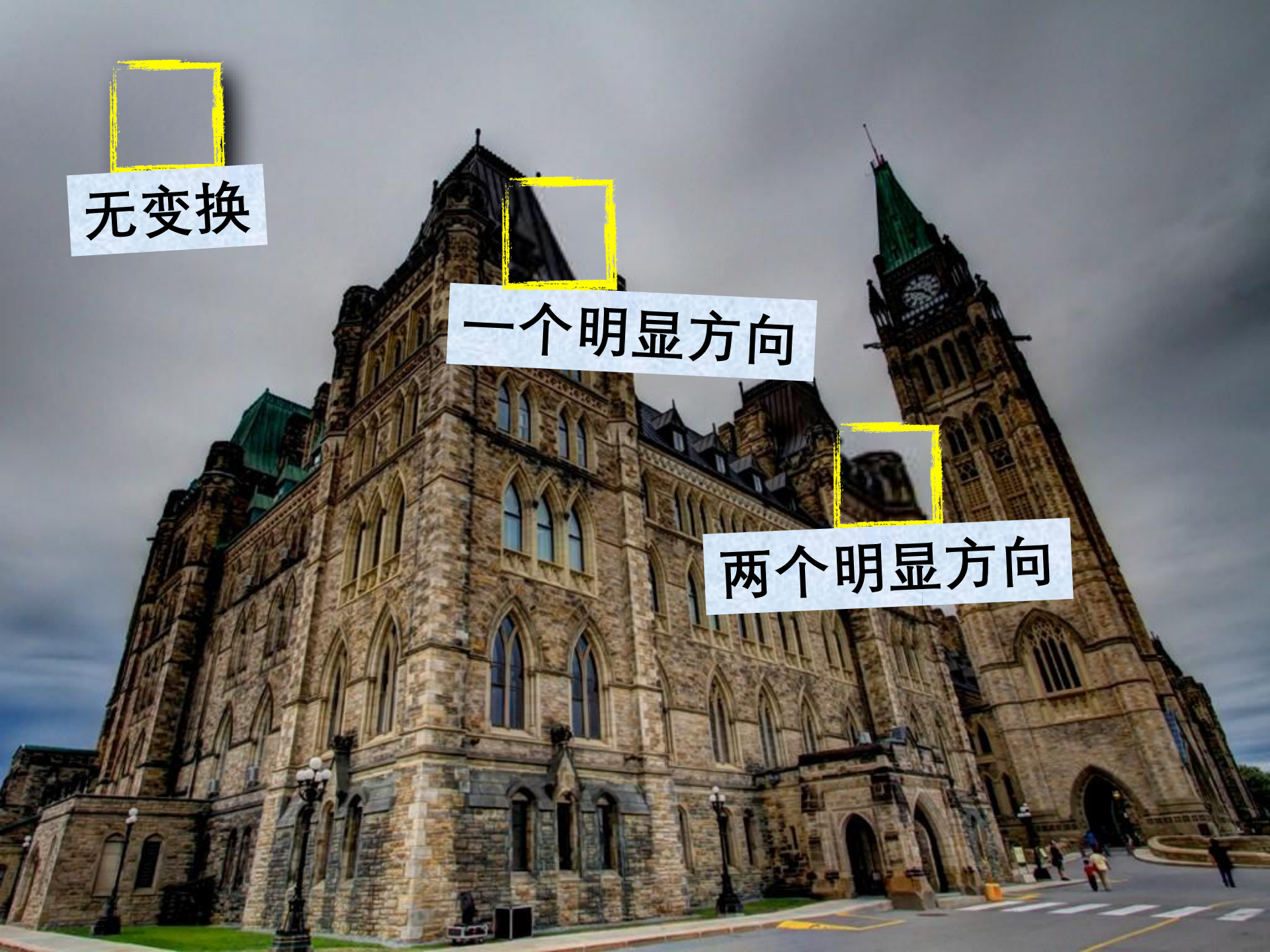
出现两个明显的图像方向结构的位置


无变换


无变换


无变换
一个明显方向


无变换
一个明显方向


无变换
一个明显方向
两个明显方向

为什么要用角点?

为什么要用角点？

限制图案位置的两个自由度

Harris角点

A COMBINED CORNER AND EDGE DETECTOR

Chris Harris & Mike Stephens

Plessey Research Roke Manor, United Kingdom

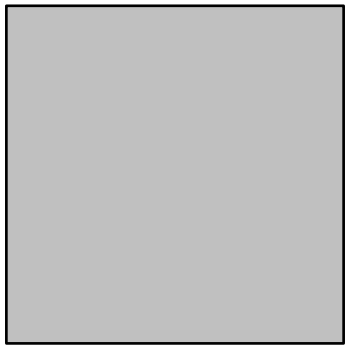
Consistency of image edge filtering is of prime importance for 3D interpretation of image sequences using feature tracking algorithms. To cater for image regions containing texture and isolated features, a combined corner and edge detector based on the local auto-correlation function is utilised, and it is shown to perform with good consistency on natural imagery.

they are discrete, reliable and meaningful[2]. However, the lack of connectivity of feature-points is a major limitation in our obtaining higher level descriptions, such as surfaces and objects. We need the richer information that is available from edges[3].

Alvey Vision Conference 1988

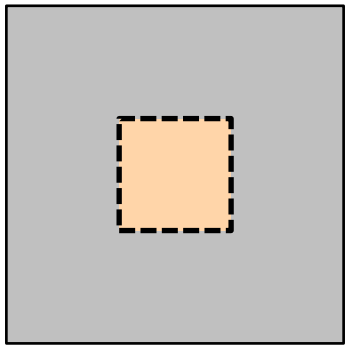
直觉

分析信号的局部变化



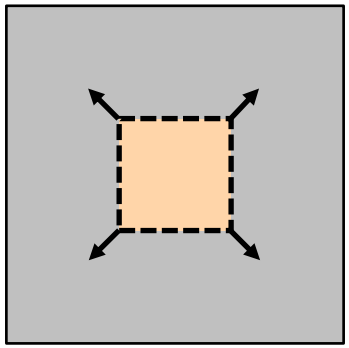
直觉

分析信号的局部变化



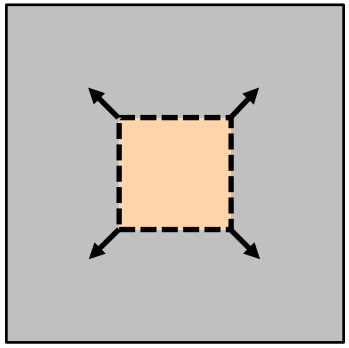
直觉

分析信号的局部变化



直觉

分析信号的局部变化



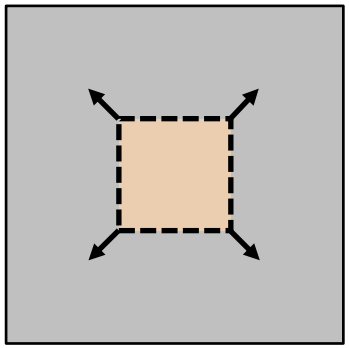
“平坦”区域

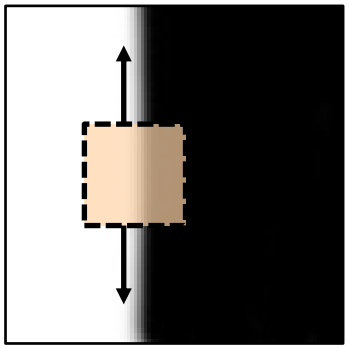
各方向

无变化

直觉

分析信号的局部变化





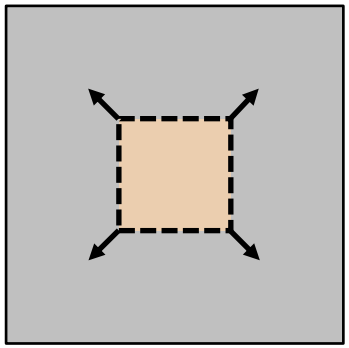
“平坦”区域

各方向

无变化

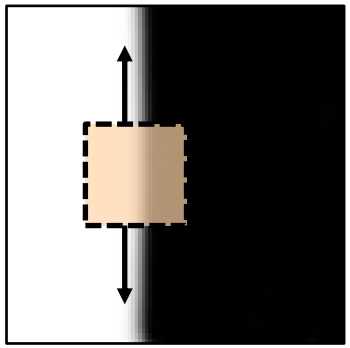
直觉

分析信号的局部变化



“平坦”区域

各方向
无变化

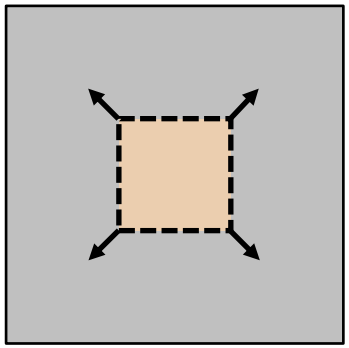


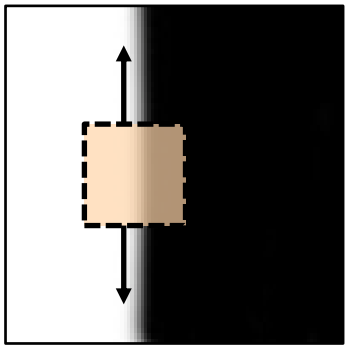
“边缘”

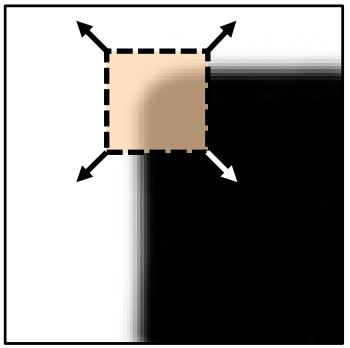
沿边缘方向
无变化

直觉

分析信号的局部变化





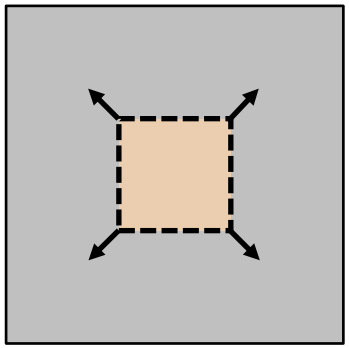


“平坦”区域
各方向
无变化

“边缘”
沿边缘方向
无变化

直觉

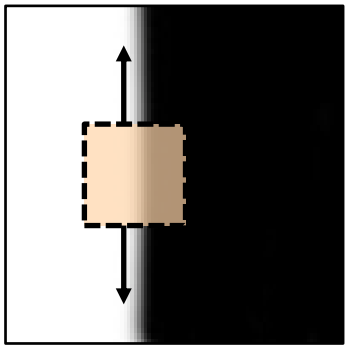
分析信号的局部变化



“平坦”区域

各方向

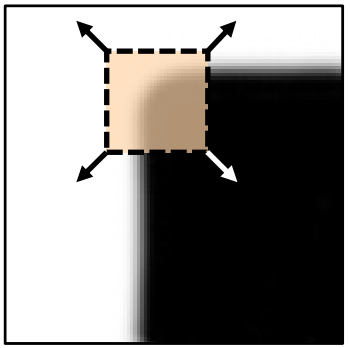
无变化



“边缘”

沿边缘方向

无变化



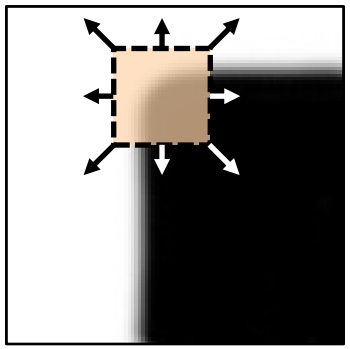
“角点”

各方向

显著变化

推导

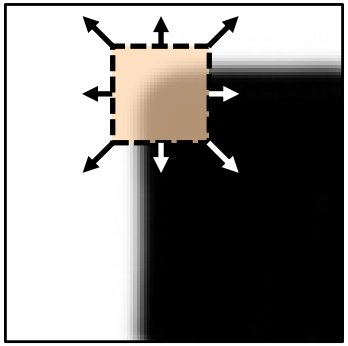
移位$(\Delta x, \Delta y)$后的强度变化



推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

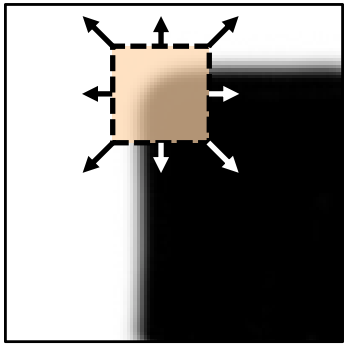


推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

图像

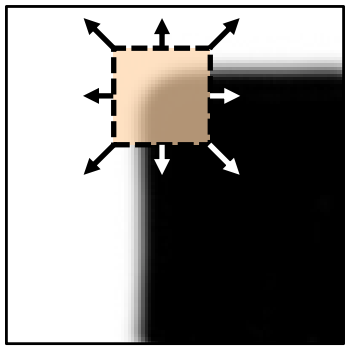


推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

移位图像

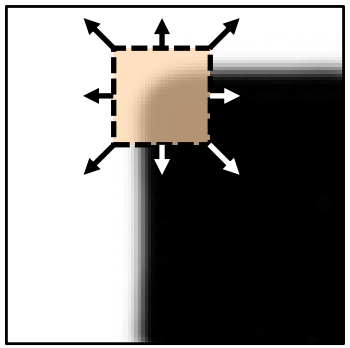


推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

窗函数

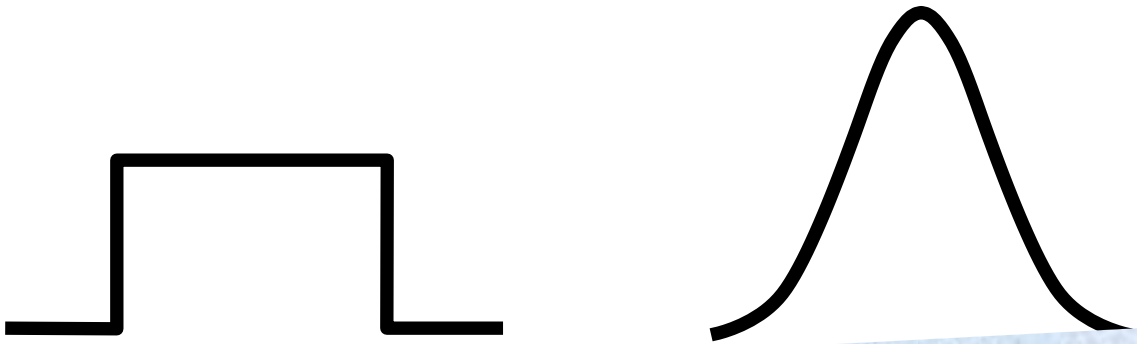


推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

窗函数

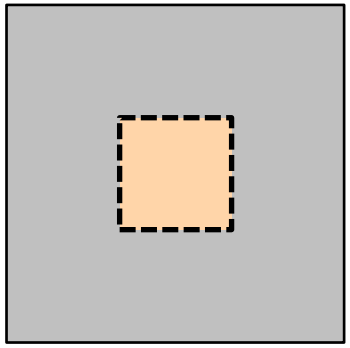


Alvey Vision Conference 1988

推导

移位$(\Delta x, \Delta y)$后的强度变化

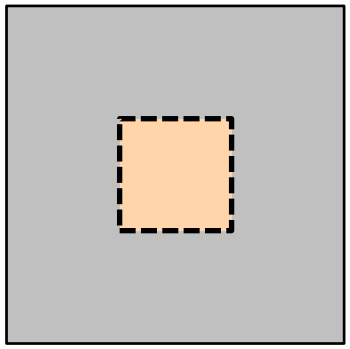
$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$



推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

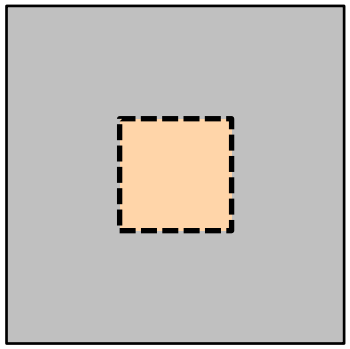


响应输出什么？

推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$



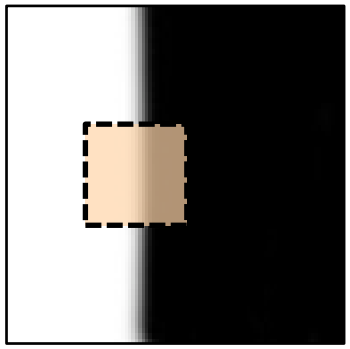
几乎不变的块

$$E(\Delta x, \Delta y) \approx 0$$

推导

移位$(\Delta x, \Delta y)$后的强度变化

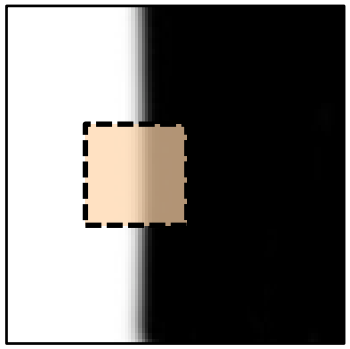
$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$



推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

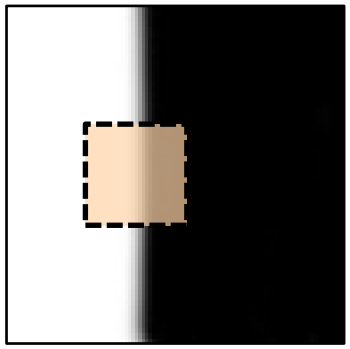


响应输出什么？

推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$



“边缘”块

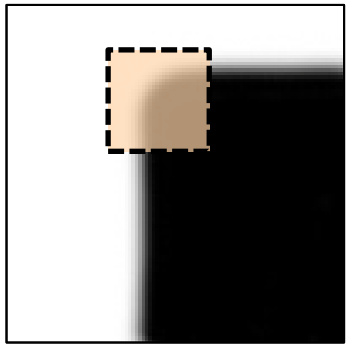
$E(\Delta x, \Delta y)$

沿着一个方向变化大

推导

移位$(\Delta x, \Delta y)$后的强度变化

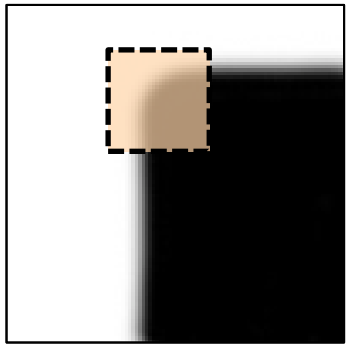
$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$



推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

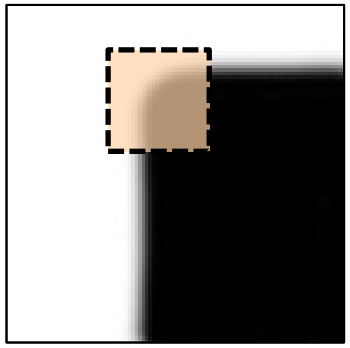


响应输出什么？

推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$



“角点”块

$E(\Delta x, \Delta y)$

沿着两个方向变化大

推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

回顾：泰勒级数

1D情况

$$f(x+u) = f(x) + f'(x)u + \frac{1}{2}f''(x)u^2 + \text{h. o. t.}$$

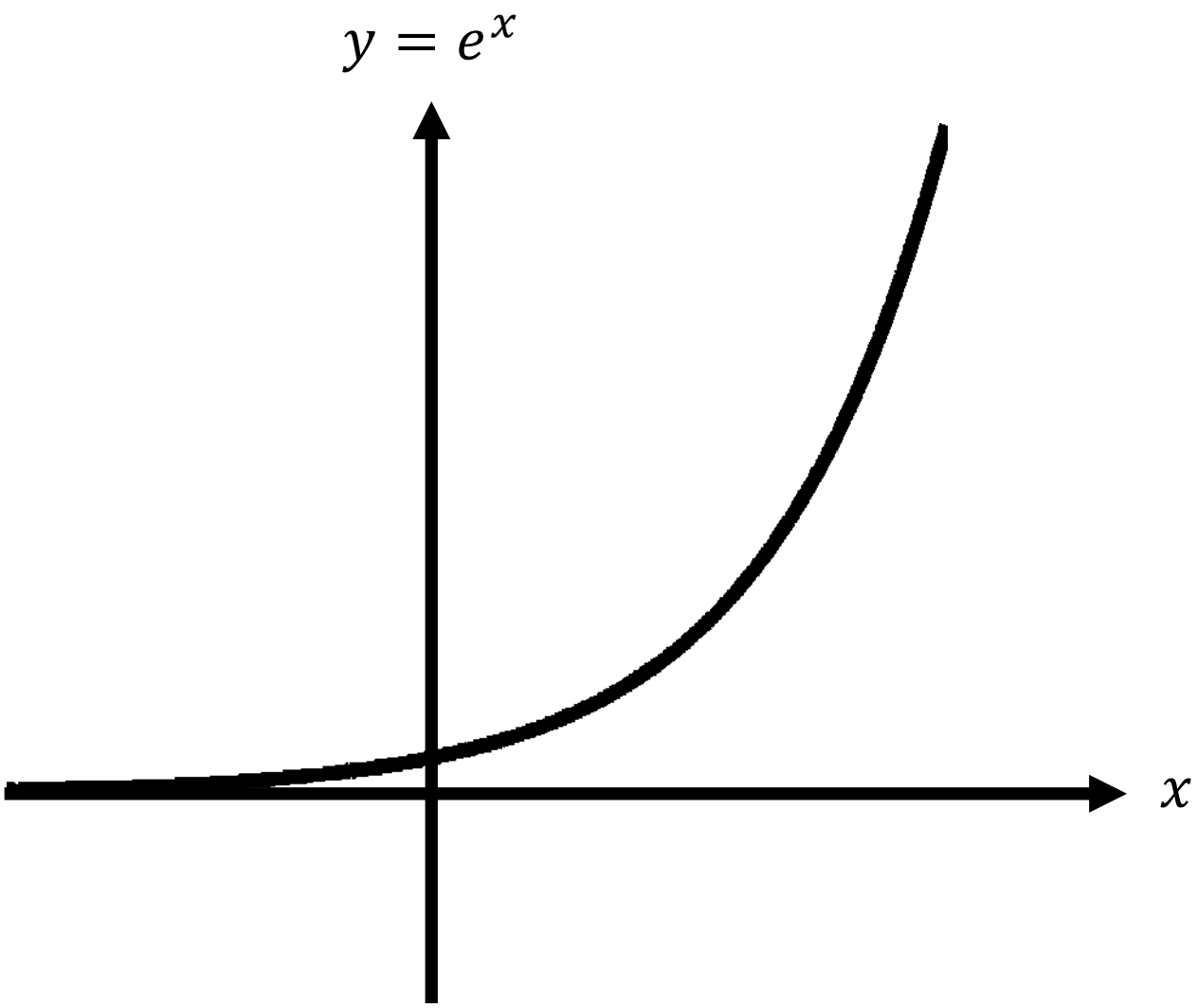
1D情况

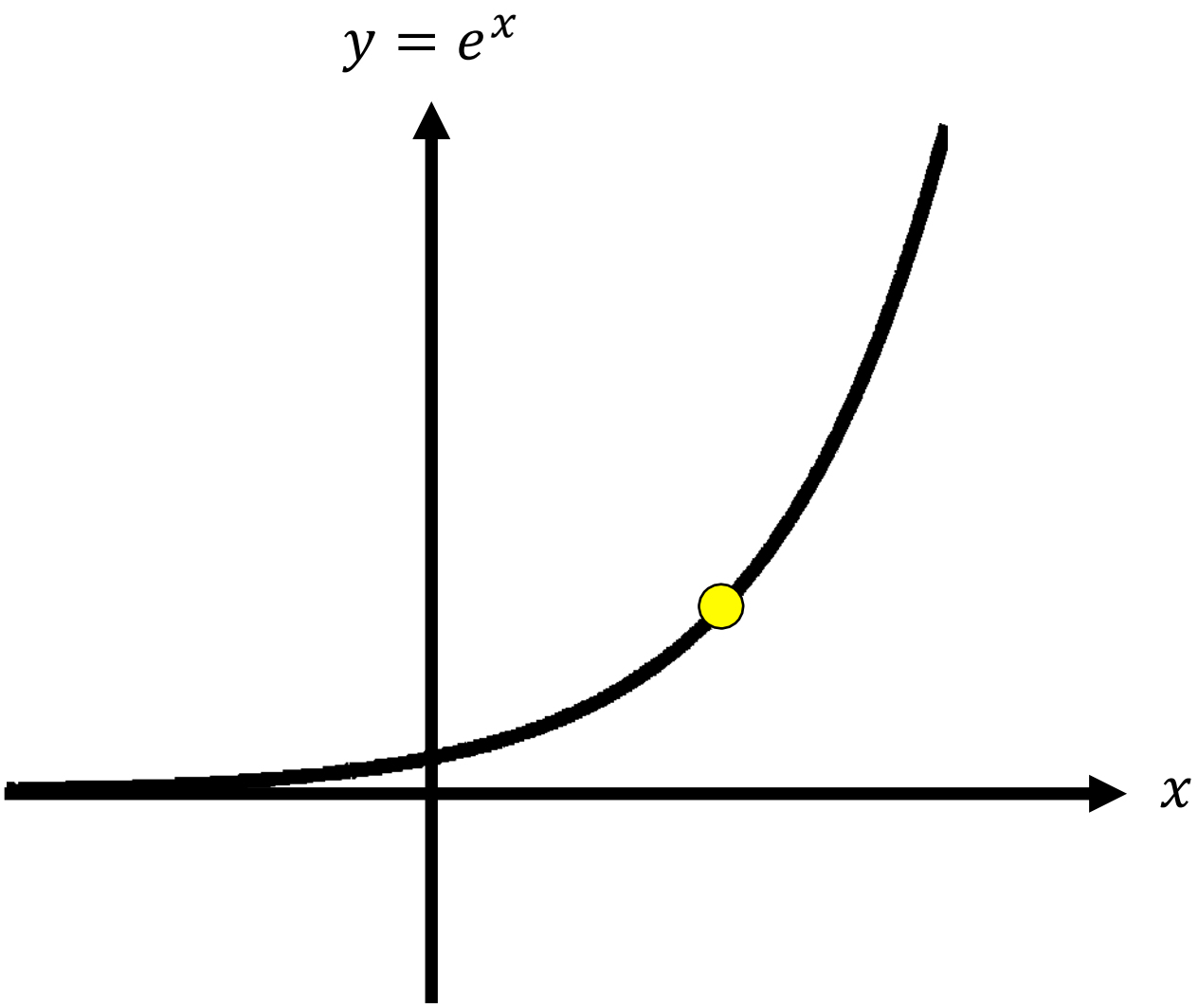
$$f(x+u) \approx f(x) + f'(x)u + \frac{1}{2}f''(x)u^2 + \text{h. o. t.}$$

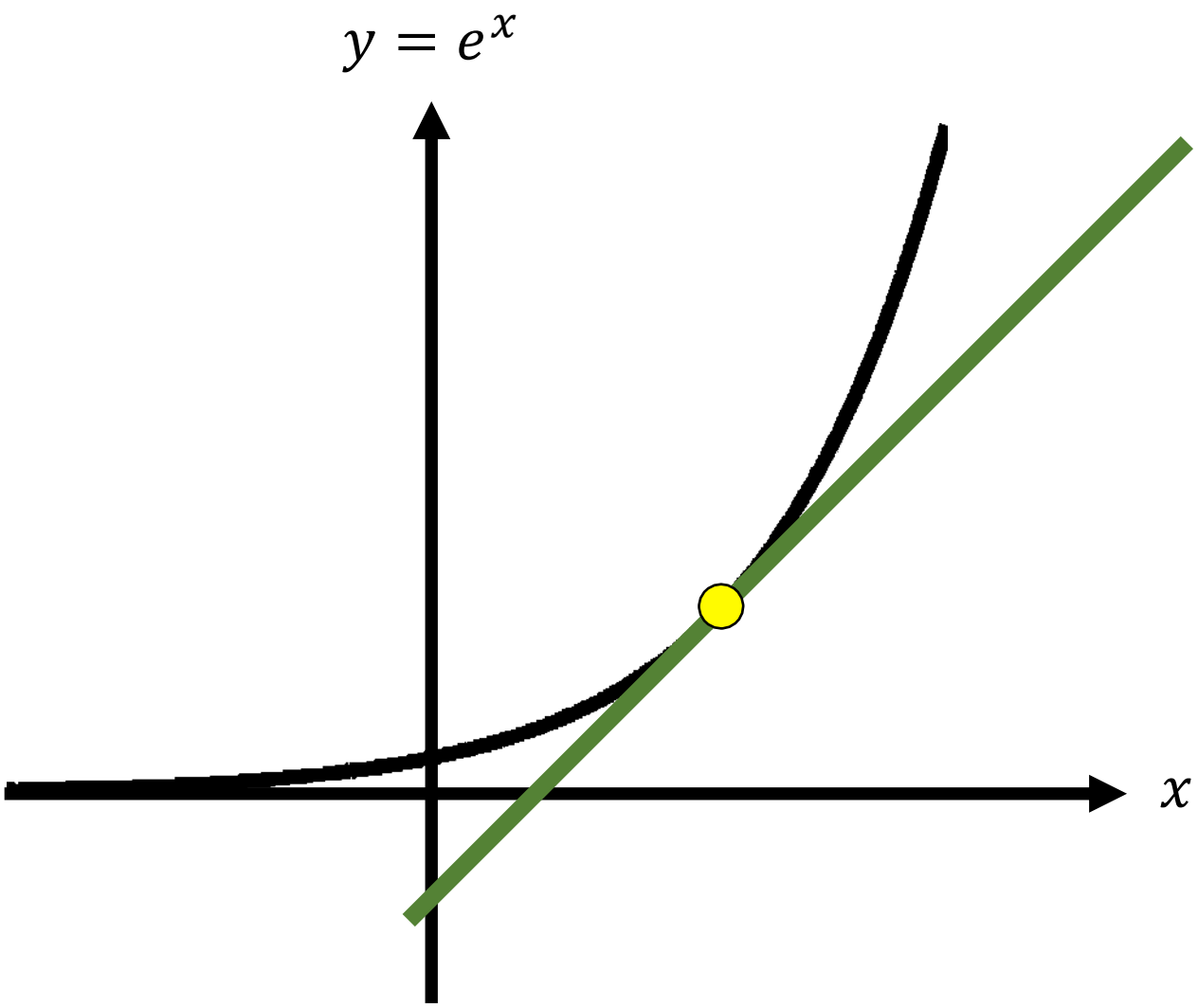
1D情况

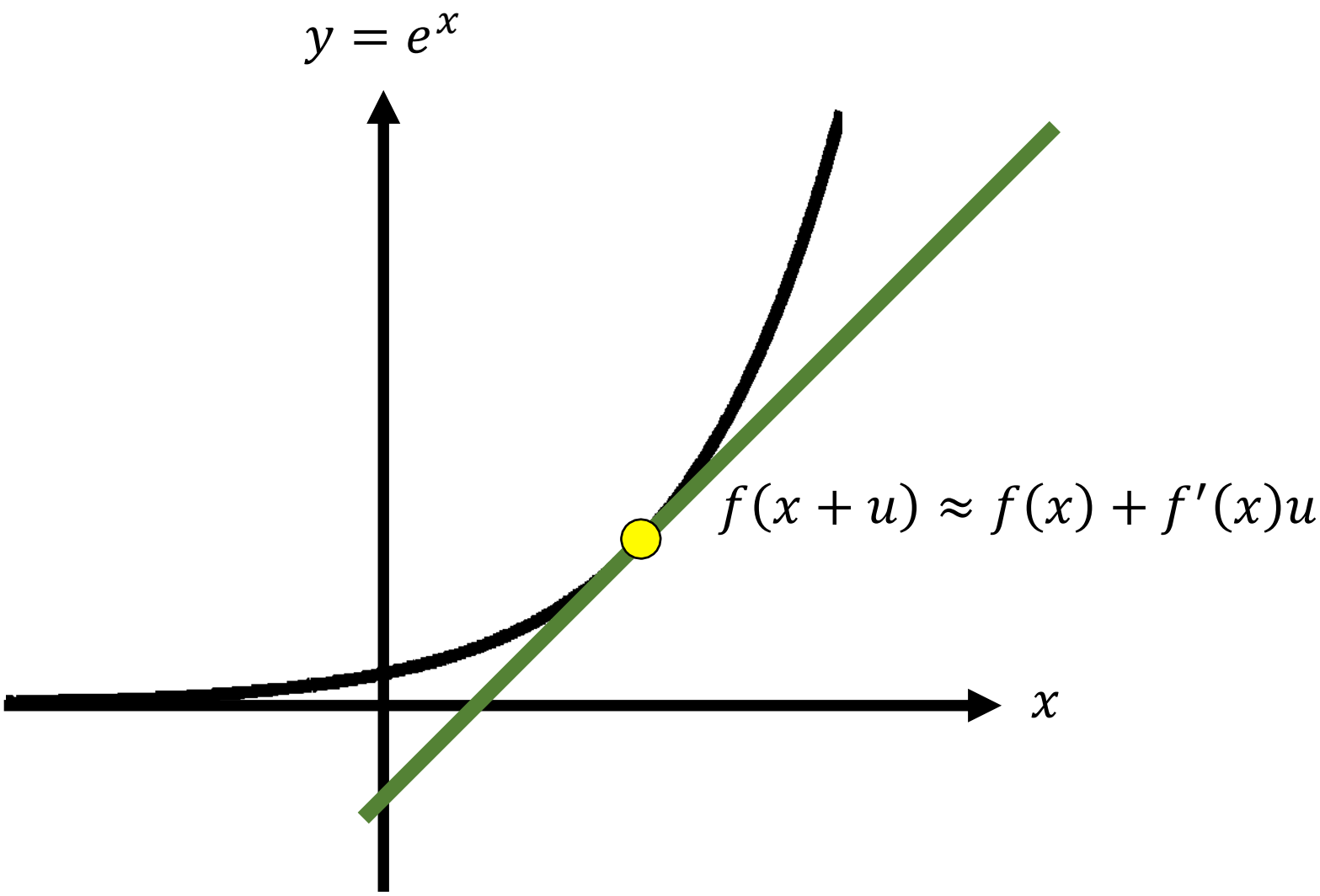
$$f(x+u) \approx f(x) + f'(x)u + \frac{1}{2}f''(x)u^2 + \text{h.o.t.}$$

一阶近似


$y = e^x$
x


$y = e^x$
x


$y = e^x$
x


$y = e^x$
$f(x+u) \approx f(x) + f'(x)u$
x

2D情况

$$f(x+u,y+v)=f(x,y)+f_x(x,y)u+f_y(x,y)v$$
$$+\frac{1}{2}\left[f_{xx}(x,y)u^2+f_{xy}(x,y)uv+f_{yy}(x,y)v^2\right]$$
$$+\text{h. o. t.}$$

2D情况

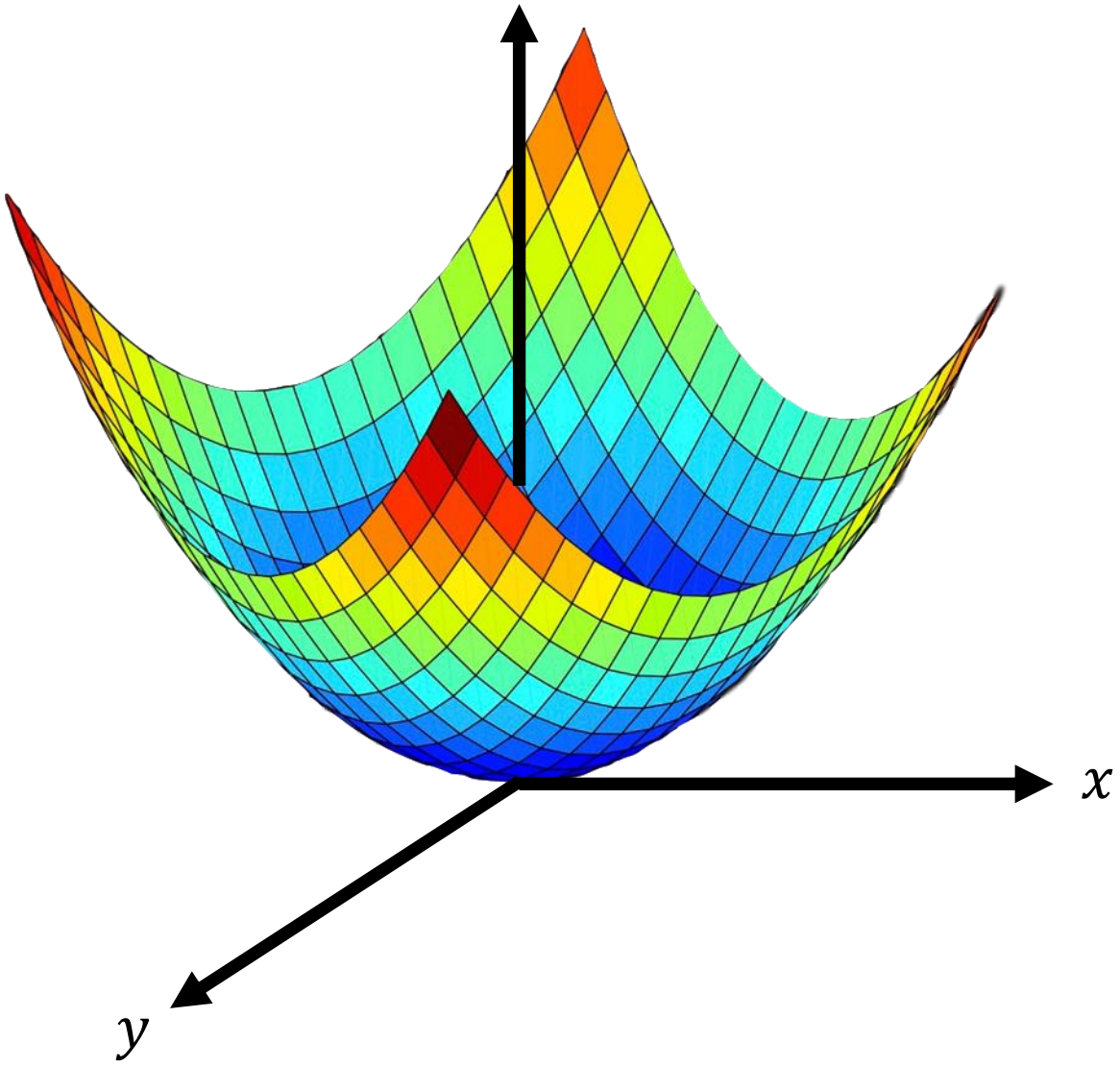
$$f(x+u, y+v) \approx f(x,y) + f_x(x,y)u + f_y(x,y)v$$
$$+\frac{1}{2}\left[f_{xx}(x,y)u^2 + f_{xy}(x,y)uv + f_{yy}(x,y)v^2\right]$$
$$+\text{h. o. t.}$$

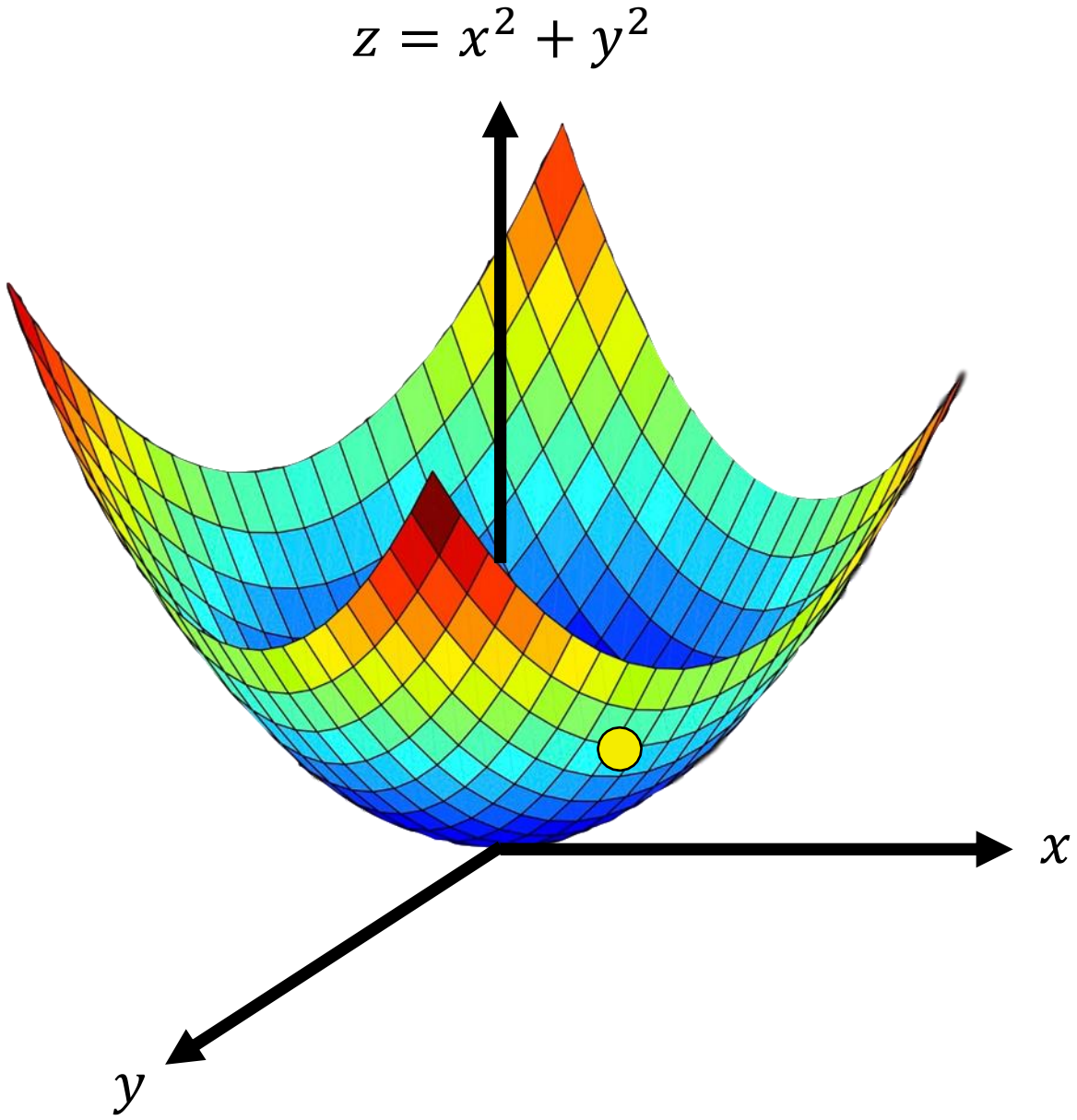
2D情况

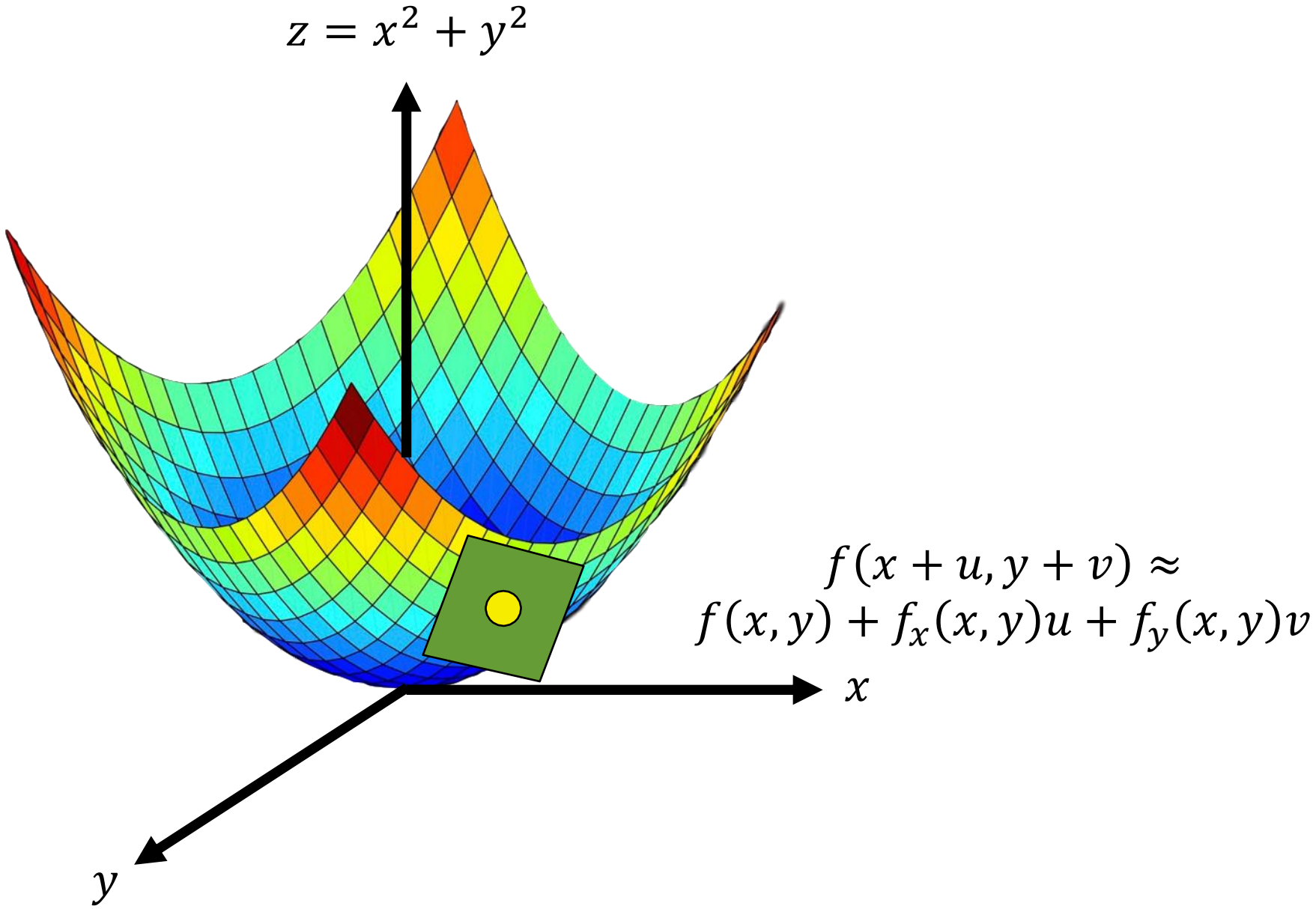
$$f(x+u,y+v) \approx f(x,y) + f_x(x,y)u + f_y(x,y)v$$
$$+\frac{1}{2}\left[f_{xx}(x,y)u^2 + f_{xy}(x,y)uv + f_{yy}(x,y)v^2\right]$$
$$+\text{h. o. t.}$$

一阶近似

$z = x^2 + y^2$




$z = x^2 + y^2$
x
y


$z = x^2 + y^2$
$f(x+u, y+v) \approx$
$f(x,y) + f_x(x,y)u + f_y(x,y)v$
x
y

回顾：泰勒级数

推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

如何展开这一项?

推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

泰勒级数展开

推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

泰勒级数展开

$$\approx \sum_{x,y} w(x,y)\left[I(x,y) - I(x,y) - I_x(x,y)\Delta x - I_y(x,y)\Delta y\right]^2$$

推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

泰勒级数展开

$$\approx \sum_{x,y} w(x,y)\left[I(x,y) - I(x,y) - I_x(x,y)\Delta x - I_y(x,y)\Delta y\right]^2$$

推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

泰勒级数展开

$$\approx \sum_{x,y} w(x,y)\left[I(x,y) - I(x,y) - I_x(x,y)\Delta x - I_y(x,y)\Delta y\right]^2$$

展开

推导

移位$(\Delta x, \Delta y)$后的强度变化

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

泰勒级数展开

$$\approx \sum_{x,y} w(x,y)\left[I(x,y) - I(x,y) - I_x(x,y)\Delta x - I_y(x,y)\Delta y\right]^2$$

展开

$$= \sum_{x,y} w(x,y)\left[I_x(x,y)^2\Delta x^2 + 2I_x(x,y)I_y(x,y)\Delta x\Delta y + I_y(x,y)^2\Delta y^2\right]$$

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

泰勒级数展开

$$\approx \sum_{x,y} w(x,y)\left[I(x,y) - I(x,y) - I_x(x,y)\Delta x - I_y(x,y)\Delta y\right]^2$$

展开

$$= \sum_{x,y} w(x,y)\left[I_x(x,y)^2\Delta x^2 + 2I_x(x,y)I_y(x,y)\Delta x\Delta y + I_y(x,y)^2\Delta y^2\right]$$

改写成矩阵

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

泰勒级数展开

$$\approx \sum_{x,y} w(x,y)\left[I(x,y) - I(x,y) - I_x(x,y)\Delta x - I_y(x,y)\Delta y\right]^2$$

展开

$$= \sum_{x,y} w(x,y)\left[I_x(x,y)^2\Delta x^2 + 2I_x(x,y)I_y(x,y)\Delta x\Delta y + I_y(x,y)^2\Delta y^2\right]$$

改写成矩阵

$$E(\Delta x, \Delta y) = (\Delta x \quad \Delta y)\mathbf{M}\begin{pmatrix}\Delta x \\ \Delta y\end{pmatrix}$$

其中

$$\mathbf{M} = \sum_{x,y} w(x,y)\begin{pmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{pmatrix}$$

$$E(\Delta x, \Delta y) = (\Delta x \quad \Delta y)\mathbf{M}\begin{pmatrix}\Delta x \\ \Delta y\end{pmatrix}$$

其中

$$\mathbf{M} = \sum_{x,y} w(x,y)\begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

回顾：线性代数

定义：一个$d \times d$矩阵$\mathbf{M}$有特征值λ，如果存在一个d维向量$\mathbf{u} \neq \mathbf{0}$，使得

定义：一个$d \times d$矩阵$\mathbf{M}$有特征值λ，如果存在一个d维向量$\mathbf{u} \neq \mathbf{0}$，使得

$$\mathbf{Mu} = \lambda\mathbf{u}$$

定义：一个$d \times d$矩阵$\mathbf{M}$有特征值λ，如果存在一个d维向量$\mathbf{u} \neq \mathbf{0}$，使得

$$\mathbf{M}\mathbf{u} = \lambda\mathbf{u}$$

这个$\mathbf{u}$是对应于特征值λ的特征向量。

定义：一个矩阵$\mathbf{B}$是对称的，如果$\mathbf{B} = \mathbf{B}^{\mathrm{T}}$。

定义：一个矩阵$\mathbf{B}$是对称的，如果$\mathbf{B} = \mathbf{B}^{\mathrm{T}}$。

例如，$\begin{pmatrix} 1 & 7 & 3 \\ 7 & 4 & -5 \\ 3 & -5 & 6 \end{pmatrix}$

谱分解

定理：令$\mathbf{M}$是一个$d \times d$实对称矩阵，它具有特征值$\lambda_1, \cdots, \lambda_d$，及对应的正交特征向量$\mathbf{u}_1, \cdots, \mathbf{u}_d$。那么：

谱分解

定理：令$\mathbf{M}$是一个$d \times d$实对称矩阵，它具有特征值$\lambda_1, \cdots, \lambda_d$，及对应的正交特征向量$\mathbf{u}_1, \cdots, \mathbf{u}_d$。那么：

$$\mathbf{M} = (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_d) \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_d \end{pmatrix} \begin{pmatrix} \mathbf{u}_1^{\mathrm{T}} \\ \mathbf{u}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{u}_d^{\mathrm{T}} \end{pmatrix}$$

谱分解

定理：令$\mathbf{M}$是一个$d \times d$实对称矩阵，它具有特征值$\lambda_1, \cdots, \lambda_d$，及对应的正交特征向量$\mathbf{u}_1, \cdots, \mathbf{u}_d$。那么：

$$\mathbf{M} = \underbrace{\begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_d \end{pmatrix}}_{\mathbf{R}} \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_d \end{pmatrix} \begin{pmatrix} \mathbf{u}_1^{\mathrm{T}} \\ \mathbf{u}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{u}_d^{\mathrm{T}} \end{pmatrix}$$

谱分解

定理：令$\mathbf{M}$是一个$d \times d$实对称矩阵，它具有特征值$\lambda_1, \cdots, \lambda_d$，及对应的正交特征向量$\mathbf{u}_1, \cdots, \mathbf{u}_d$。那么：

$$\mathbf{M} = \underbrace{\begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_d \end{pmatrix}}_{\mathbf{R}} \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_d \end{pmatrix} \underbrace{\begin{pmatrix} \mathbf{u}_1^{\mathrm{T}} \\ \mathbf{u}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{u}_d^{\mathrm{T}} \end{pmatrix}}_{\mathbf{R}^{\mathrm{T}}}$$

定义：一个$d \times d$实对称矩阵$\mathbf{M}$是半正定的，如果

定义：一个$d \times d$实对称矩阵$\mathbf{M}$是半正定的，如果

对所有$\mathbf{z} \in \mathbb{R}^d$，有$\mathbf{z}^{\mathrm{T}}\mathbf{M}\mathbf{z} \geq 0$

定义：一个$d \times d$实对称矩阵$\mathbf{M}$是半正定的，如果

对所有$\mathbf{z} \in \mathbb{R}^d$，有$\mathbf{z}^{\mathrm{T}}\mathbf{M}\mathbf{z} \geq 0$

$\mathbf{M}$的所有特征值都大于等于0

回顾：线性代数

$$E(\Delta x, \Delta y) = (\Delta x \quad \Delta y)\mathbf{M}\begin{pmatrix}\Delta x \\ \Delta y\end{pmatrix}$$

其中

$$\mathbf{M} = \sum_{x,y} w(x,y)\begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

$$E(\Delta x, \Delta y) = (\Delta x \quad \Delta y)\mathbf{M}\begin{pmatrix}\Delta x \\ \Delta y\end{pmatrix}$$

其中

$$\mathbf{M} = \sum_{x,y} w(x,y)\begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

M是一个对称矩阵

$$E(\Delta x, \Delta y) = (\Delta x \quad \Delta y)\mathbf{M}\begin{pmatrix}\Delta x \\ \Delta y\end{pmatrix}$$

其中

$$\mathbf{M} = \sum_{x,y} w(x,y)\begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

M是一个对称矩阵

$$\mathbf{M} = \mathbf{R}\begin{pmatrix}\lambda_1 & 0 \\ 0 & \lambda_2\end{pmatrix}\mathbf{R}^{\mathrm{T}}$$

$$E(\Delta x, \Delta y) = (\Delta x \quad \Delta y)\mathbf{M}\begin{pmatrix}\Delta x \\ \Delta y\end{pmatrix}$$

其中

$$\mathbf{M} = \sum_{x,y} w(x,y)\begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

M是一个对称矩阵

$$\mathbf{M} = \mathbf{R}\begin{pmatrix}\lambda_1 & 0 \\ 0 & \lambda_2\end{pmatrix}\mathbf{R}^{\mathrm{T}}$$

M是半正定的

$$E(\Delta x, \Delta y) = (\Delta x \quad \Delta y)\mathbf{M}\begin{pmatrix}\Delta x \\ \Delta y\end{pmatrix}$$

其中

$$\mathbf{M} = \sum_{x,y} w(x,y)\begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

M是一个对称矩阵

$$\mathbf{M} = \mathbf{R}\begin{pmatrix}\lambda_1 & 0 \\ 0 & \lambda_2\end{pmatrix}\mathbf{R}^{\mathrm{T}}$$

M是半正定的

$$E(\Delta x, \Delta y) = (\Delta x \quad \Delta y)\mathbf{M}\begin{pmatrix}\Delta x \\ \Delta y\end{pmatrix}$$

其中

$$\mathbf{M} = \sum_{x,y} w(x,y)\begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

M是一个对称矩阵

$$\mathbf{M} = \mathbf{R}\begin{pmatrix}\lambda_1 & 0 \\ 0 & \lambda_2\end{pmatrix}\mathbf{R}^{\mathrm{T}}$$

M是半正定的

特征值非负

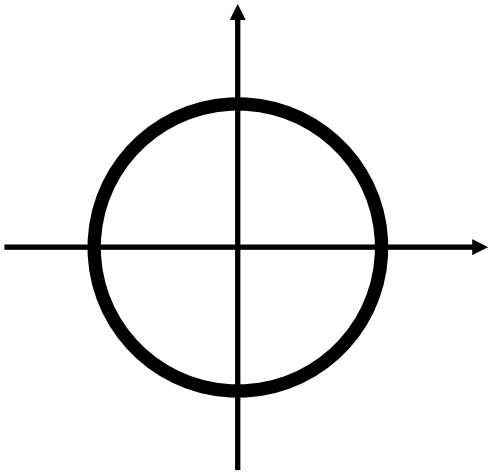
$$\mathbf{R}\begin{pmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{pmatrix}\mathbf{R}^{\mathrm{T}}$$

$$\mathbf{R} \qquad \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \qquad \mathbf{R}^{\mathrm{T}}$$

$$\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}$$

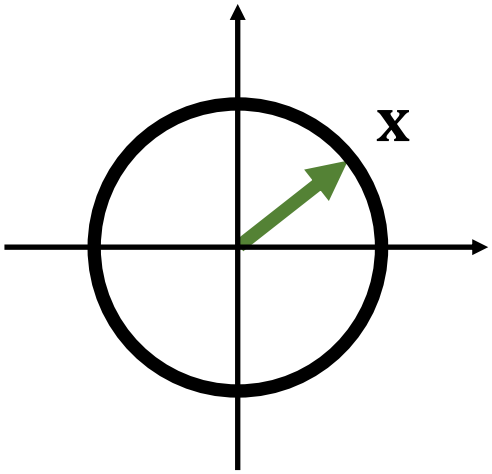
$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$

$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$



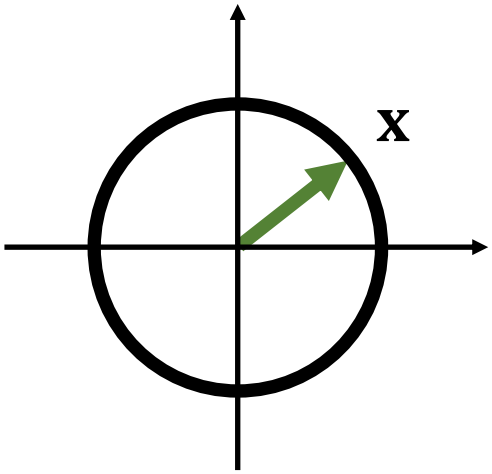
单位圆

$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$



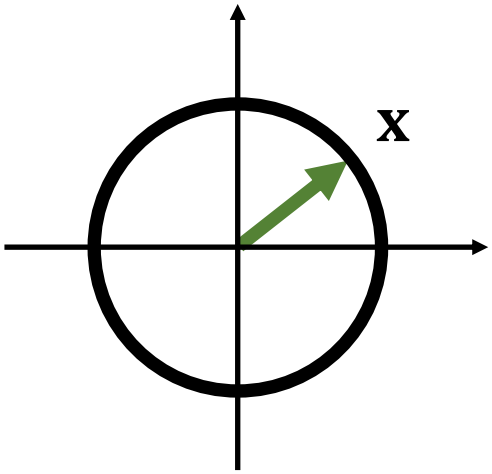
单位圆

$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$



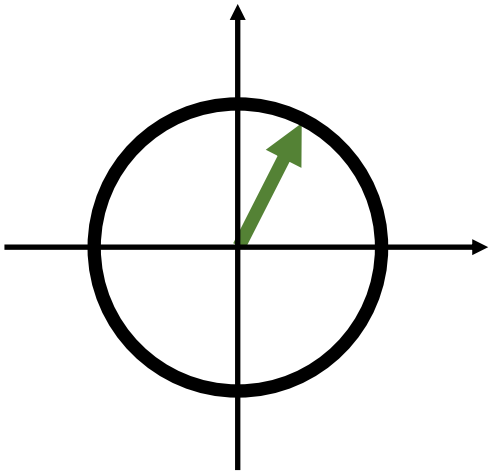
单位圆

$$\mathbf{x}^T\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^T\mathbf{x}$$



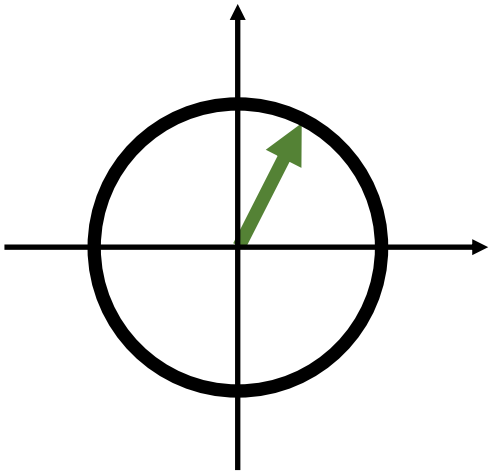
$\mathbf{R}^T\mathbf{x}$

$$\mathbf{x}^T\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^T\mathbf{x}$$

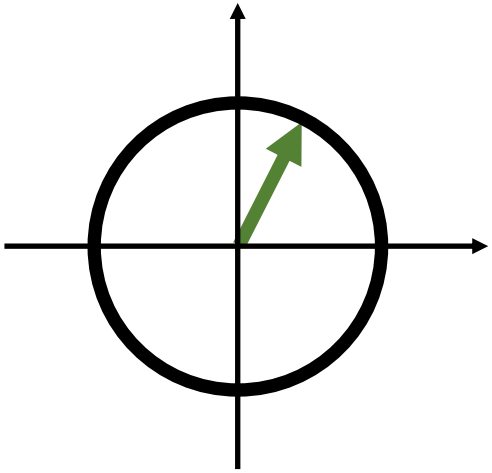


$\mathbf{R}^T\mathbf{x}$

$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$

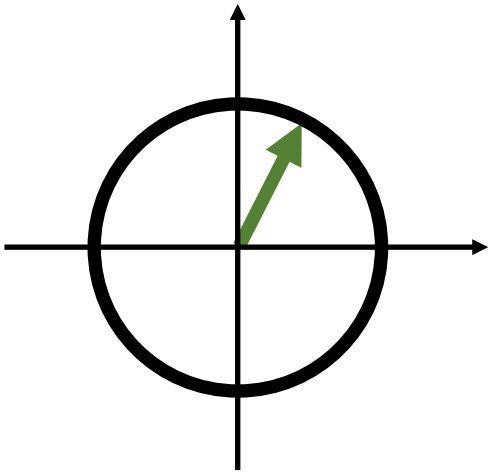


$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$



$$\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$

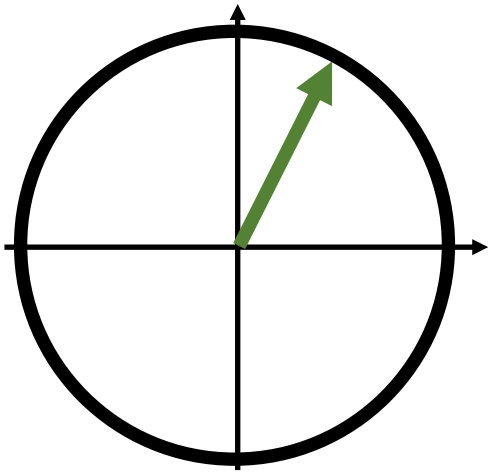
$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$



$$\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$

如果$\lambda_1 = \lambda_2$且大于1

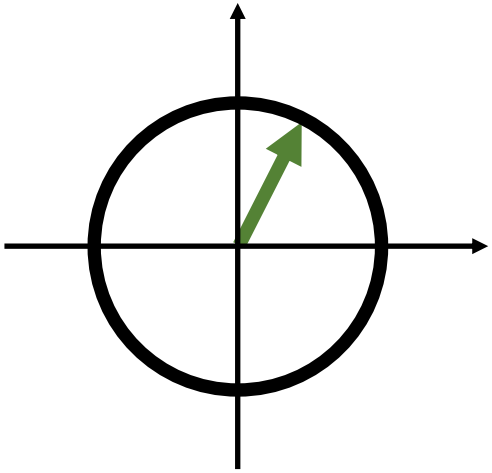
$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$



$$\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$

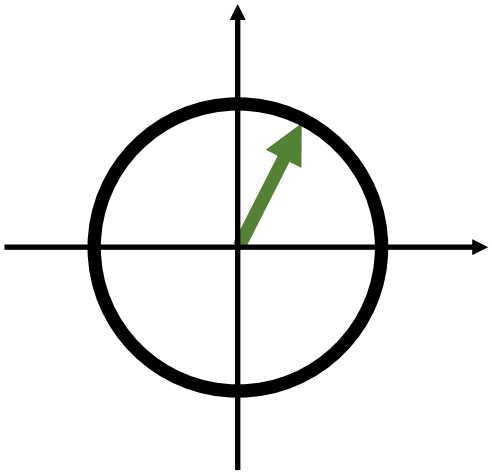
如果$\lambda_1 = \lambda_2$且大于1

$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$



$$\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$

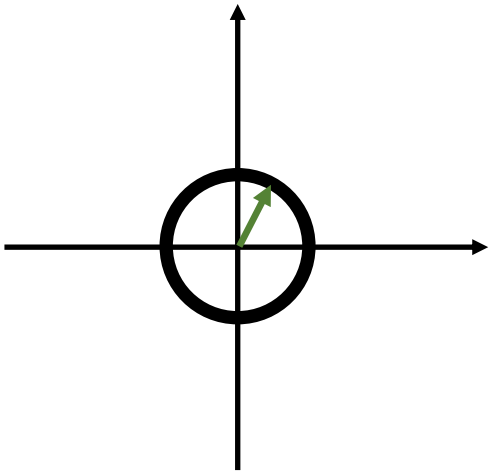
$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$



$$\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$

如果$\lambda_1 = \lambda_2$且小于1

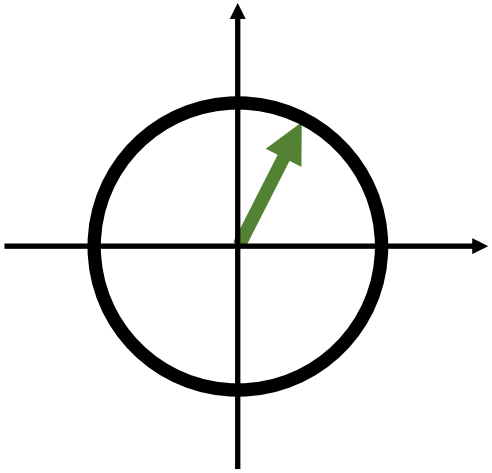
$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$



$$\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$

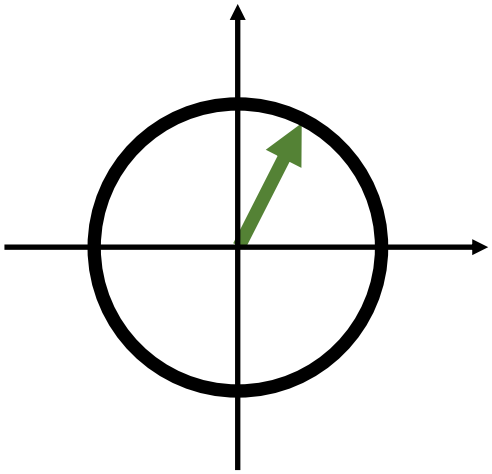
如果$\lambda_1 = \lambda_2$且小于1

$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$



$$\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$

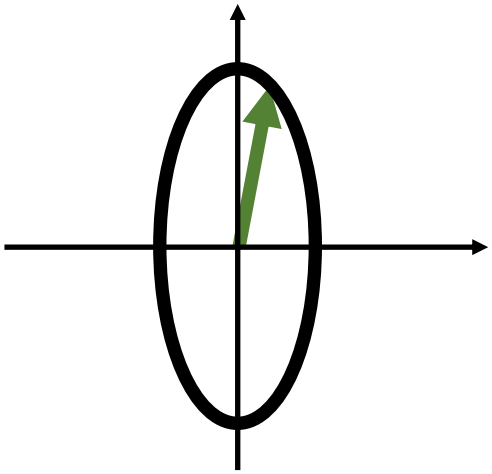
$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$



$$\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$

如果$\lambda_2 \gg \lambda_1$且大于1

$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$



$$\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$

如果$\lambda_2 \gg \lambda_1$且大于1

$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$

$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$

以前在哪见过?

$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$

以前在哪见过?

右侧的转置

$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$

$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$

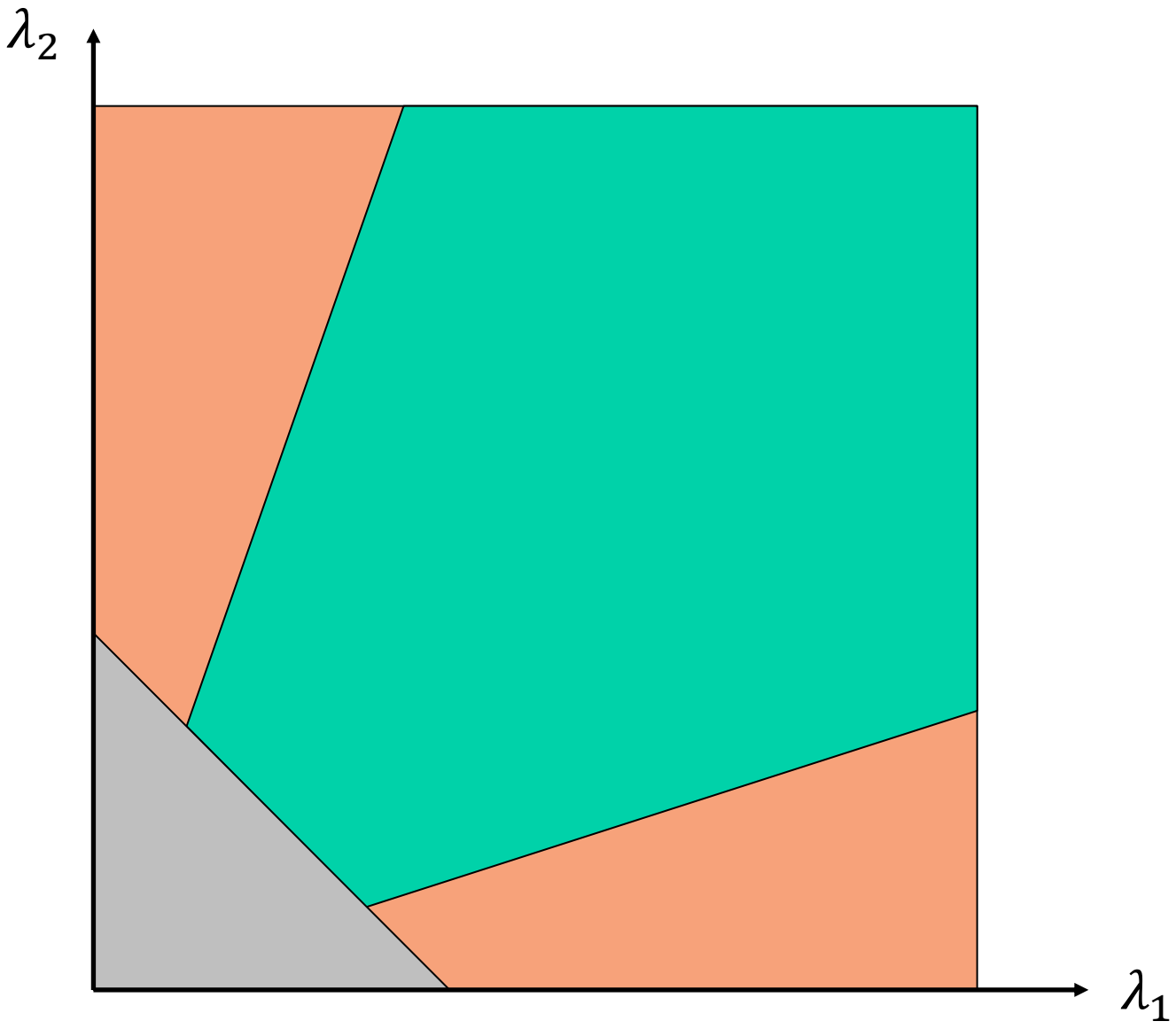
内积 (距离平方)

$$\mathbf{x}^{\mathrm{T}}\mathbf{R}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\begin{pmatrix}\sqrt{\lambda_1} & 0\\ 0 & \sqrt{\lambda_2}\end{pmatrix}\mathbf{R}^{\mathrm{T}}\mathbf{x}$$

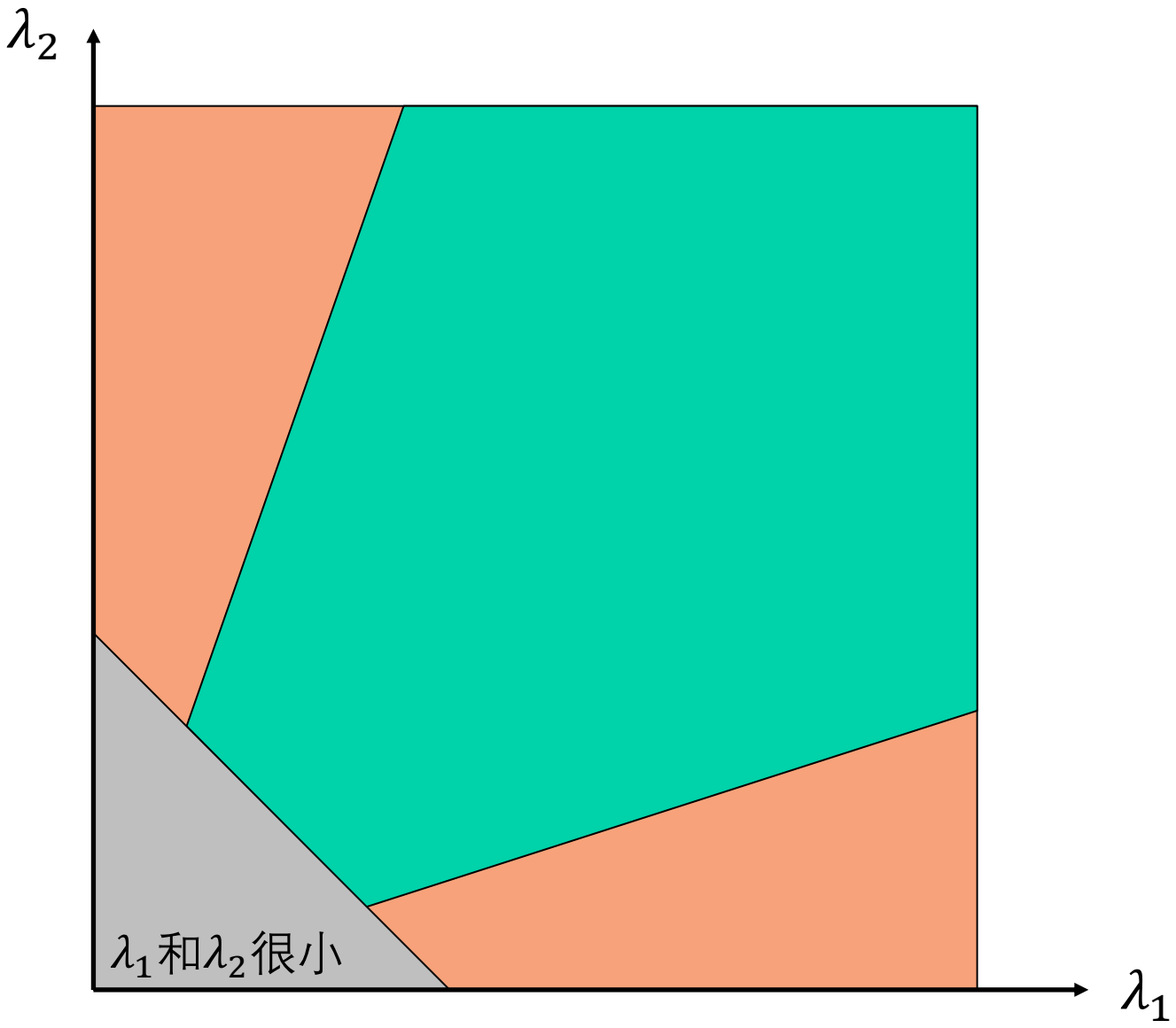
内积 (距离平方)

特征值表明一个块是否是一个角点

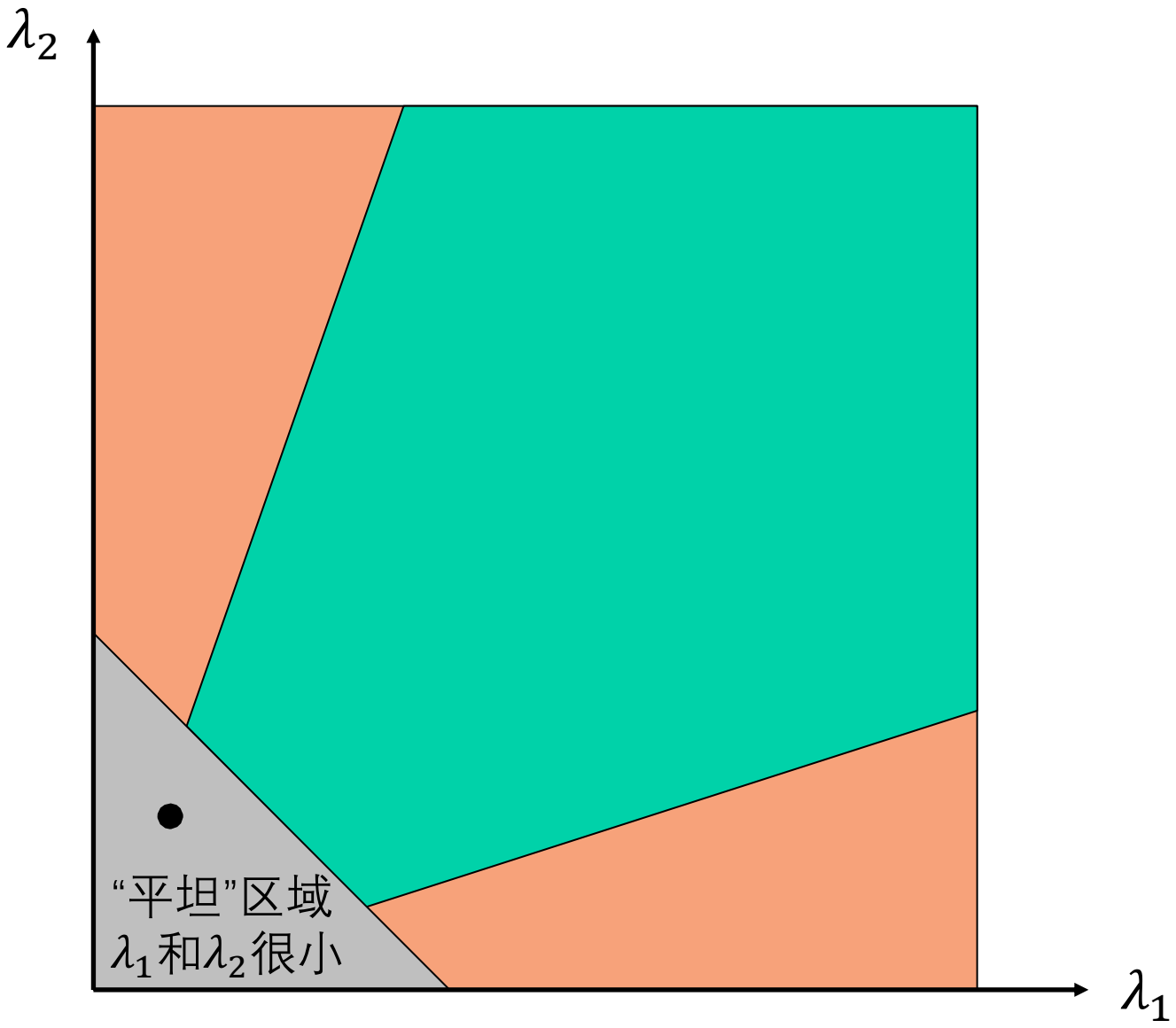
使用M的特征值对图像点分类



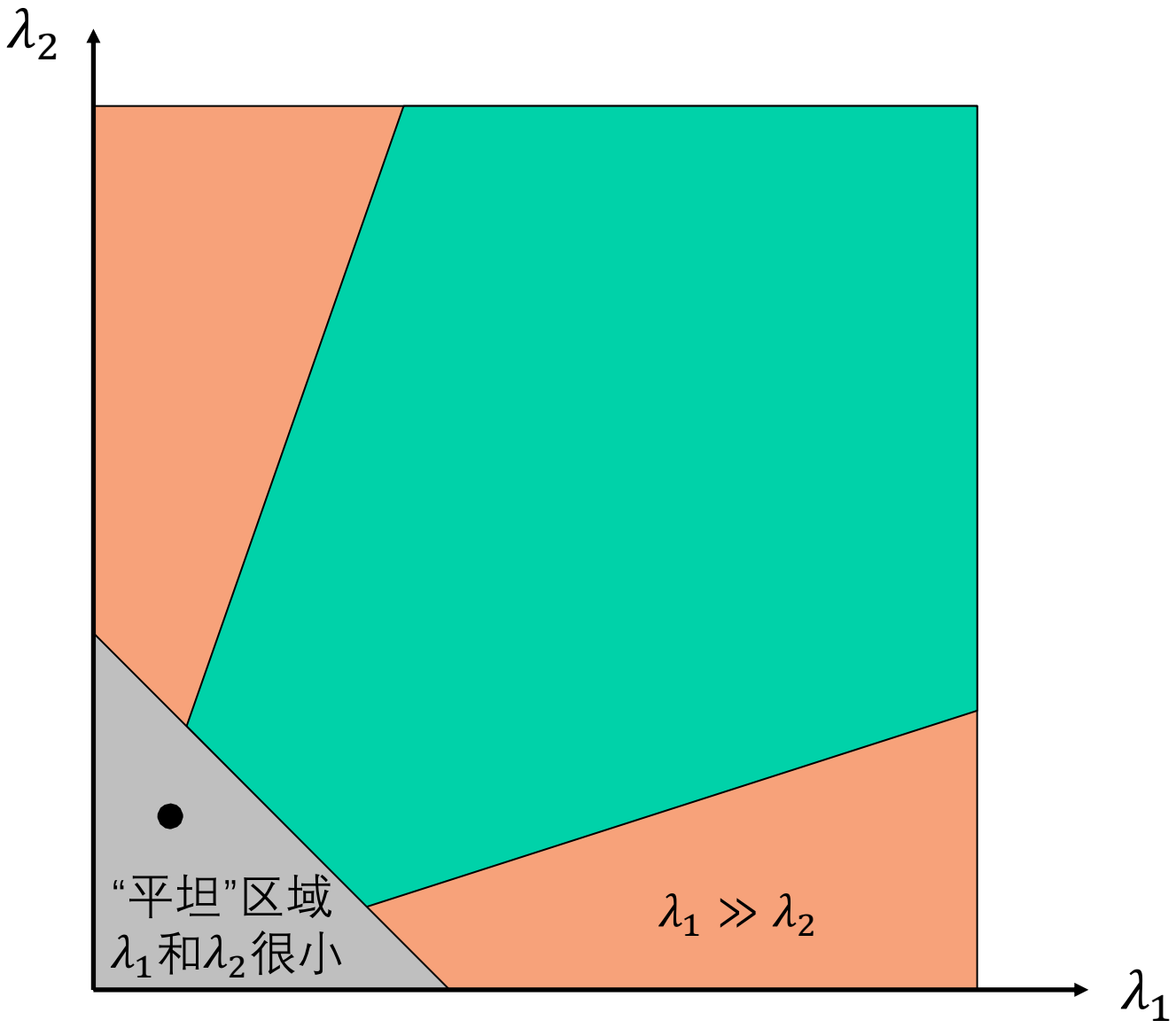
使用M的特征值对图像点分类



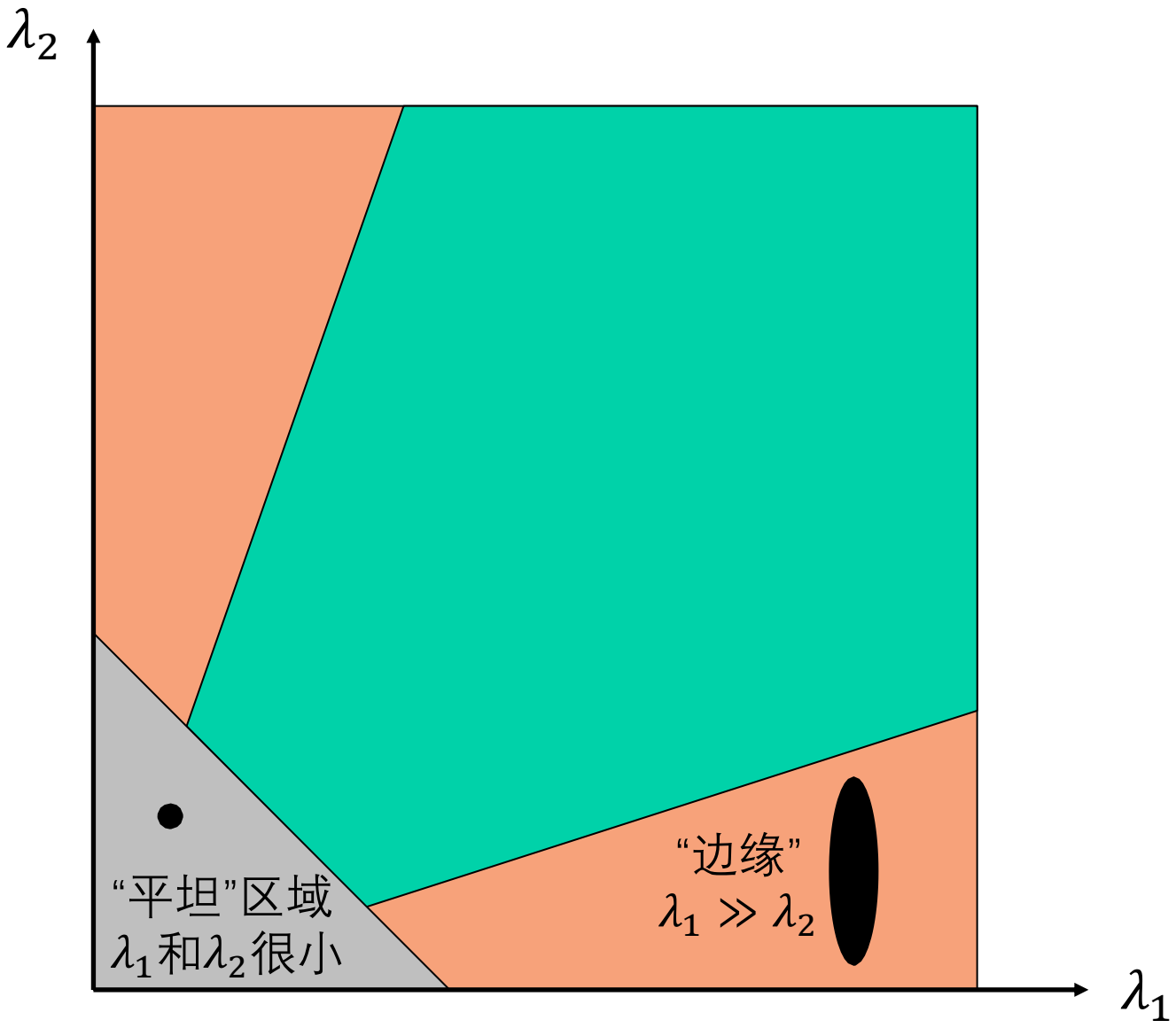
使用M的特征值对图像点分类



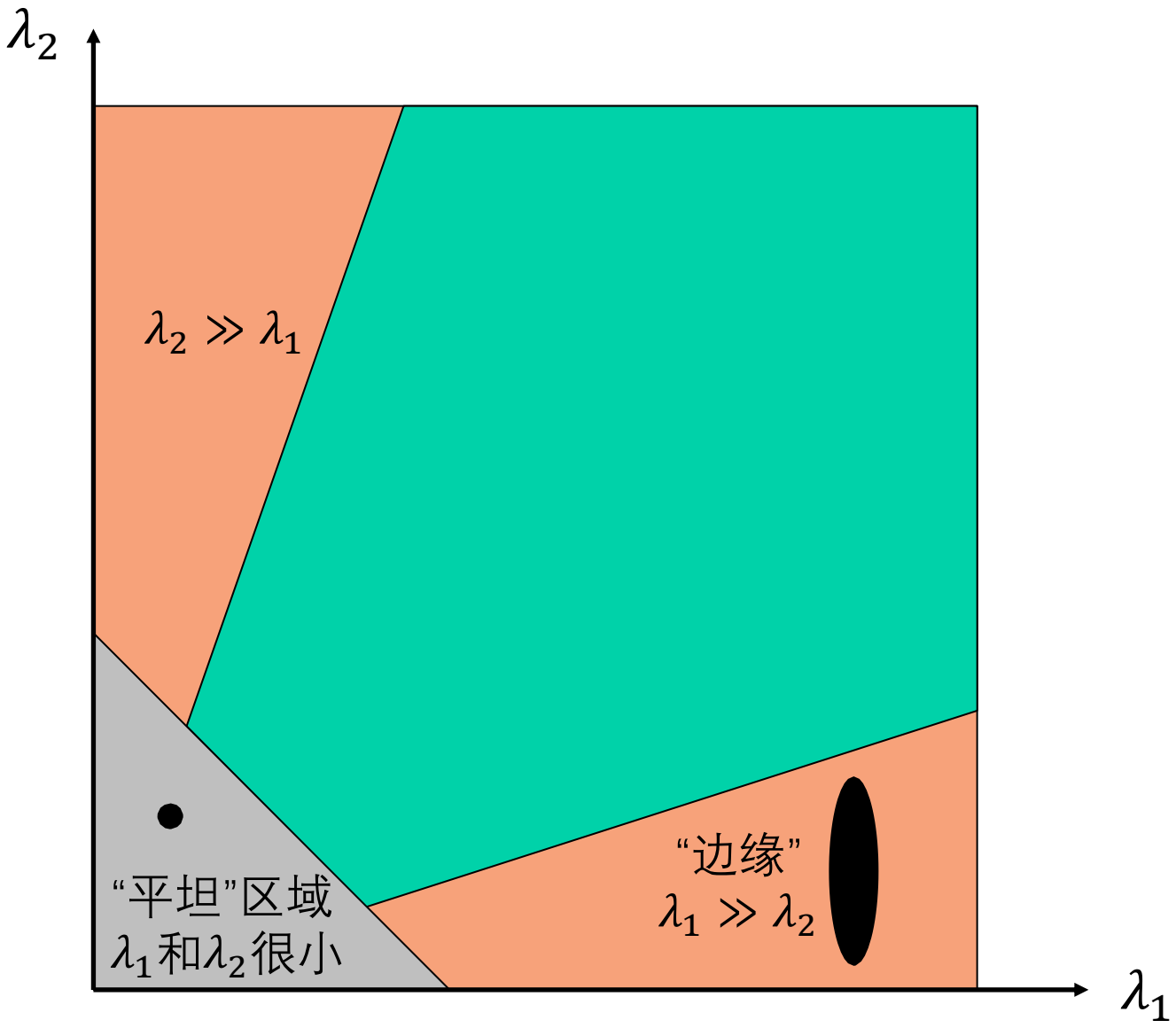
使用M的特征值对图像点分类



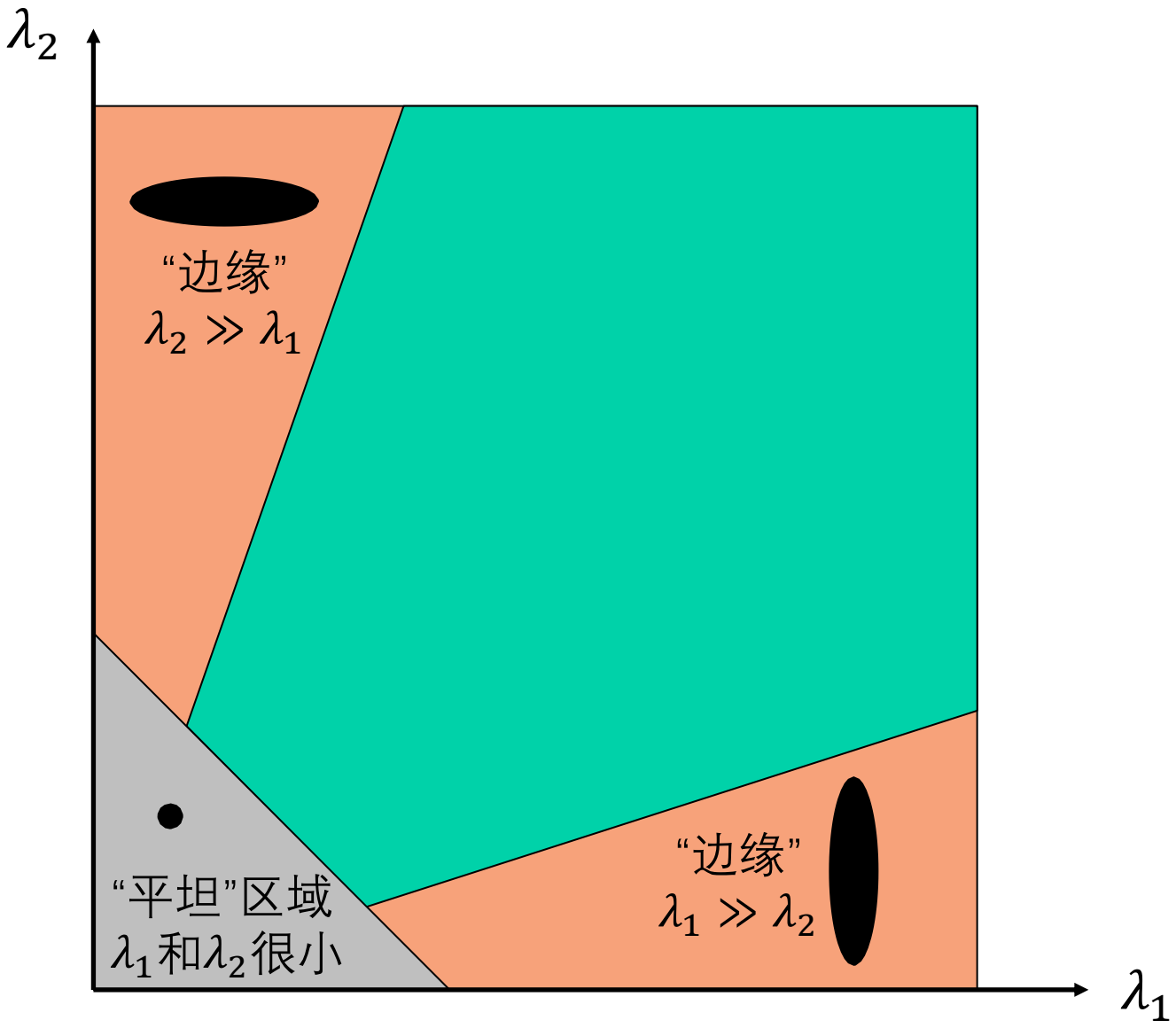
使用M的特征值对图像点分类



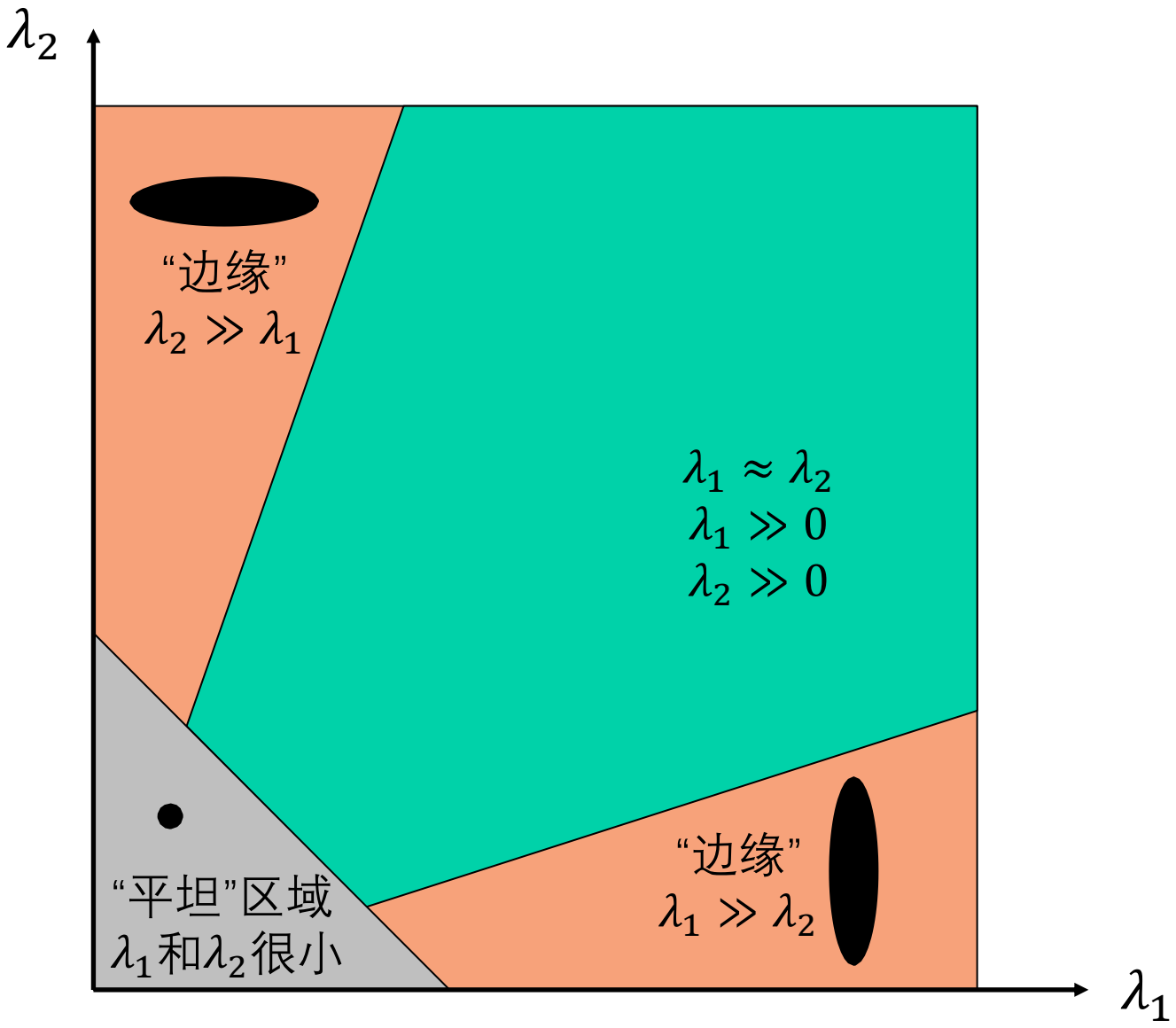
使用M的特征值对图像点分类



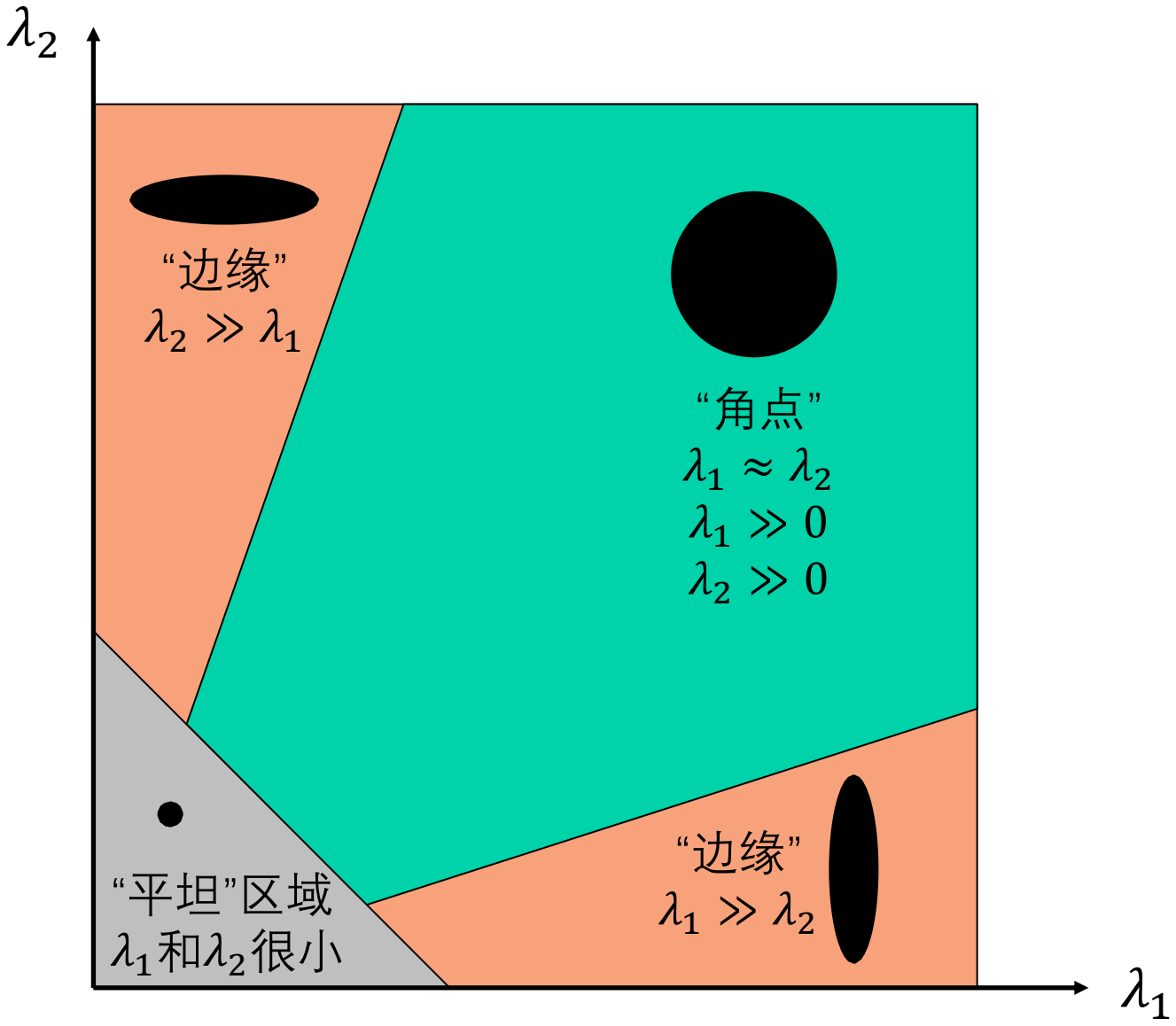
使用M的特征值对图像点分类



使用M的特征值对图像点分类



使用M的特征值对图像点分类



计算每个图像点的特征值的计算代价很高

Harris角点响应函数

$$r = \det \mathbf{M} - k(\operatorname{trace} \mathbf{M})^2$$

k根据经验设置为常数：0.04 – 0.06

Harris角点响应函数

$$r = \det \mathbf{M} - k(\text{trace}\,\mathbf{M})^2$$

k根据经验设置为常数：0.04 – 0.06

特征值的隐式使用方法

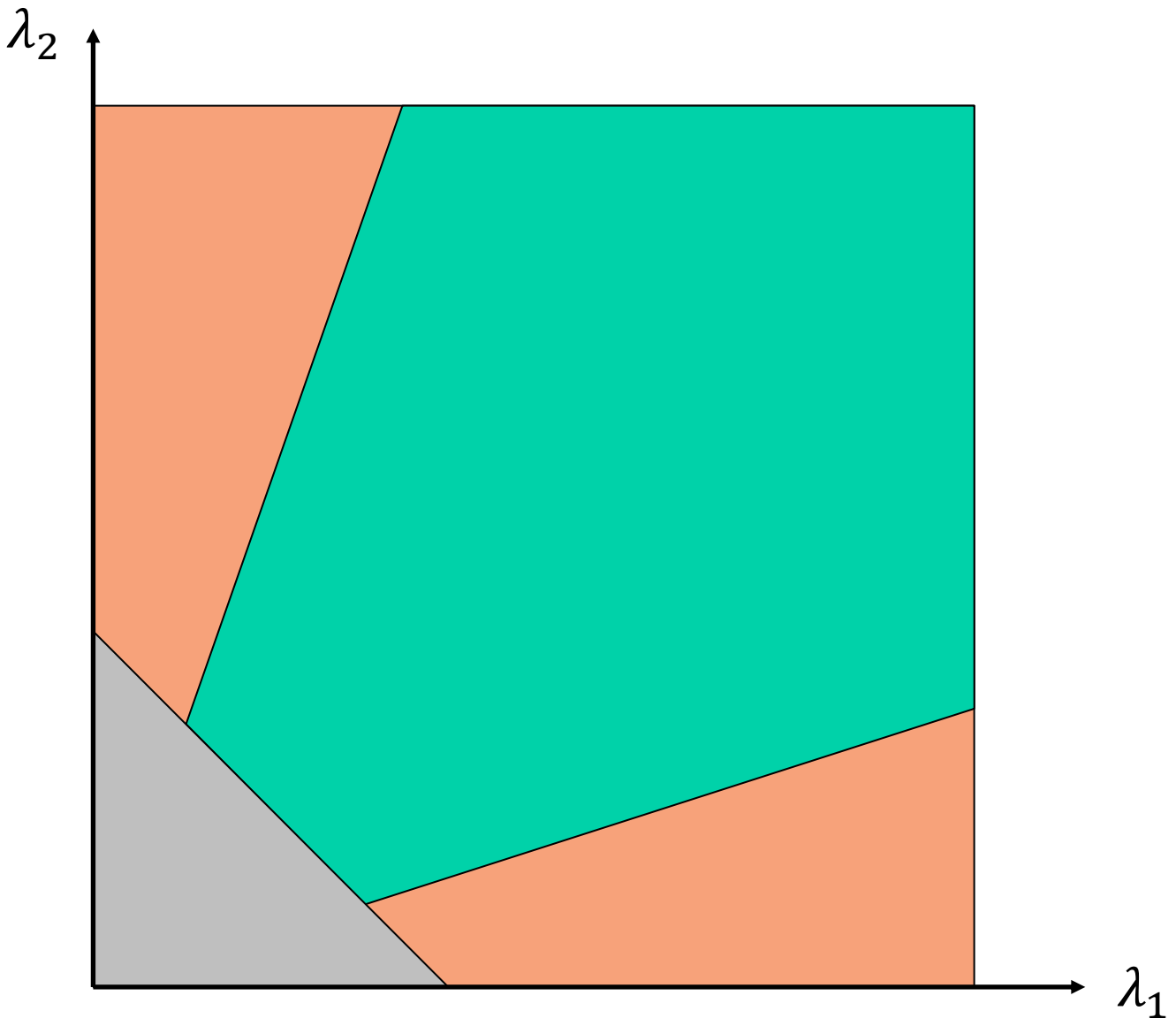
Harris角点响应函数

$$r = \det \mathbf{M} - k(\operatorname{trace} \mathbf{M})^2$$

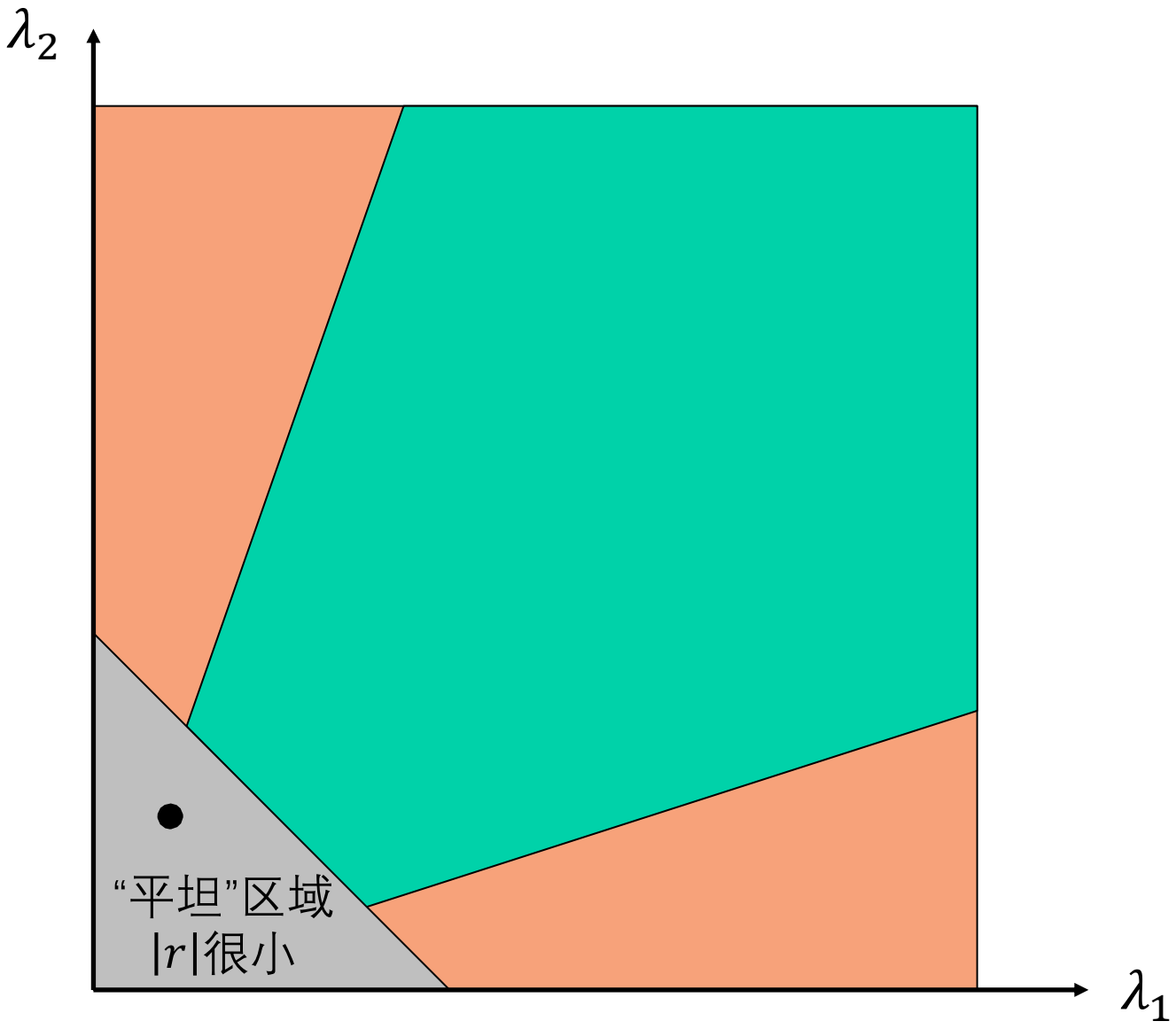
k根据经验设置为常数：0.04 – 0.06

$$\det \mathbf{M} = \lambda_1 \lambda_2 \qquad \operatorname{trace} \mathbf{M} = \lambda_1 + \lambda_2$$

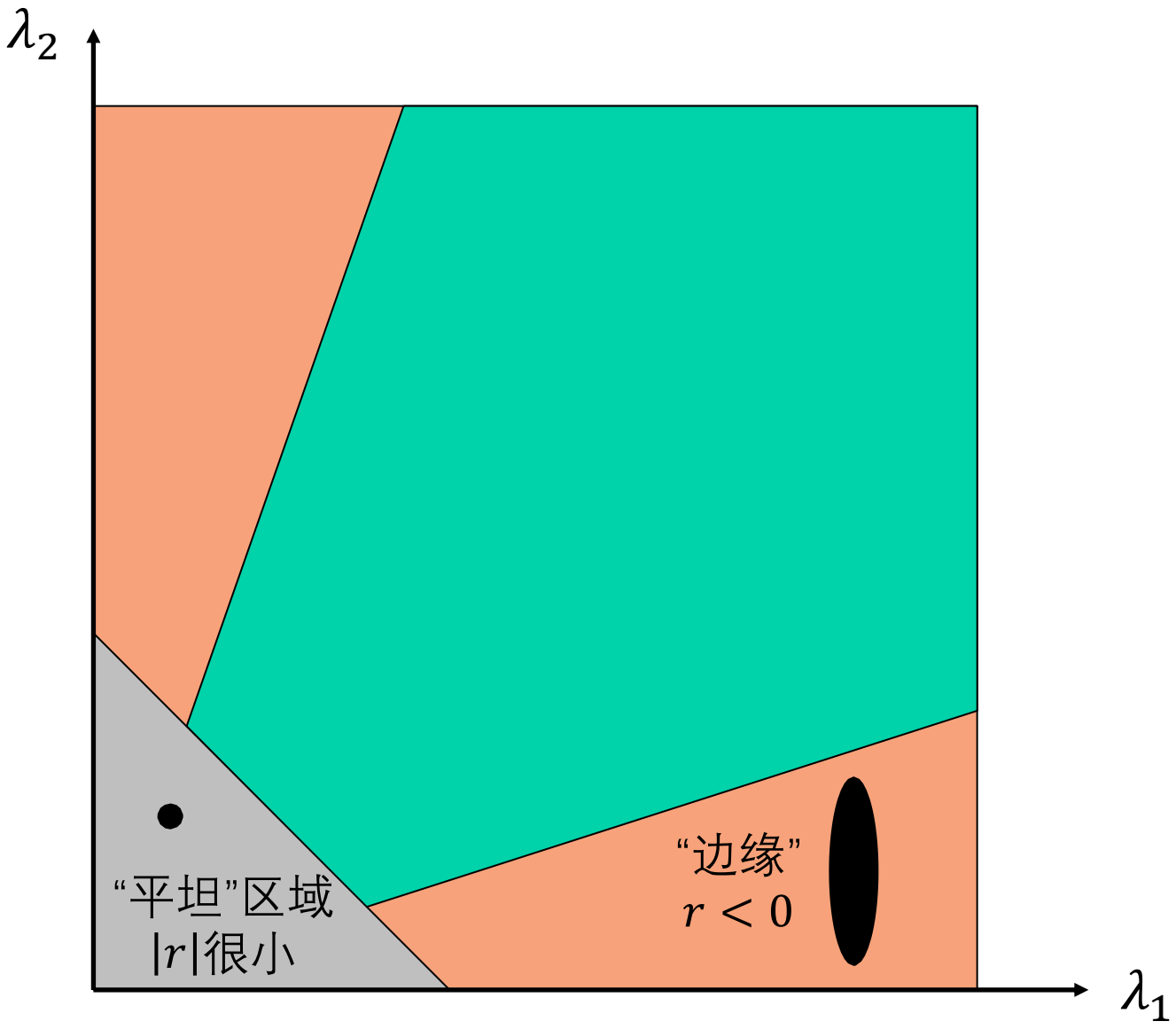
使用r对图像点分类



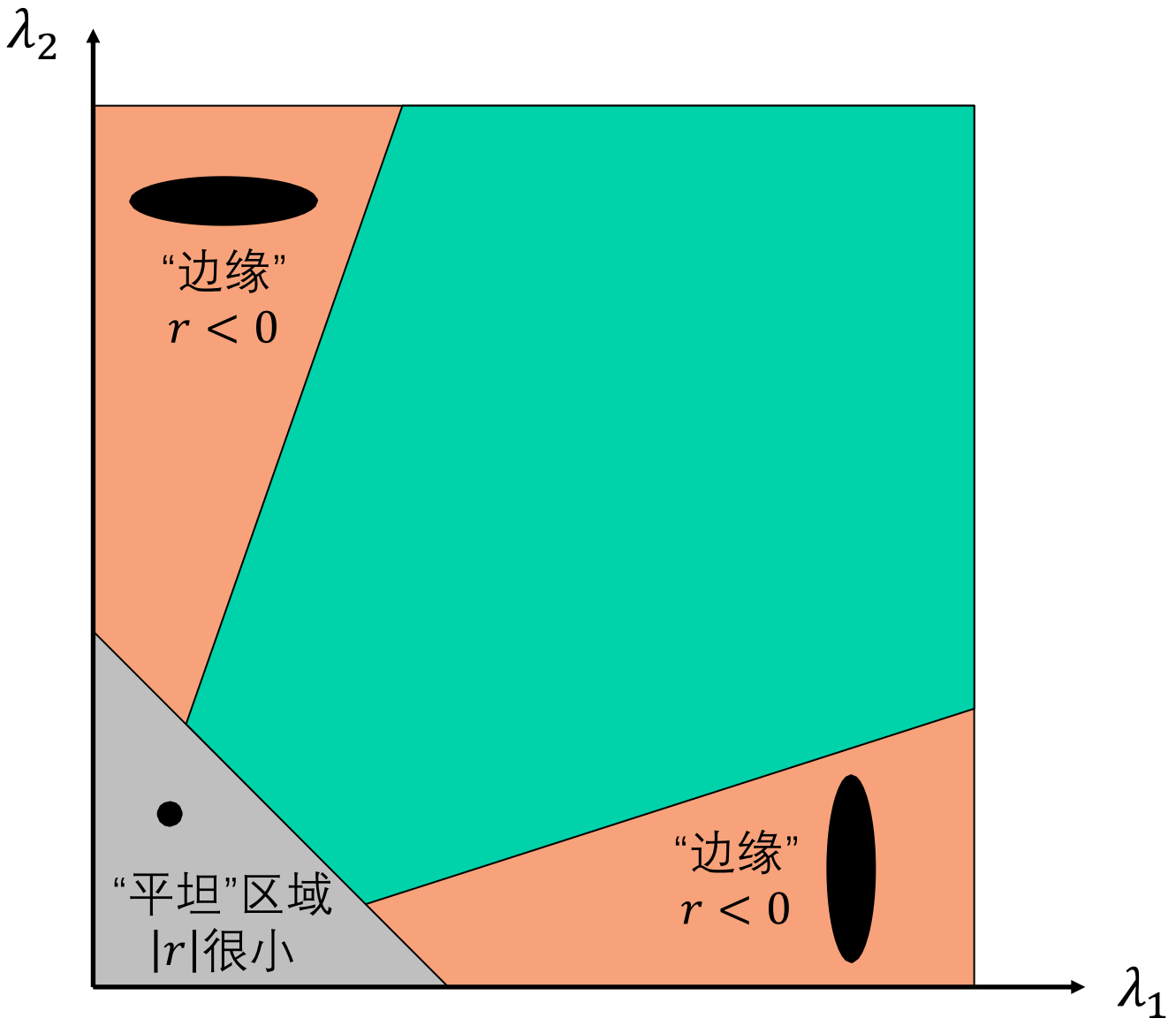
使用r对图像点分类



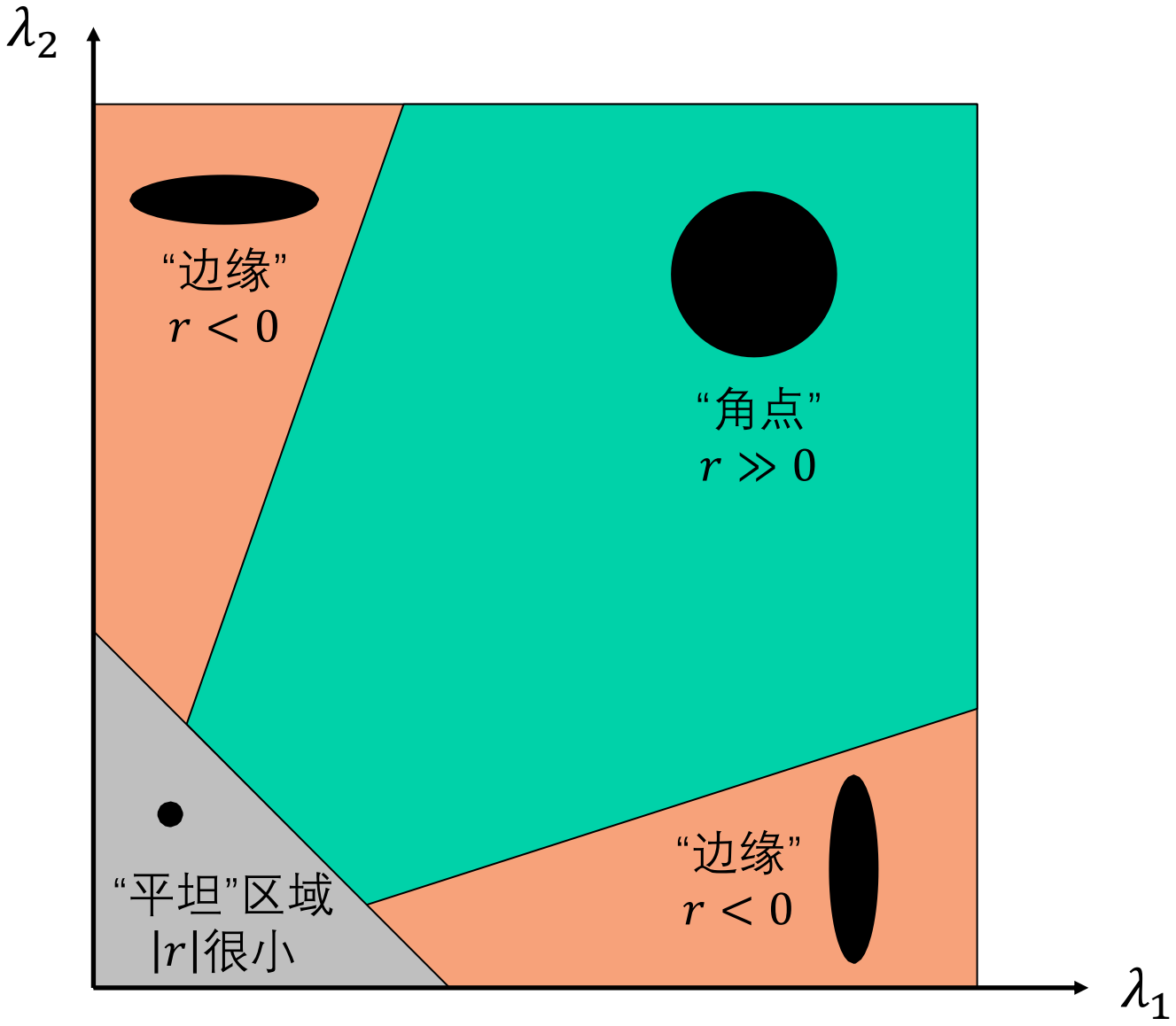
使用r对图像点分类



使用r对图像点分类



使用r对图像点分类



Harris角点性质

Harris角点性质

旋转不变性


输入图像

旋转变换

旋转变换

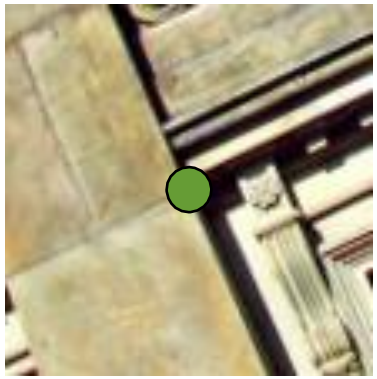
角点响应对旋转不变

旋转变换

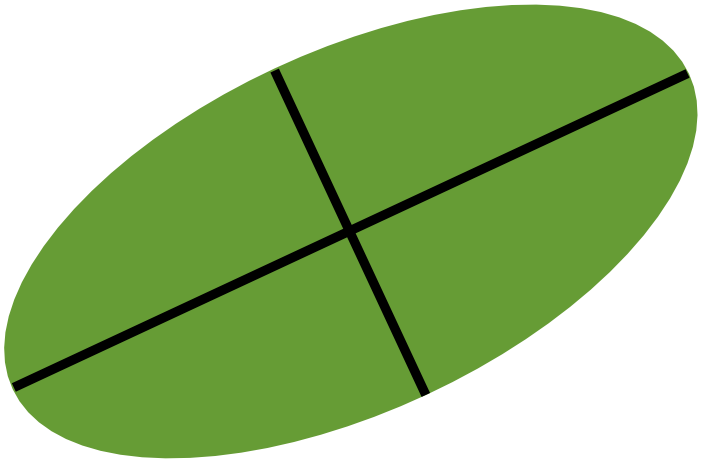
角点响应对旋转不变

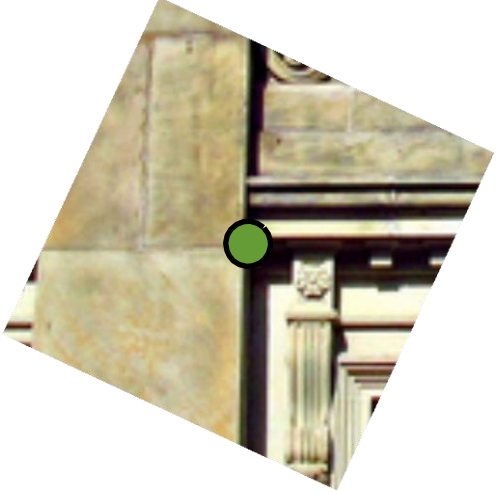
旋转变换

角点响应对旋转不变

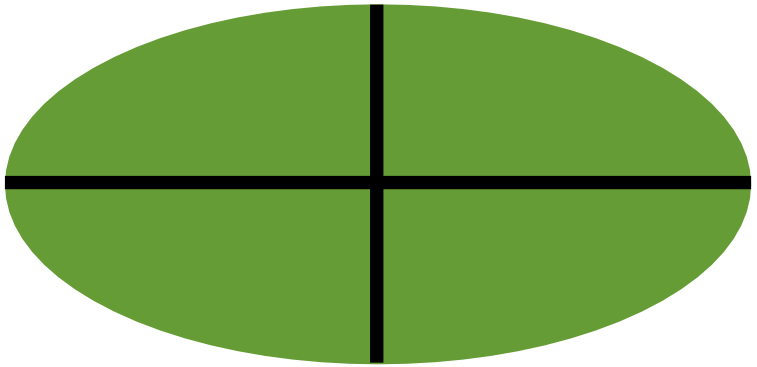


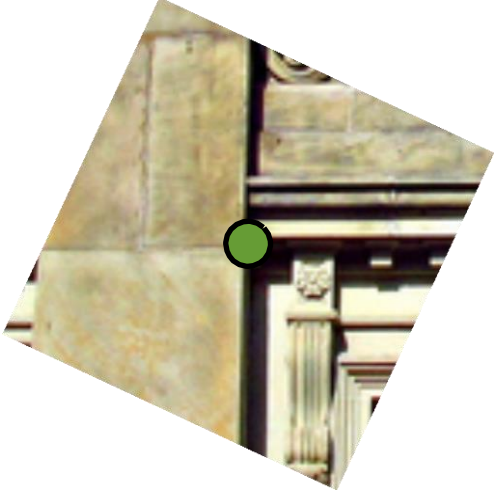
旋转不变？



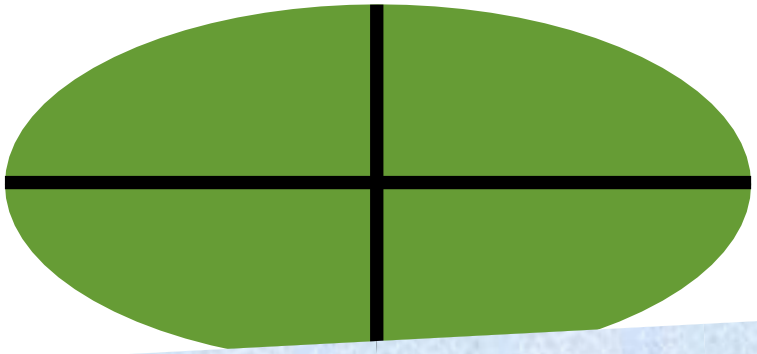


旋转不变？

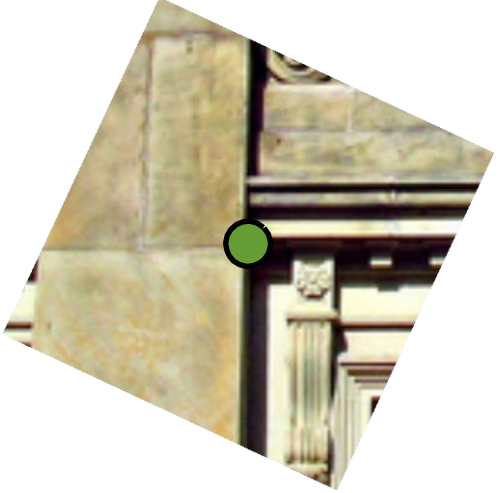




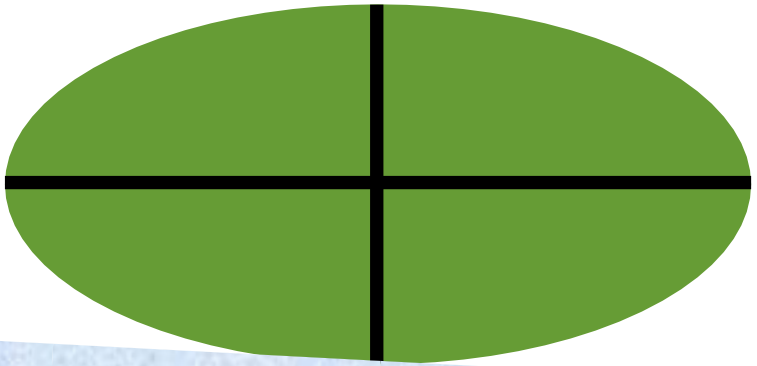
旋转不变？



形状保持不变



旋转不变？



还有什么保持不变？

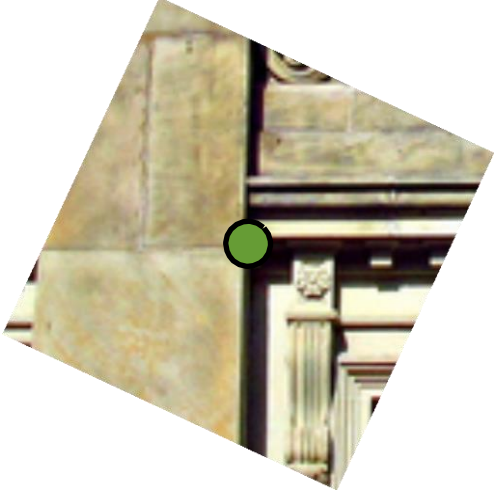
局部强度变化

$$E(\Delta x, \Delta y) = (\Delta x \quad \Delta y)\mathbf{M}\begin{pmatrix}\Delta x \\ \Delta y\end{pmatrix}$$

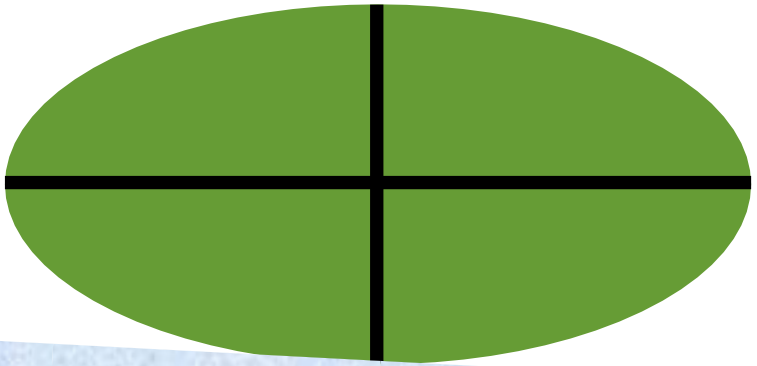
其中

$$\mathbf{M} = \sum_{x,y} w(x,y)\begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

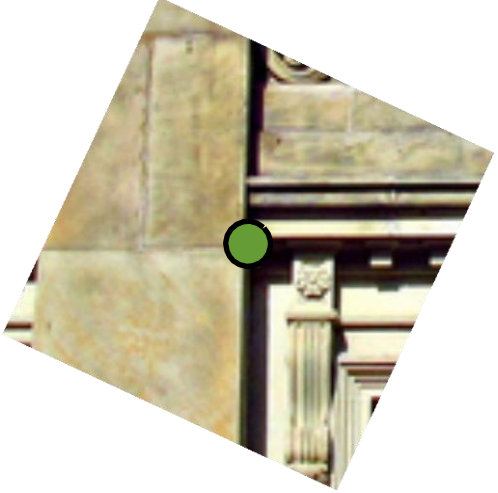
$$= \mathbf{R}\begin{pmatrix}\lambda_1 & 0 \\ 0 & \lambda_2\end{pmatrix}\mathbf{R}^{\mathrm{T}}$$



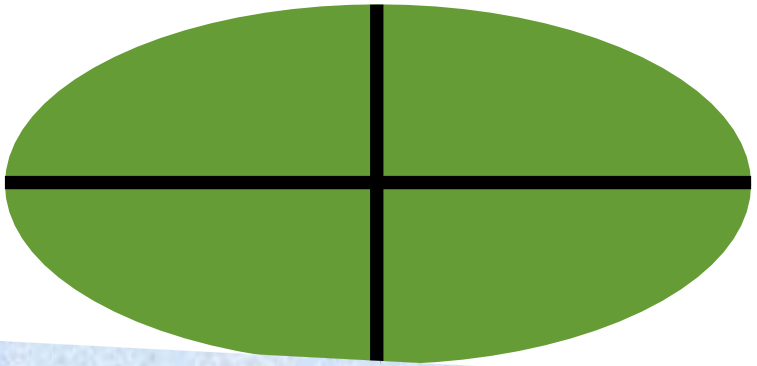
旋转不变？



还有什么保持不变？

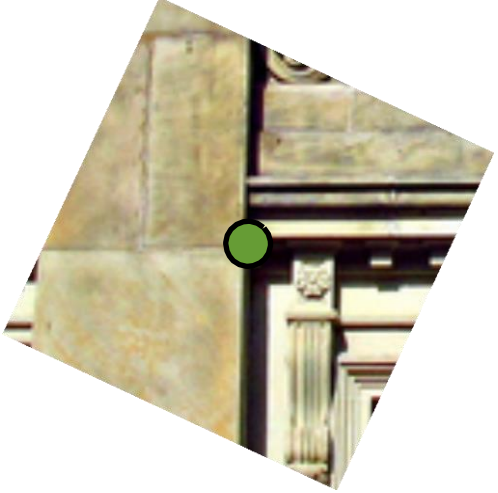


旋转不变？

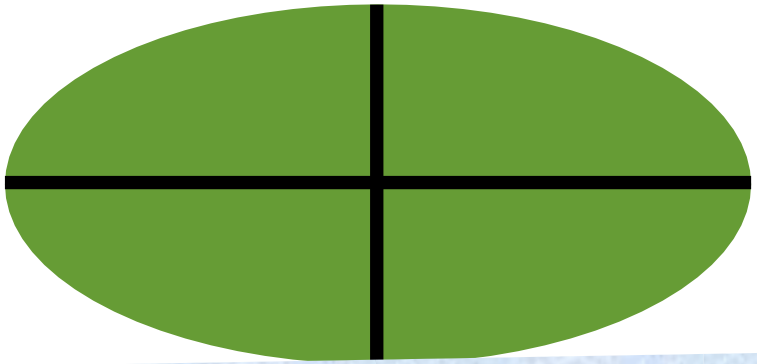


还有什么保持不变？

特征值



旋转不变？



Harris角点响应对旋转不变

Harris角点性质

旋转不变性

Harris角点性质

旋转不变性

对光照变化部分不变


输入图像

光度变换

$$I_{\text{new}}(x,y) = \alpha I(x,y) + \beta$$

$$I_{\text{new}}(x, y) = \alpha I(x, y) + \beta$$

$$I_{\text{new}}(x,y) = \alpha I(x,y) + \beta$$

角点响应是一个导数的函数

局部强度变化

$$E(\Delta x, \Delta y) = (\Delta x \quad \Delta y)\mathbf{M}\begin{pmatrix}\Delta x \\ \Delta y\end{pmatrix}$$

其中

$$\mathbf{M} = \sum_{x,y} w(x,y)\begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

局部强度变化

$$E(\Delta x, \Delta y) = (\Delta x \quad \Delta y)\mathbf{M}\begin{pmatrix}\Delta x \\ \Delta y\end{pmatrix}$$

其中

$$\mathbf{M} = \sum_{x,y} w(x,y)\begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

图像导数

$$I_{\text{new}}(x,y) = \alpha I(x,y) + \beta$$

角点响应是一个导数的函数

$$I_{\text{new}}(x,y) = \alpha I(x,y) + \beta$$

角点响应是一个导数的函数

$$\frac{\partial I_{\text{new}}(x,y)}{\partial x} = \alpha \frac{\partial I(x,y)}{\partial x}$$

$$I_{\text{new}}(x,y) = \alpha I(x,y) + \beta$$

角点响应是一个导数的函数

$$\frac{\partial I_{\text{new}}(x,y)}{\partial x} = \alpha \frac{\partial I(x,y)}{\partial x}$$

$$\frac{\partial I_{\text{new}}(x,y)}{\partial y} = \alpha \frac{\partial I(x,y)}{\partial y}$$

强度偏移

$$I_{\text{new}}(x,y) = \alpha I(x,y) + \beta$$

角点响应是一个导数的函数

$$\frac{\partial I_{\text{new}}(x,y)}{\partial x} = \alpha \frac{\partial I(x,y)}{\partial x}$$

$$\frac{\partial I_{\text{new}}(x,y)}{\partial y} = \alpha \frac{\partial I(x,y)}{\partial y}$$

强度偏移

$$I_{\text{new}}(x,y) = \alpha I(x,y) + \beta$$

角点响应是一个导数的函数

$$\frac{\partial I_{\text{new}}(x,y)}{\partial x} = \alpha \frac{\partial I(x,y)}{\partial x}$$

$$\frac{\partial I_{\text{new}}(x,y)}{\partial y} = \alpha \frac{\partial I(x,y)}{\partial y}$$

是否对强度偏移不变？

强度偏移

$$I_{\text{new}}(x,y) = \alpha I(x,y) + \beta$$

角点响应是一个导数的函数

$$\frac{\partial I_{\text{new}}(x,y)}{\partial x} = \alpha \frac{\partial I(x,y)}{\partial x}$$

$$\frac{\partial I_{\text{new}}(x,y)}{\partial y} = \alpha \frac{\partial I(x,y)}{\partial y}$$

是否对强度偏移不变？

是

强度尺度

$$I_{\text{new}}(x,y) = \alpha I(x,y) + \beta$$

角点响应是一个导数的函数

$$\frac{\partial I_{\text{new}}(x,y)}{\partial x} = \alpha \frac{\partial I(x,y)}{\partial x}$$

$$\frac{\partial I_{\text{new}}(x,y)}{\partial y} = \alpha \frac{\partial I(x,y)}{\partial y}$$

强度尺度

$$I_{\text{new}}(x,y) = \alpha I(x,y) + \beta$$

角点响应是一个导数的函数

$$\frac{\partial I_{\text{new}}(x,y)}{\partial x} = \alpha \frac{\partial I(x,y)}{\partial x}$$

$$\frac{\partial I_{\text{new}}(x,y)}{\partial y} = \alpha \frac{\partial I(x,y)}{\partial y}$$

是否对强度尺度不变？

强度尺度

$$I_{\text{new}}(x,y) = \alpha I(x,y) + \beta$$

角点响应是一个导数的函数

$$\frac{\partial I_{\text{new}}(x,y)}{\partial x} = \alpha \frac{\partial I(x,y)}{\partial x}$$

$$\frac{\partial I_{\text{new}}(x,y)}{\partial y} = \alpha \frac{\partial I(x,y)}{\partial y}$$

是否对强度尺度不变？

强度尺度

$$I_{\text{new}}(x,y) = \alpha I(x,y) + \beta$$

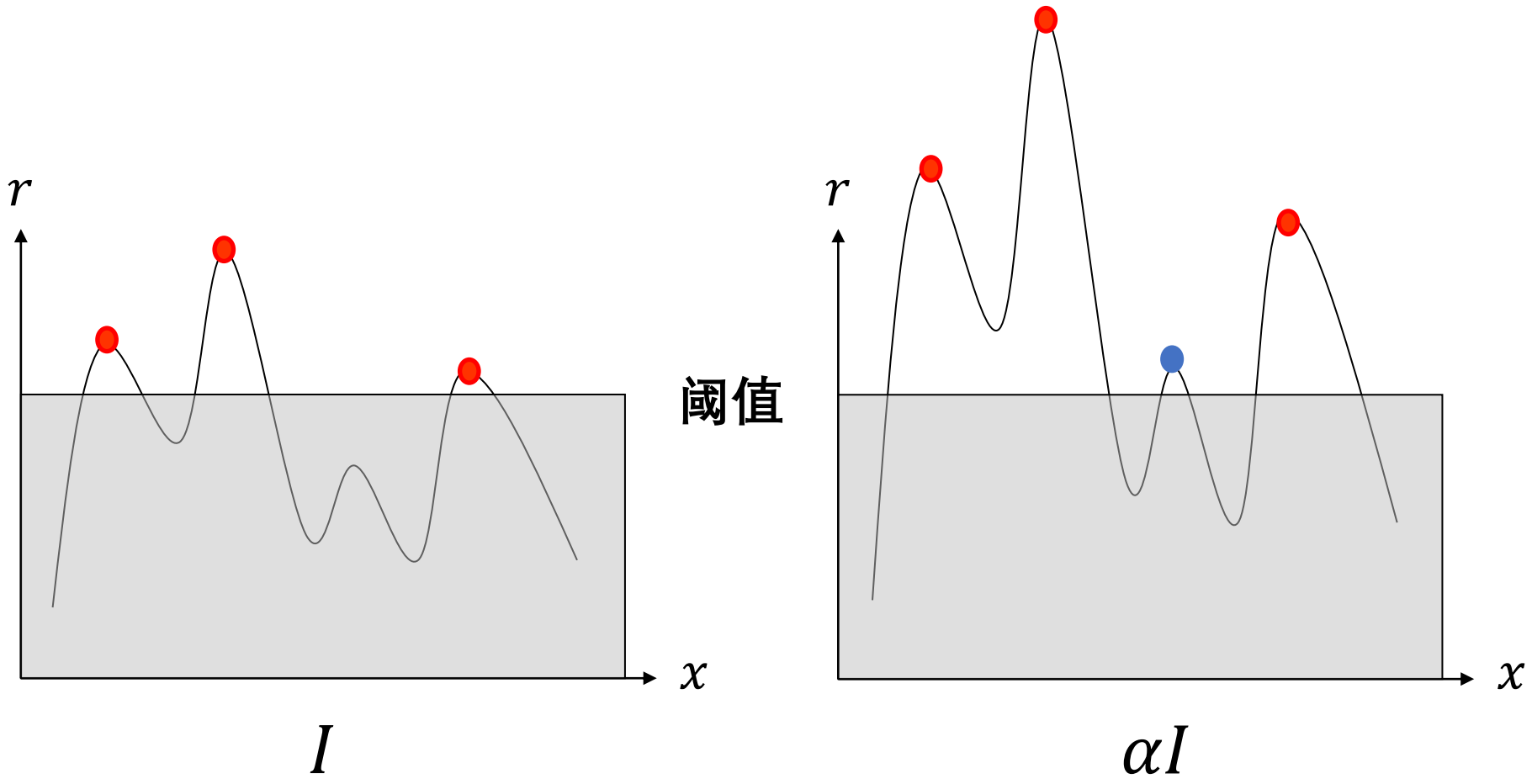
角点响应是一个导数的函数

$$\frac{\partial I_{\text{new}}(x,y)}{\partial x} = \alpha \frac{\partial I(x,y)}{\partial x}$$

$$\frac{\partial I_{\text{new}}(x,y)}{\partial y} = \alpha \frac{\partial I(x,y)}{\partial y}$$

是否对强度尺度不变？

否


r
x
阈值
r
x
I
αI

强度尺度

$$I_{\text{new}}(x,y) = \alpha I(x,y) + \beta$$

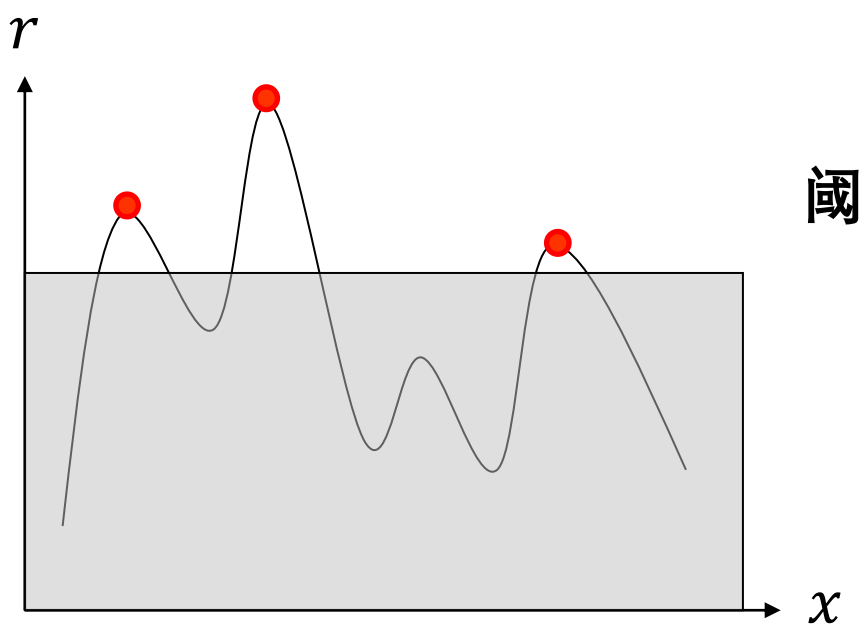
角点响应是一个导数的函数

$$\frac{\partial I_{\text{new}}(x,y)}{\partial x} = \alpha \frac{\partial I(x,y)}{\partial x}$$

$$\frac{\partial I_{\text{new}}(x,y)}{\partial y} = \alpha \frac{\partial I(x,y)}{\partial y}$$

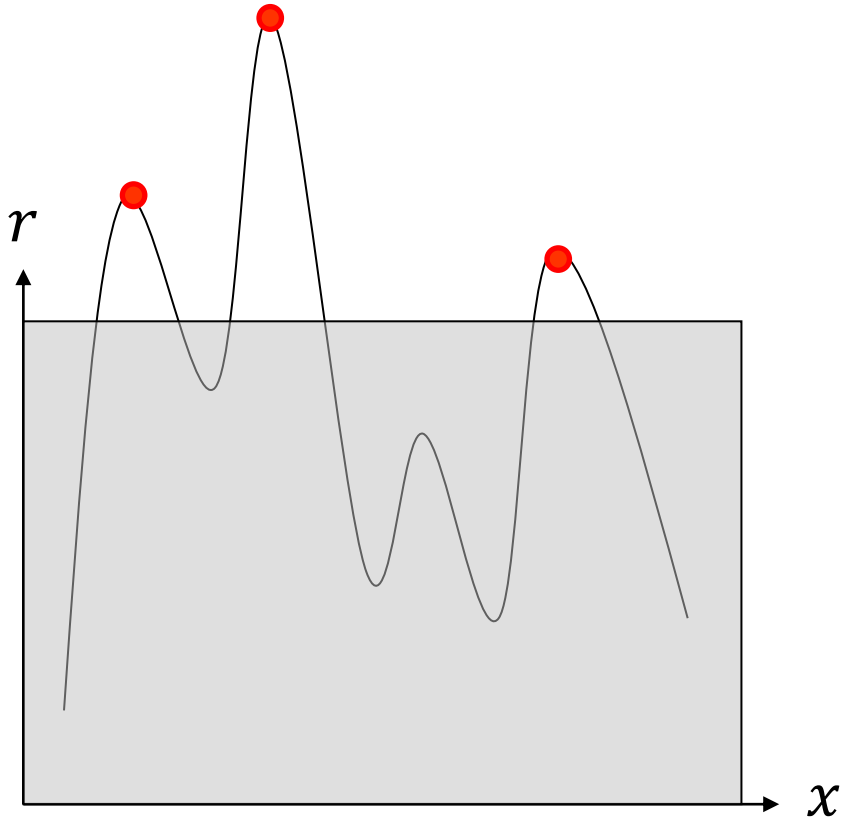
是否对强度尺度不变？

调整角点阈值



I

阈值



αI

Harris角点性质

旋转不变性

对光照变化部分不变

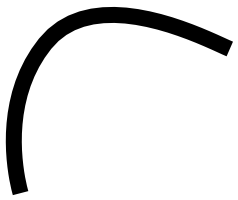
Harris角点性质

旋转不变性

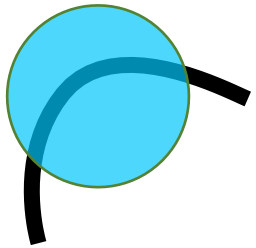
对光照变化部分不变

无尺度不变性

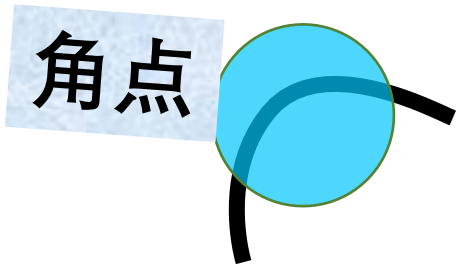
无尺度不变性



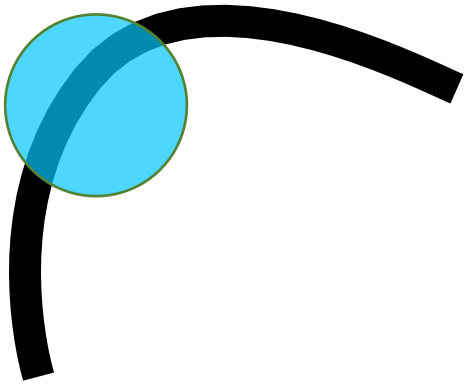
无尺度不变性



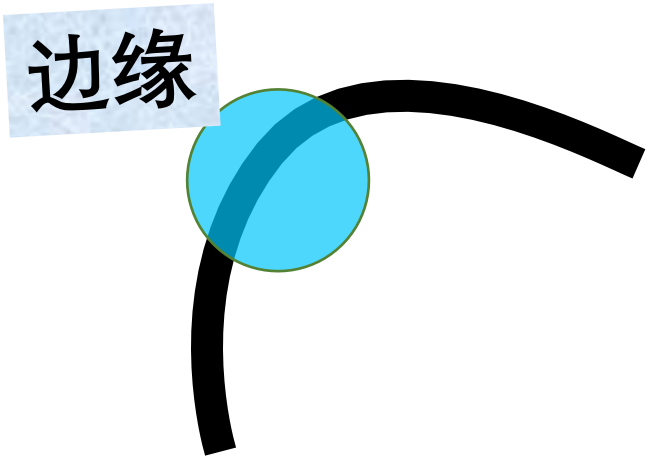
无尺度不变性



无尺度不变性



无尺度不变性



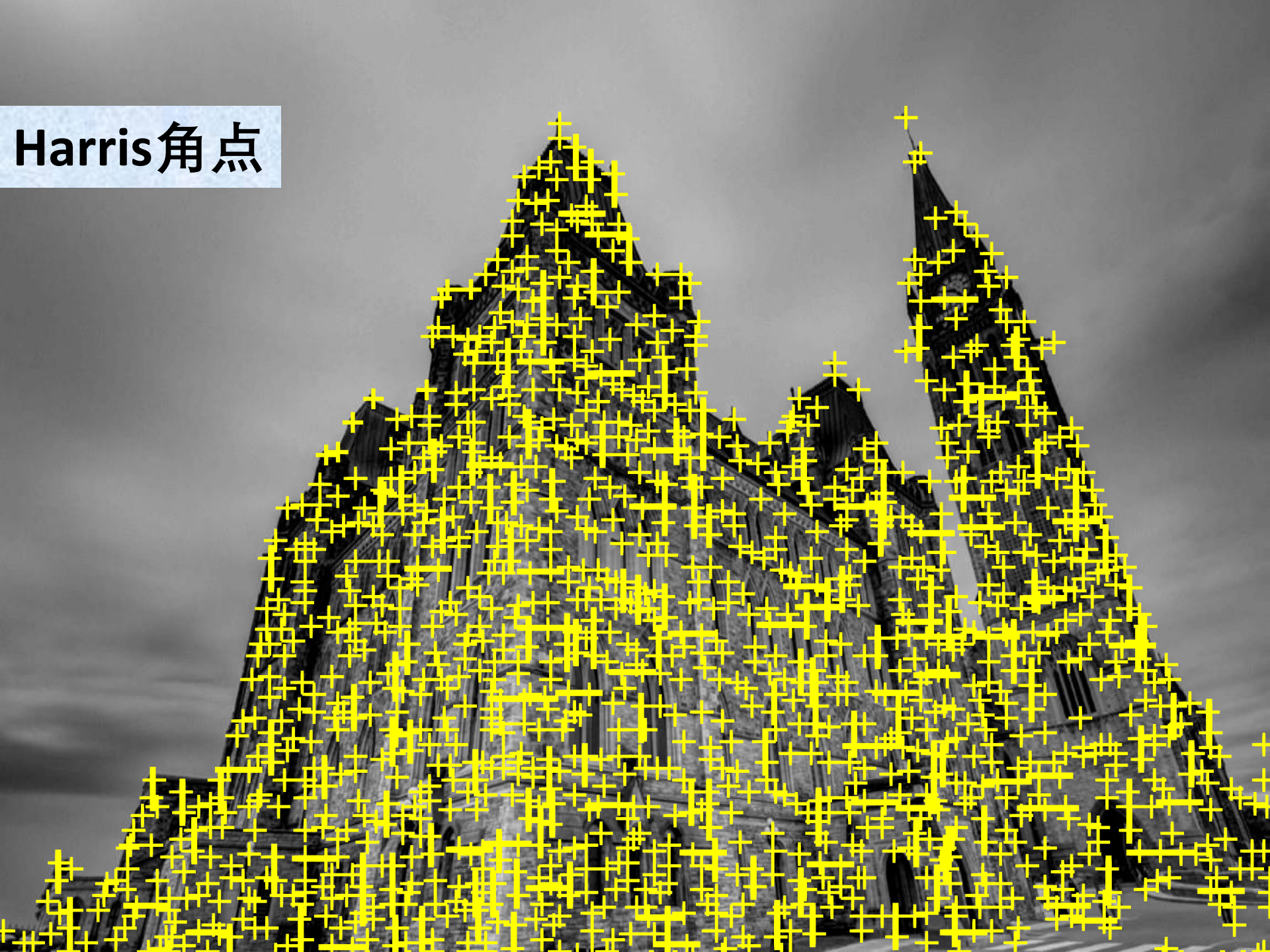
Harris角点性质

旋转不变性

对光照变化部分不变

无尺度不变性

Python时间


Harris角点

Harris角点检测器

```
im = cv2.imread('parliament.jpg', cv2.IMREAD_GRA
im = im.astype(float)/255.0

sigma = 5
g = cv2.getGaussianKernel(2*sigma*3+1, sigma)

Ix = cv2.Sobel(im, -1, dx=1, dy=0)
Iy = cv2.Sobel(im, -1, dx=0, dy=1)

Ix2 = cv2.filter2D(Ix**2, -1, g, cv2.BORDER_REFLECT)
Iy2 = cv2.filter2D(Iy**2, -1, g, cv2.BORDER_REFLECT)
Ixy = cv2.filter2D(Ix*Iy, -1, g, cv2.BORDER_REFLECT)

k = 0.04  # k is between 0.04 and 0.06
# --- r = Det(M) – k·Trace(M)^2 ---
r = (Ix2*Iy2 - Ixy**2) - k*(Ix2 + Iy2)**2

# --- find local maxima and threshold ---
```

```
im = cv2.imread('parliament.jpg', cv2.IMREAD_GRAYSCALE)
im = im.astype(float)/255.0

sigma = 5
g = cv2.getGaussianKernel(2*sigma*3+1, sigma)

Ix = cv2.Sobel(im, -1, dx=1, dy=0)
Iy = cv2.Sobel(im, -1, dx=0, dy=1)

Ix2 = cv2.filter2D(Ix**2, -1, g, cv2.BORDER_REFLECT)
Iy2 = cv2.filter2D(Iy**2, -1, g, cv2.BORDER_REFLECT)
Ixy = cv2.filter2D(Ix*Iy, -1, g, cv2.BORDER_REFLECT)

k = 0.04  # k is between 0.04 and 0.06
# --- r = Det(M) – k·Trace(M)^2 ---
r = (Ix2*Iy2 - Ixy**2) - k*(Ix2 + Iy2)**2

# --- find local maxima and threshold ---
```


输入图像

```
im = cv2.imread('parliament.jpg', cv2.IMREAD_GRAYSCALE)
im = im.astype(float)/255.0

sigma = 5
g = cv2.getGaussianKernel(2*sigma*3+1, sigma)

Ix = cv2.Sobel(im, -1, dx=1, dy=0)
Iy = cv2.Sobel(im, -1, dx=0, dy=1)

Ix2 = cv2.filter2D(Ix**2, -1, g, cv2.BORDER_REFLECT)
Iy2 = cv2.filter2D(Iy**2, -1, g, cv2.BORDER_REFLECT)
Ixy = cv2.filter2D(Ix*Iy, -1, g, cv2.BORDER_REFLECT)

k = 0.04  # k is between 0.04 and 0.06
# --- r = Det(M) – k·Trace(M)^2 ---
r = (Ix2*Iy2 - Ixy**2) - k*(Ix2 + Iy2)**2

# --- find local maxima and threshold ---
```

```
im = cv2.imread('parliament.jpg', cv2.IMREAD_GRAYSCALE)
im = im.astype(float)/255.0

sigma = 5
g = cv2.getGaussianKernel(2*sigma*3+1, sigma)

Ix = cv2.Sobel(im, -1, dx=1, dy=0)
Iy = cv2.Sobel(im, -1, dx=0, dy=1)

Ix2 = cv2.filter2D(Ix**2, -1, g, cv2.BORDER_REFLECT)
Iy2 = cv2.filter2D(Iy**2, -1, g, cv2.BORDER_REFLECT)
Ixy = cv2.filter2D(Ix*Iy, -1, g, cv2.BORDER_REFLECT)

k = 0.04  # k is between 0.04 and 0.06
# --- r = Det(M) – k·Trace(M)^2 ---
r = (Ix2*Iy2 - Ixy**2) - k*(Ix2 + Iy2)**2

# --- find local maxima and threshold ---
```

```
im = cv2.imread('parliament.jpg', cv2.IMREAD_GRAYSCALE)
im = im.astype(float)/255.0

sigma = 5
g = cv2.getGaussianKernel(2*sigma*3+1, sigma)

Ix = cv2.Sobel(im, -1, dx=1, dy
Iy = cv2.Sobel(im, -1, dx=0, dy

Ix2 = cv2.filter2D(Ix**2, -1, g, cv2.BORDER_REFLECT)
Iy2 = cv2.filter2D(Iy**2, -1, g, cv2.BORDER_REFLECT)
Ixy = cv2.filter2D(Ix*Iy, -1, g, cv2.BORDER_REFLECT)

k = 0.04  # k is between 0.04 and 0.06
# --- r = Det(M) – k·Trace(M)^2 ---
r = (Ix2*Iy2 - Ixy**2) - k*(Ix2 + Iy2)**2

# --- find local maxima and threshold ---
```

3-σ法则

```
im = cv2.imread('parliament.jpg', cv2.IMREAD_GRAYSCALE)
im = im.astype(float)/255.0

sigma = 5
g = cv2.getGaussianKernel(2*sigma*3+1, sigma)

Ix = cv2.Sobel(im, -1, dx=1, dy=0)
Iy = cv2.Sobel(im, -1, dx=0, dy=1)

Ix2 = cv2.filter2D(Ix**2, -1, g, cv2.BORDER_REFLECT)
Iy2 = cv2.filter2D(Iy**2, -1, g, cv2.BORDER_REFLECT)
Ixy = cv2.filter2D(Ix*Iy, -1, g, cv2.BORDER_REFLECT)

k = 0.04  # k is between 0.04 and 0.06
# --- r = Det(M) – k·Trace(M)^2 ---
r = (Ix2*Iy2 - Ixy**2) - k*(Ix2 + Iy2)**2

# --- find local maxima and threshold ---
```

```
im = cv2.imread('parliament.jpg', cv2.IMREAD_GRAYSCALE)
im = im.astype(float)/255.0

sigma = 5
g = cv2.getGaussianKernel(2*sigma*3+1, sigma)

Ix = cv2.Sobel(im, -1, dx=1, dy=0)
Iy = cv2.Sobel(im, -1, dx=0, dy=1)

Ix2 = cv2.filter2D(Ix**2, -1, g, cv2.BORDER_REFLECT)
Iy2 = cv2.filter2D(Iy**2, -1, g, cv2.BORDER_REFLECT)
Ixy = cv2.filter2D(Ix*Iy, -1, g, cv2.BORDER_REFLECT)

k = 0.04  # k is between 0.04 and 0.06
# --- r = Det(M) – k·Trace(M)^2 ---
r = (Ix2*Iy2 - Ixy**2) - k*(Ix2 + Iy2)**2

# --- find local maxima and threshold ---
```

局部强度变化

$$E(\Delta x, \Delta y) = (\Delta x \quad \Delta y)\mathbf{M}\begin{pmatrix}\Delta x \\ \Delta y\end{pmatrix}$$

其中

$$\mathbf{M} = \sum_{x,y} w(x,y)\begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

```
im = cv2.imread('parliament.jpg', cv2.IMREAD_GRAYSCALE)
im = im.astype(float)/255.0

sigma = 5
g = cv2.getGaussianKernel(2*sigma*3+1, sigma)

Ix = cv2.Sobel(im, -1, dx=1, dy=0)
Iy = cv2.Sobel(im, -1, dx=0, dy=1)

Ix2 = cv2.filter2D(Ix**2, -1, g, cv2.BORDER_REFLECT)
Iy2 = cv2.filter2D(Iy**2, -1, g, cv2.BORDER_REFLECT)
Ixy = cv2.filter2D(Ix*Iy, -1, g, cv2.BORDER_REFLECT)

k = 0.04  # h
r = (Ix2*Iy2 - Ixy**2) - k*(Ix2 + Iy2)**2

# --- find local maxima and threshold ---
```

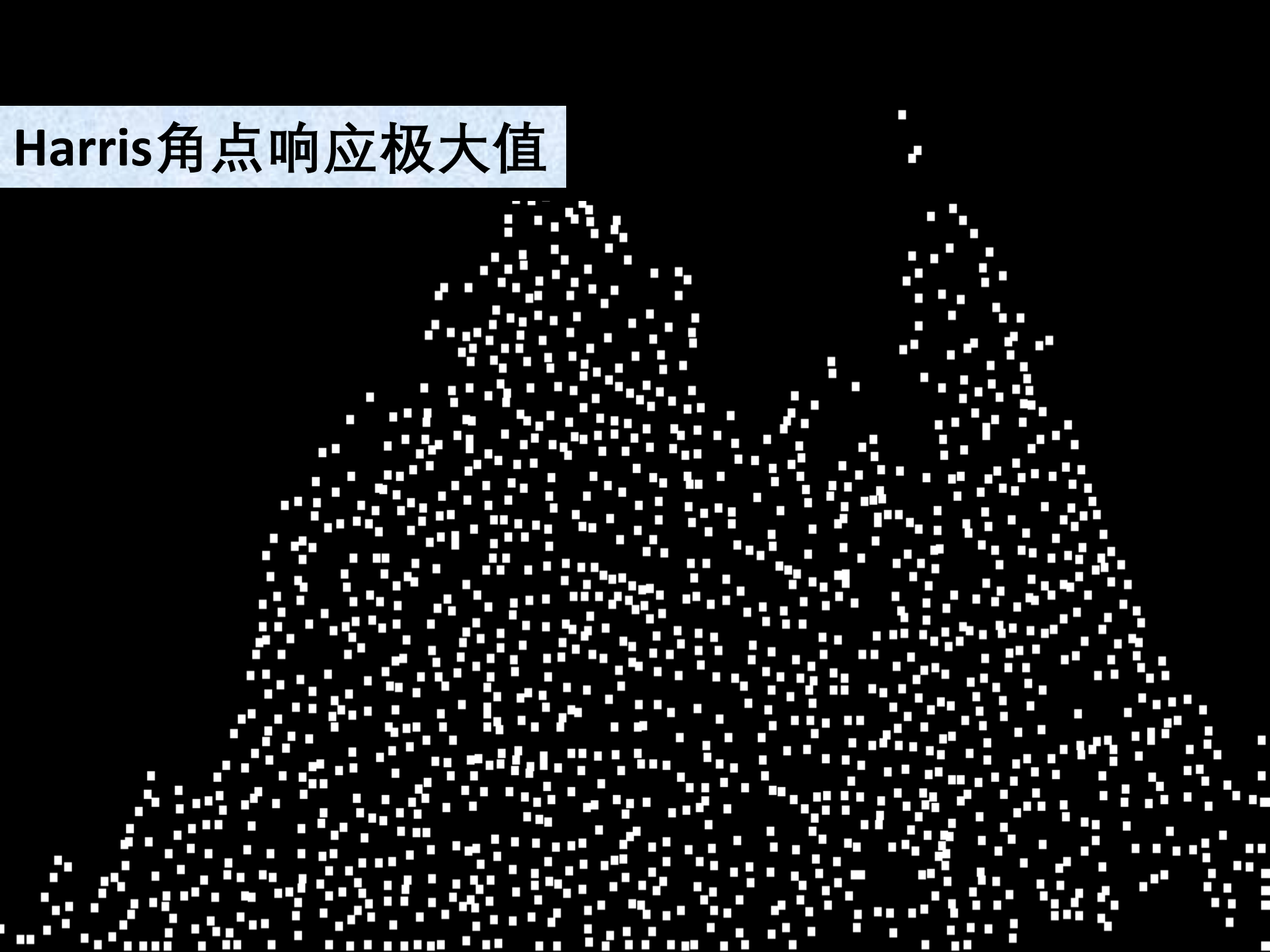
Harris矩阵元素的加权和

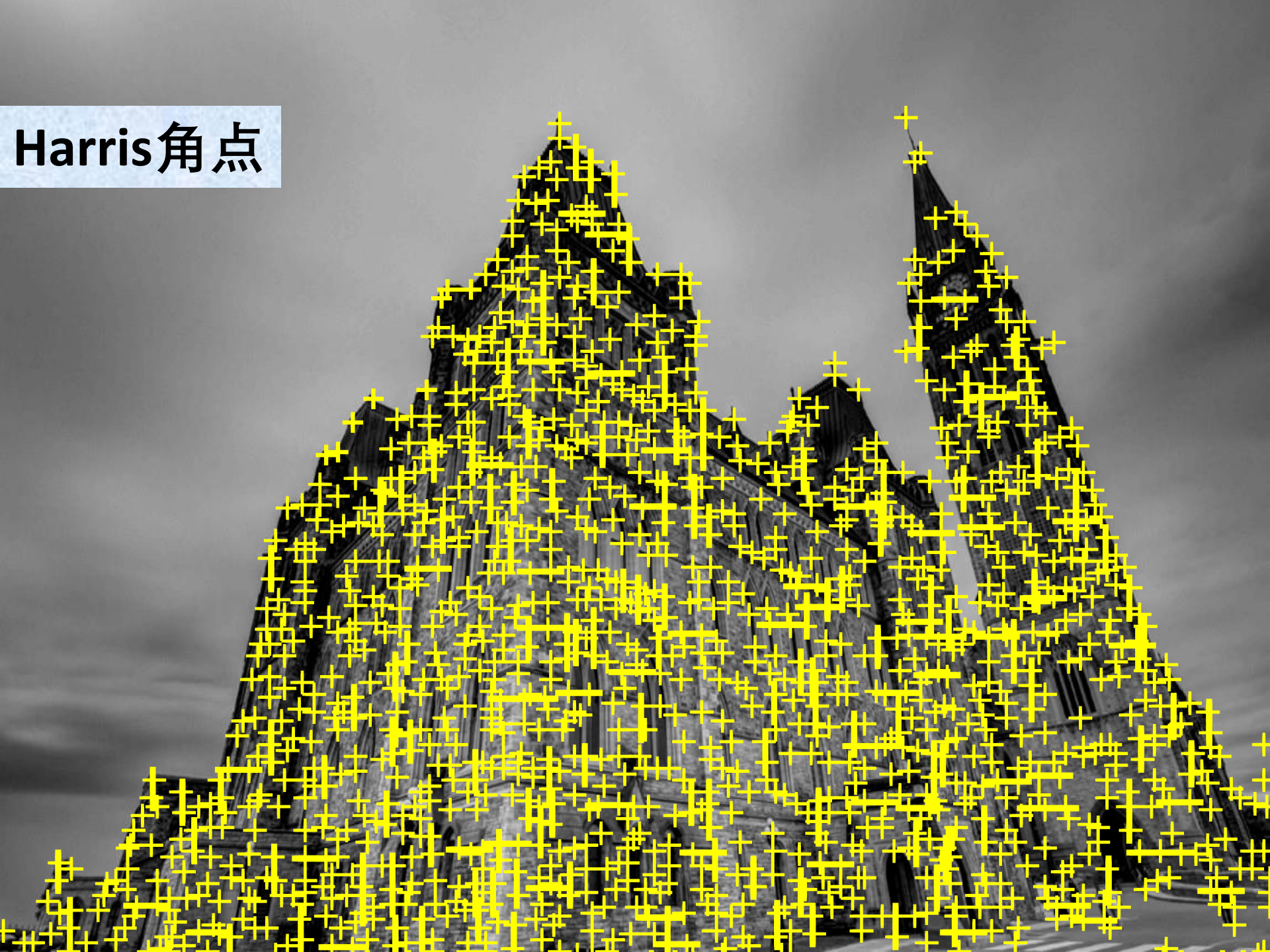
```
im = cv2.imread('parliament.jpg', cv2.IMREAD_GRAYSCALE)
im = im.astype(float)/255.0

sigma = 5
g = cv2.getGaussianKernel(2*sigma*3+1, sigma)

Ix = cv2.Sobel(im, -1, dx=1, dy=0)
Iy = cv2.Sobel(im, -1, dx=0, dy=1)

Ix2 = cv2.filter2D(Ix**2, -1, g, cv2.BORDER_REFLECT)
Iy2 = cv2.filter2D(Iy**2, -1, g, cv2.BORDER_REFLECT)
Ixy = cv2.filter2D(Ix*Iy, -1, g, cv2.BORDER_REFLECT)

k = 0.04  # k is between 0.04 and 0.06
# --- r = Det(M) – k·Trace(M)^2 ---
r = (Ix2*Iy2 - Ixy**2) - k*(Ix2 + Iy2)**2

# --- find local maxima and threshold ---
```


Harris角点响应

```
im = cv2.imread('parliament.jpg', cv2.IMREAD_GRAYSCALE)
im = im.astype(float)/255.0

sigma = 5
g = cv2.getGaussianKernel(2*sigma*3+1, sigma)

Ix = cv2.Sobel(im, -1, dx=1, dy=0)
Iy = cv2.Sobel(im, -1, dx=0, dy=1)

Ix2 = cv2.filter2D(Ix**2, -1, g, cv2.BORDER_REFLECT)
Iy2 = cv2.filter2D(Iy**2, -1, g, cv2.BORDER_REFLECT)
Ixy = cv2.filter2D(Ix*Iy, -1, g, cv2.BORDER_REFLECT)

k = 0.04  # k is between 0.04 and 0.06
# --- r = Det(M) – k·Trace(M)^2 ---
r = (Ix2*Iy2 - Ixy**2) - k*(Ix2 + Iy2)**2

# --- find local maxima and threshold ---
```

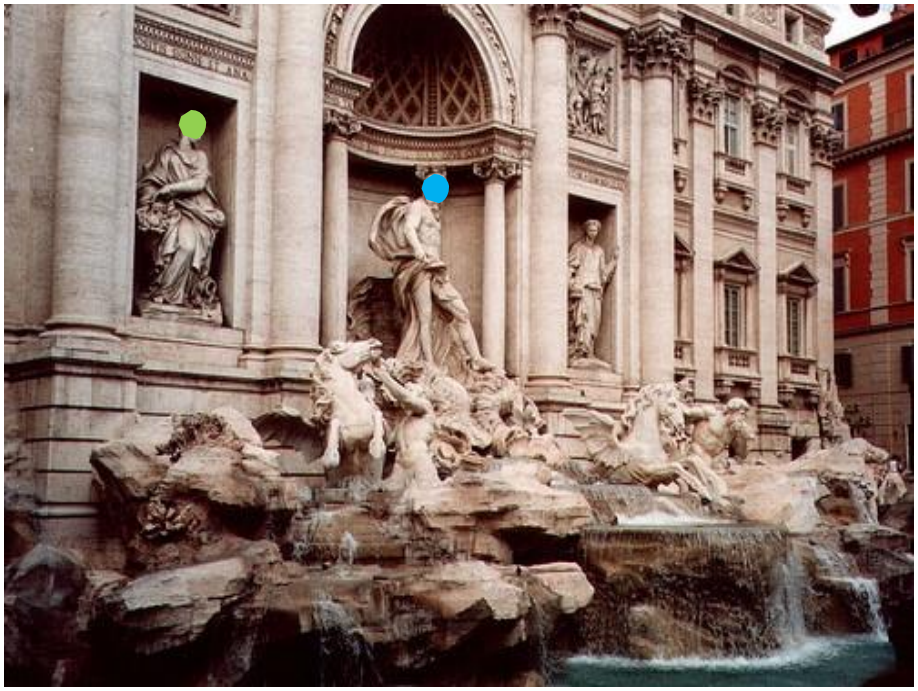
Harris角点响应极大值



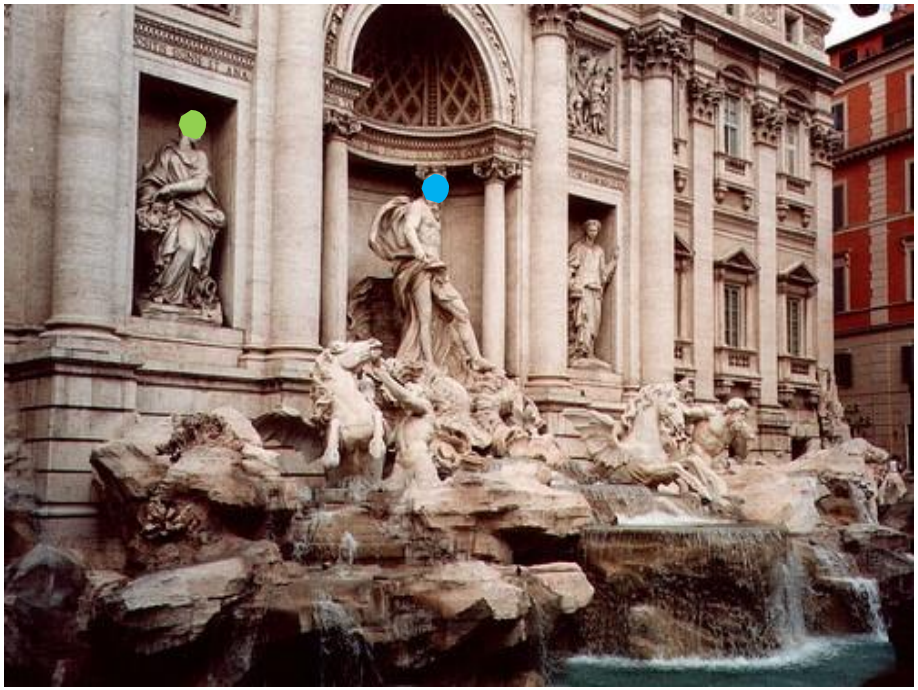

Harris角点

Python时间

左图中的哪些内容与右图内容匹配?



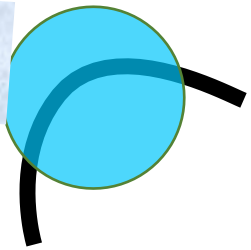
左图中的哪些内容与右图内容匹配?



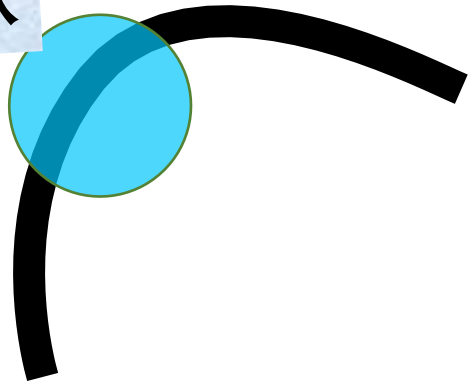
如何描述特征？

如何描述特征？

角点



边缘



如何应对尺度变化?

SIFT

Scale-Invariant Feature Transform

尺度不变特征变换

David Lowe, "Distinctive image features from scale-invariant keypoints", IJCV, 2004.

SIFT

Scale-Invariant Feature Transform

尺度不变特征变换

感兴趣点检测器+描述子

David Lowe, "Distinctive image features from scale-invariant keypoints", IJCV, 2004.

Object Recognition from Local Scale-Invariant Features

David G. Lowe

Computer Science Department
University of British Columbia
Vancouver, B.C., V6T 1Z4, Canada
lowe@cs.ubc.ca

Abstract

An object recognition system has been developed that uses a new class of local image features. The features are invariant to image scaling, translation, and rotation, and partially invariant to illumination changes and affine or 3D projection. These features share similar properties with neurons in in-

translation, scaling, and rotation, and partially invariant to illumination changes and affine or 3D projection. Previous approaches to local feature generation lacked invariance to scale and were more sensitive to projective distortion and illumination change. The SIFT features share a number of properties in common with the responses of neurons in infe-

Object Recognition from Local Scale-Invariant Features

David G. Lowe

Computer Science Department
University of British Columbia
Vancouver, B.C., V6T 1Z4, Canada
lowe@cs.ubc.ca

Abstract

An object recognition system has been developed that uses a new class of local image features. The features are invariant to image scaling, translation, and rotation, and partially invariant to illumination changes and affine or 3D projection. These features share similar properties with neurons in in-

translation, scaling, and rotation, and partially invariant to illumination changes and affine or 3D projection. Previous approaches to local feature generation lacked invariance to scale and were more sensitive to projective distortion and illumination change. The SIFT features share a number of properties in common with the responses of neurons in infe-

"I did submit papers on earlier versions of SIFT to both ICCV and CVPR (around 1997/98) and both were rejected. I then added more of a systems flavor and the paper was published at ICCV 1999, but just as a poster."

— David Lowe

Object Recognition from Local Scale-Invariant Features

David G. Lowe

Computer Science Department
University of British Columbia
Vancouver, B.C., V6T 1Z4, Canada
lowe@cs.ubc.ca

Abstract

An object recognition system has been developed that uses a new class of local image features. The features are invariant to image scaling, translation, and rotation, and partially invariant to illumination changes and affine or 3D projection. These features share similar properties with neurons in in-

translation, scaling, and rotation, and partially invariant to illumination changes and affine or 3D projection. Previous approaches to local feature generation lacked invariance to scale and were more sensitive to projective distortion and illumination change. The SIFT features share a number of properties in common with the responses of neurons in infe-

引用超过2万次！

International Journal of Computer Vision 60(2), 91–110, 2004

Distinctive Image Features from Scale-Invariant Keypoints

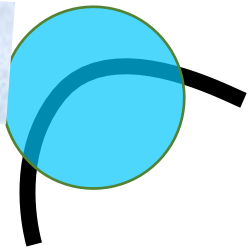
DAVID G. LOWE
Computer Science Department, University of British Columbia, Vancouver, B.C., Canada
Lowe@cs.ubc.ca

Received January 10, 2003; Revised January 7, 2004; Accepted January 22, 2004

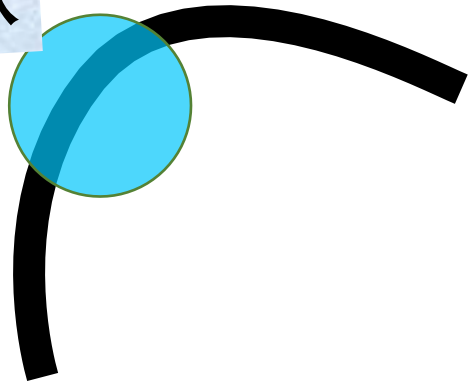
Abstract. This paper presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. The features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. The features are highly distinctive, in the sense that a

引用超过7万次！

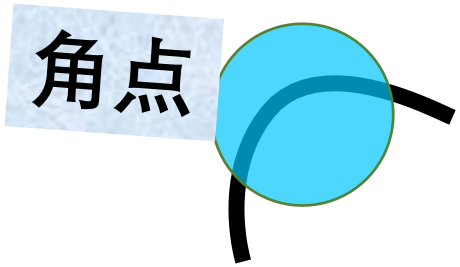
角点

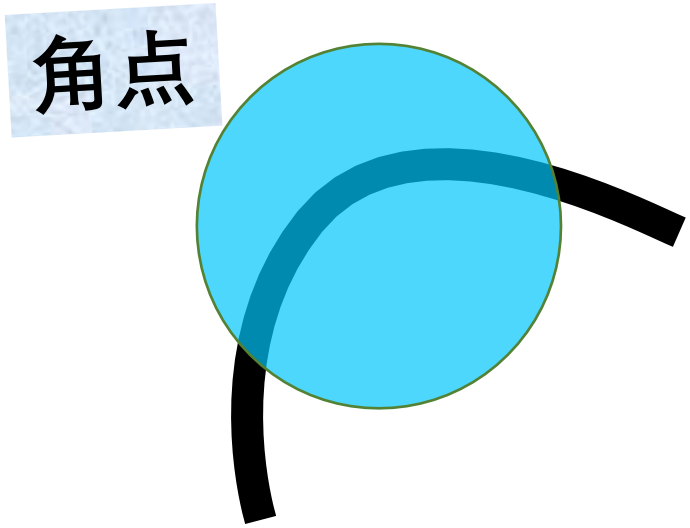


边缘

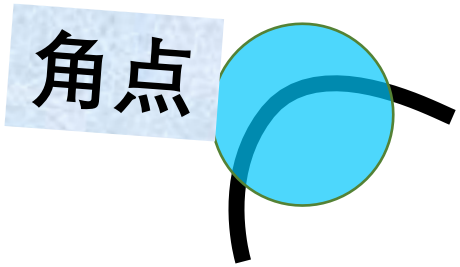


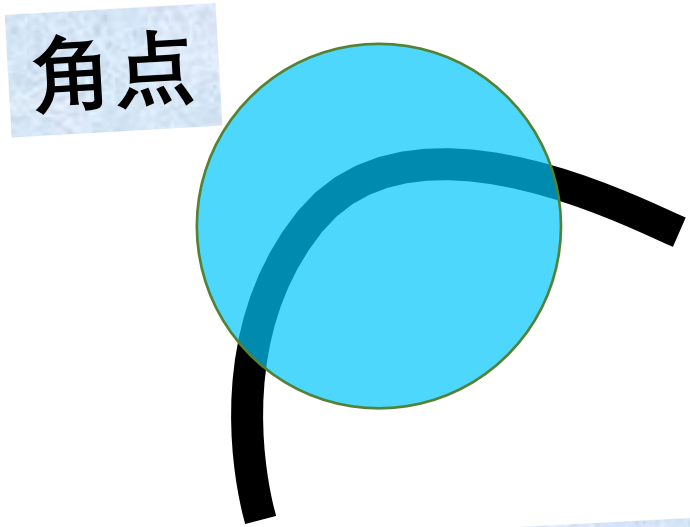
如何应对尺度变化?



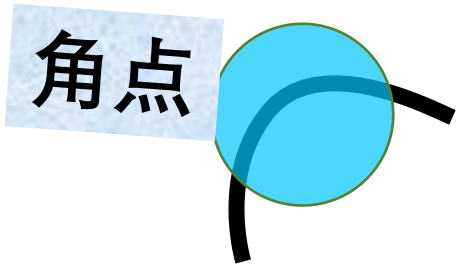


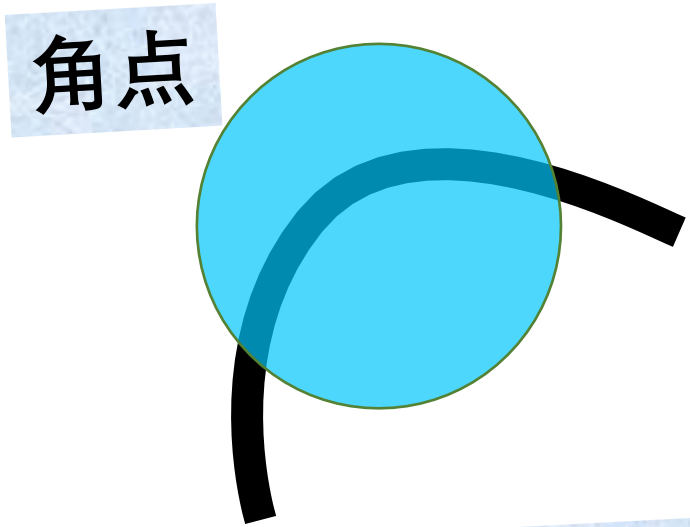
如何应对尺度变化?





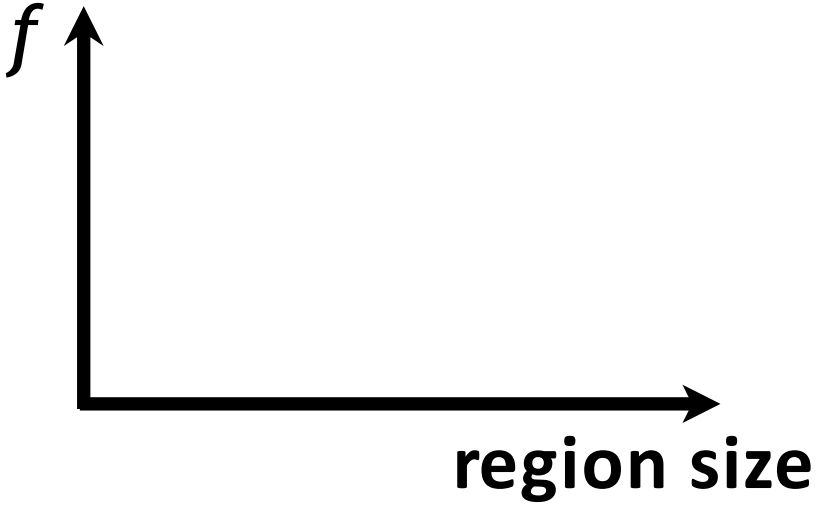
如何在每个图像中独立地
选择正确的区域大小？

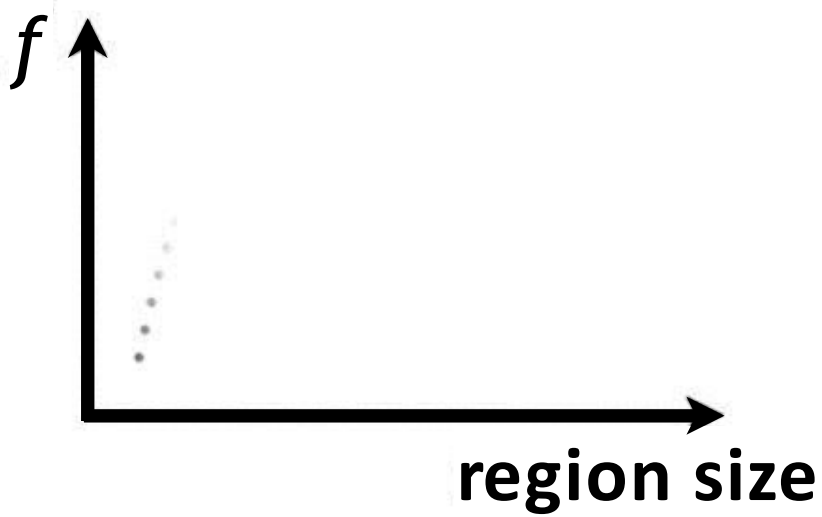


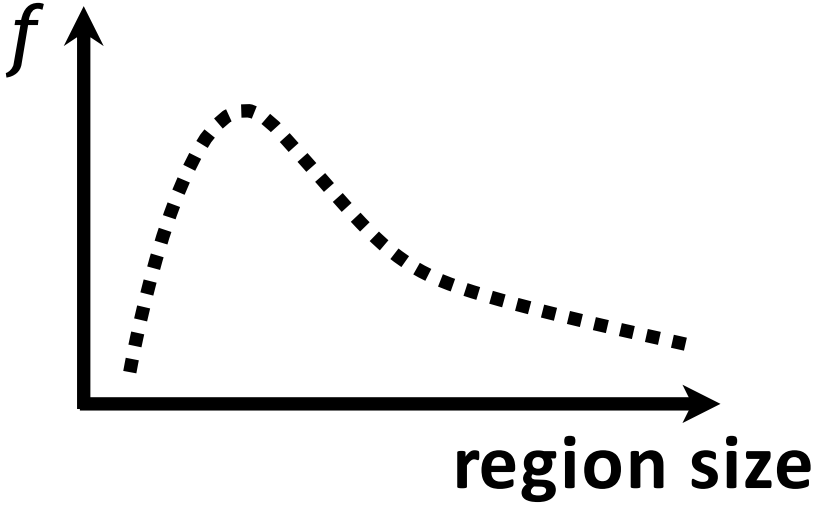


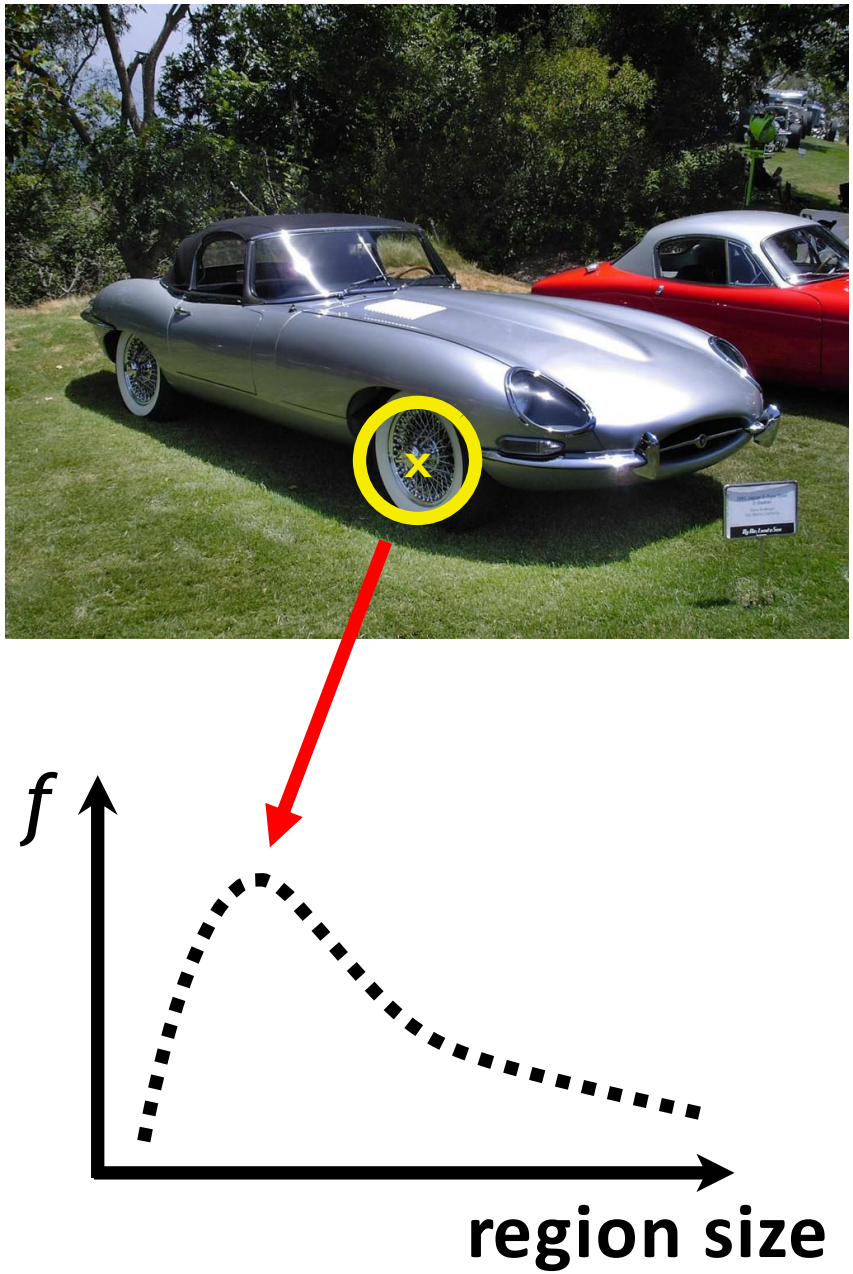
如何在每个图像中独立地
选择正确的区域大小？

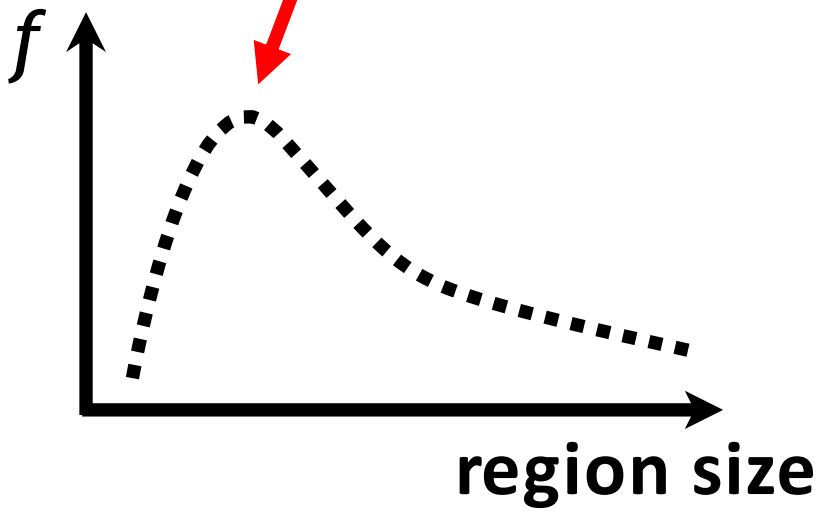
在该区域设计一个尺度不变的函数

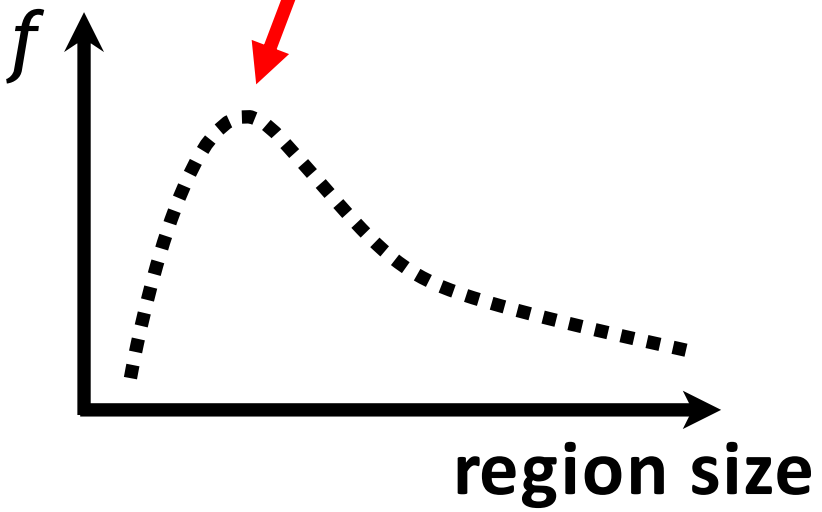

f
region size

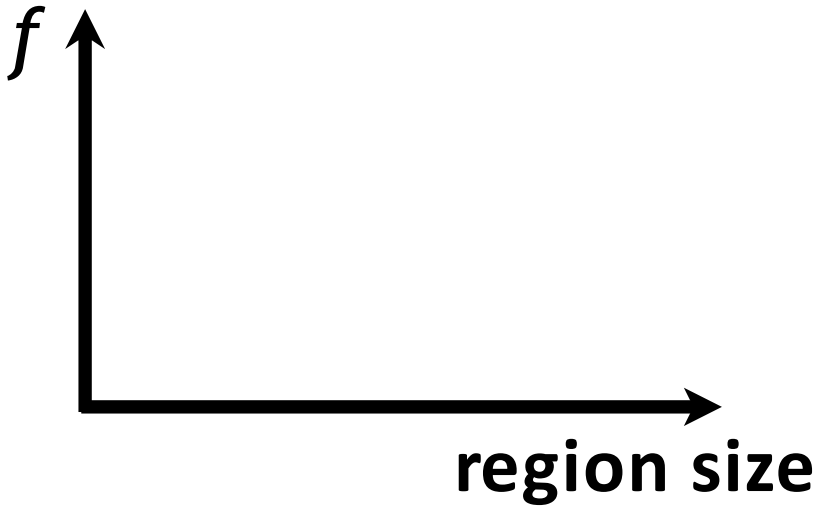

f
region size

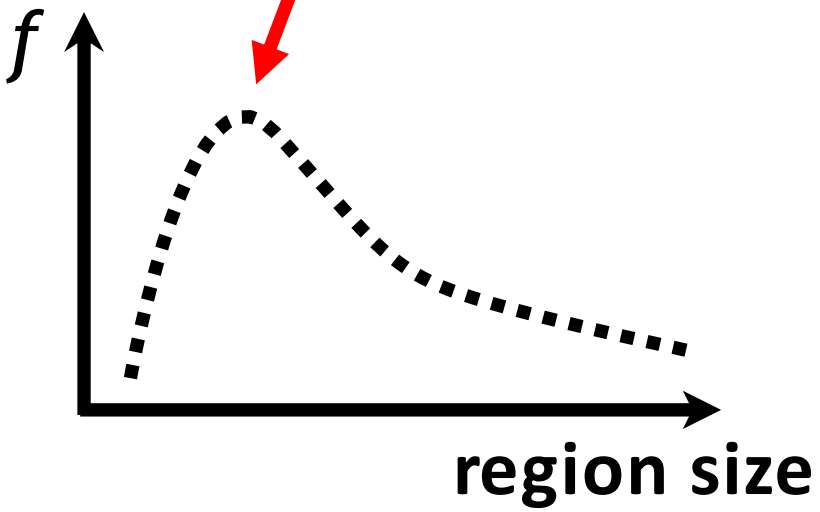

f
region size

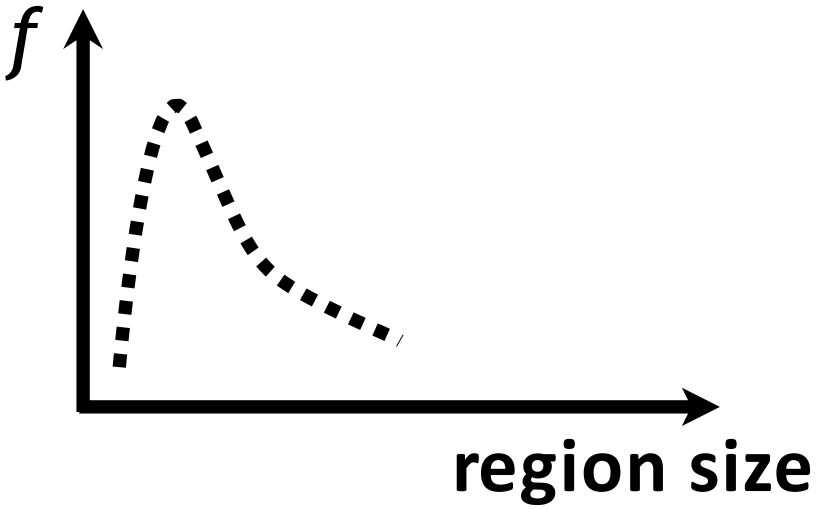

f
region size

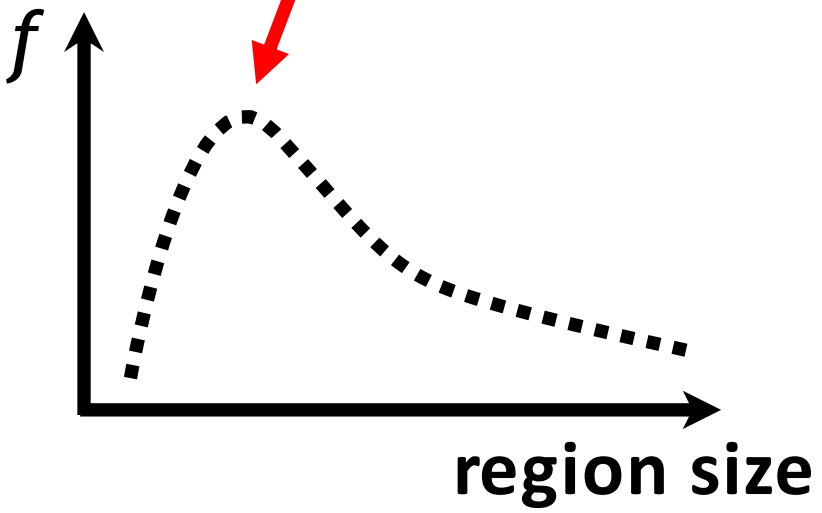

f
region size

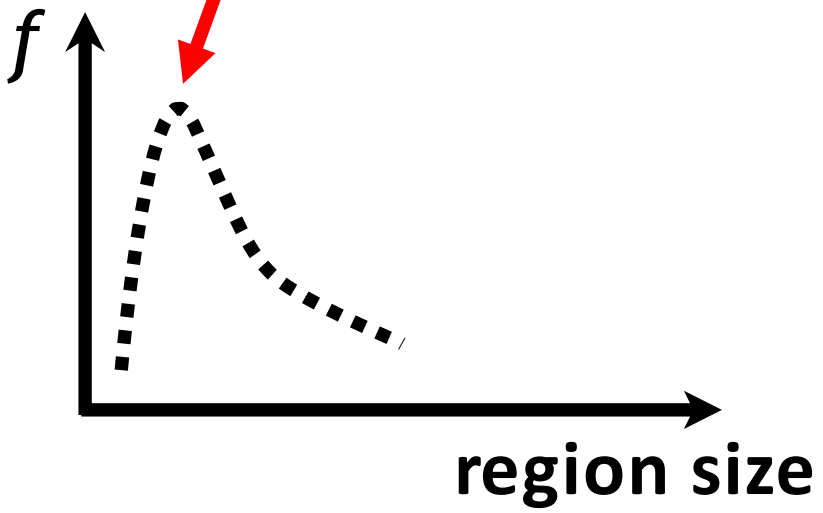

f
region size


f
region size


f
region size


f
region size


f
region size


f
region size

尺度不变描述

图像1

图像2

尺度不变描述

图像1

图像2

将关键点图像重采样为标准大小

尺度不变描述

图像1

图像2

将关键点图像重采样为标准大小

尺度不变描述

图像1

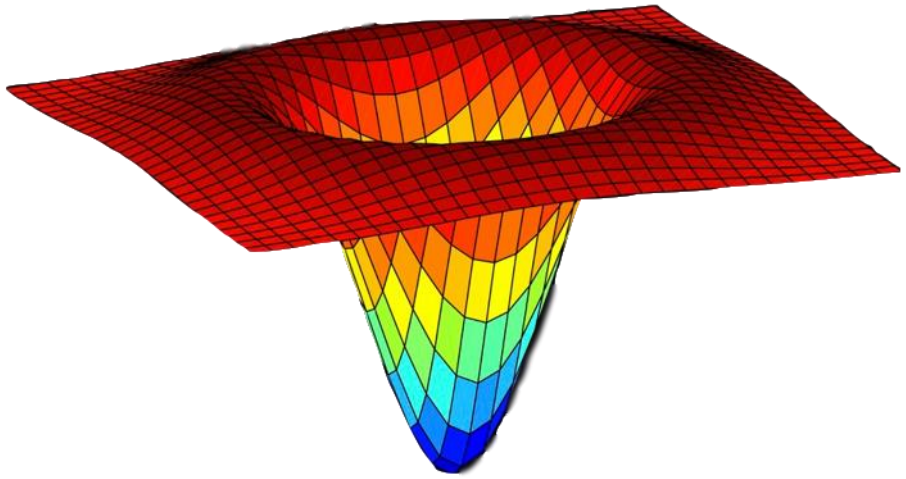
图像2

将关键点图像重采样为标准大小

哪个特征检测器?

哪个特征检测器?

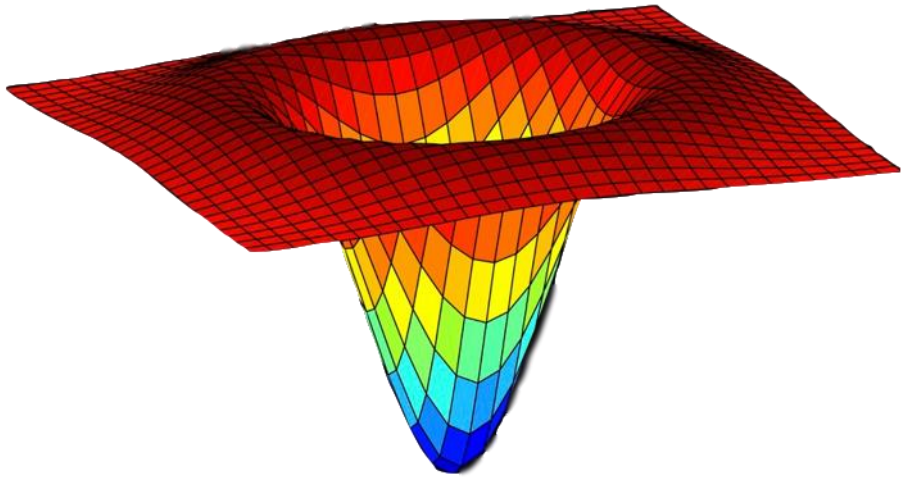
$$f = \text{Kernal} * \text{Image}$$



Laplacian of Gaussian (LoG)

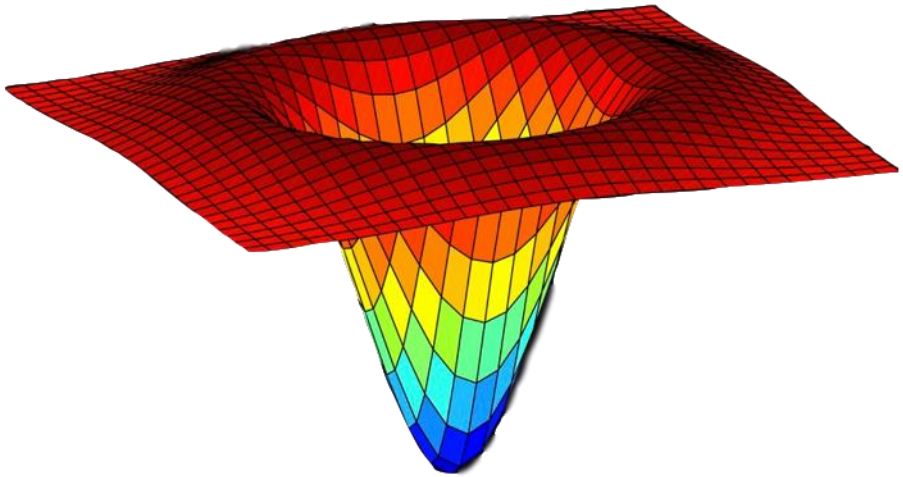
高斯-拉普拉斯算子

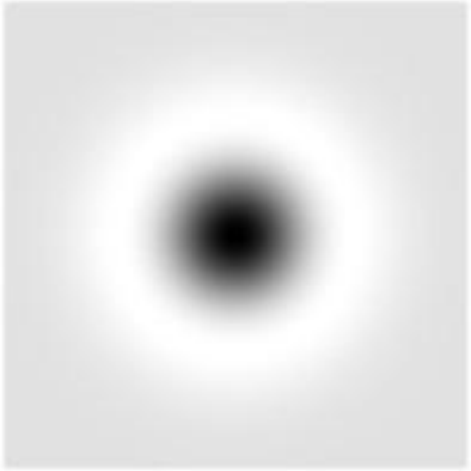
墨西哥帽算子



Laplacian of Gaussian (LoG)

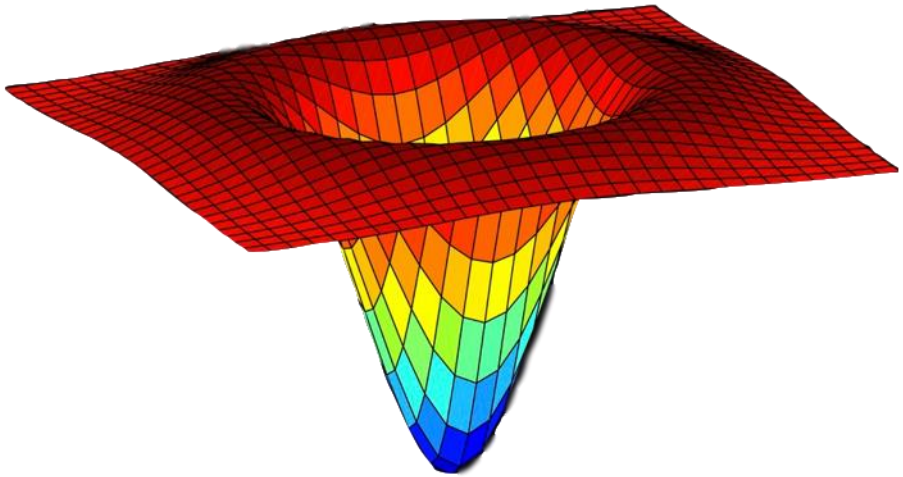
高斯-拉普拉斯算子

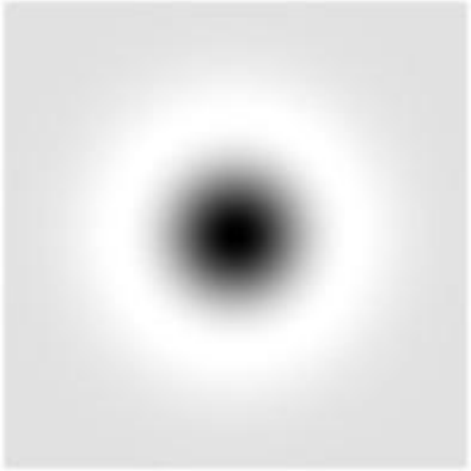




Laplacian of Gaussian (LoG)

高斯-拉普拉斯算子

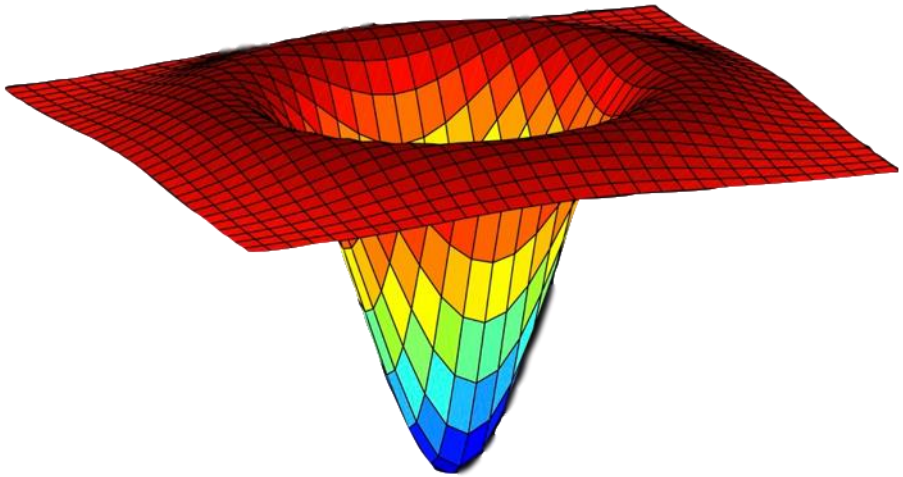


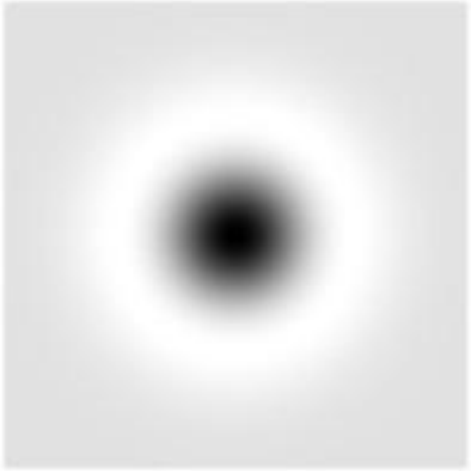


Laplacian of Gaussian (LoG)

高斯-拉普拉斯算子

该滤波器能检测什么图案?



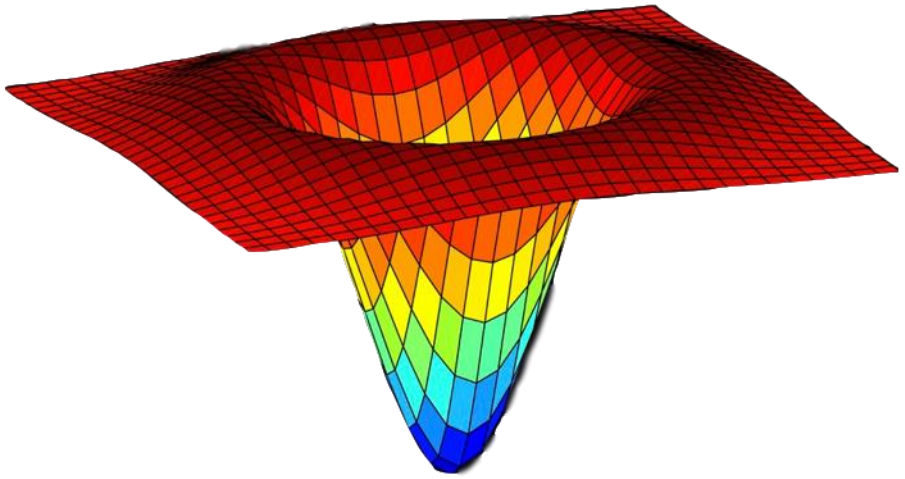


Laplacian of Gaussian (LoG)

高斯-拉普拉斯算子

该滤波器能检测什么图案？

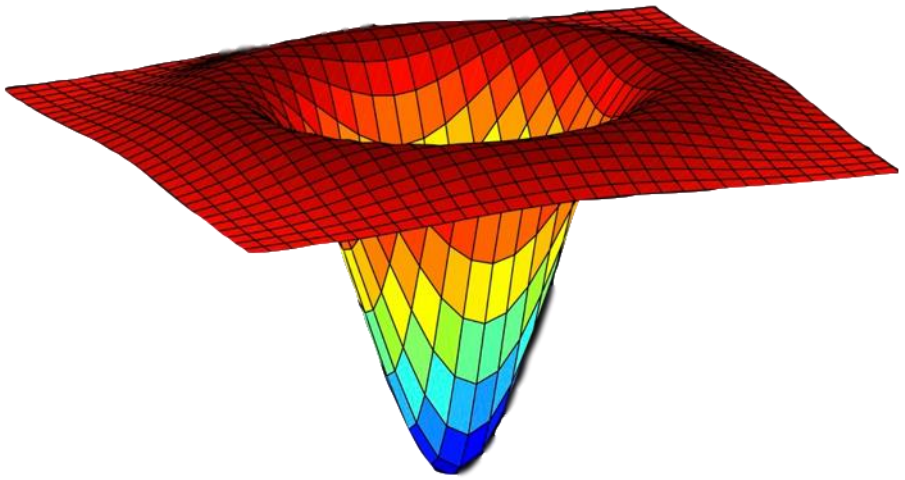
斑点



Laplacian of Gaussian (LoG)

高斯-拉普拉斯算子

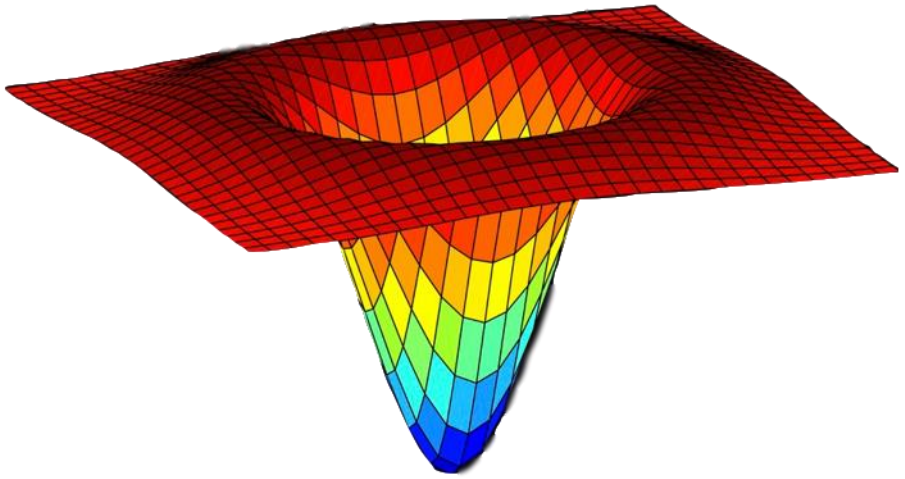
$$L(x, y; \sigma) = \sigma^2 \left(G_{xx}(x, y; \sigma) + G_{yy}(x, y; \sigma) \right)$$



Laplacian of Gaussian (LoG)

高斯-拉普拉斯算子

$$L(x, y; \sigma) = \sigma^2 \left(G_{xx}(x, y; \sigma) + G_{yy}(x, y; \sigma) \right)$$



Laplacian of Gaussian (LoG)

高斯-拉普拉斯算子

$$L(x, y; \sigma) = \sigma^2 \left(G_{xx}(x, y; \sigma) + G_{yy}(x, y; \sigma) \right)$$

$$f_{xx} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right) = \frac{\partial^2 f}{\partial x^2}$$

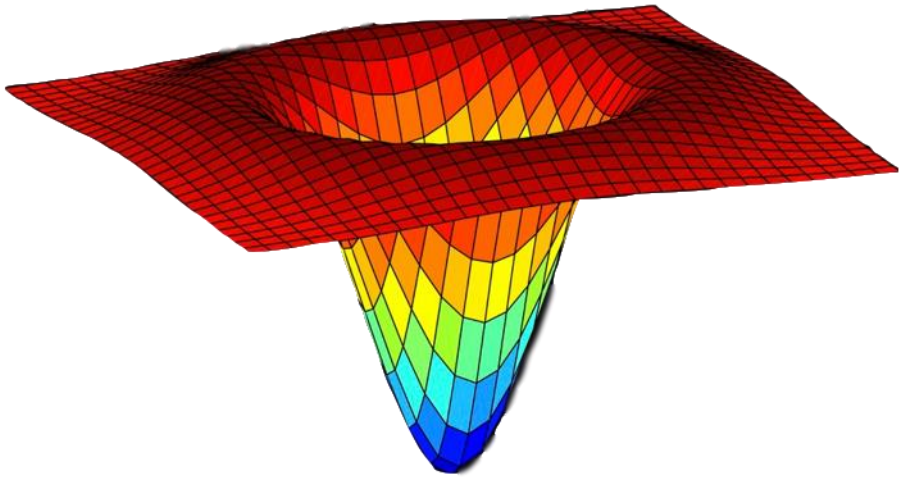
$$f_{xx} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right) = \frac{\partial^2 f}{\partial x^2}$$

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$f_{xx} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right) = \frac{\partial^2 f}{\partial x^2}$$

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

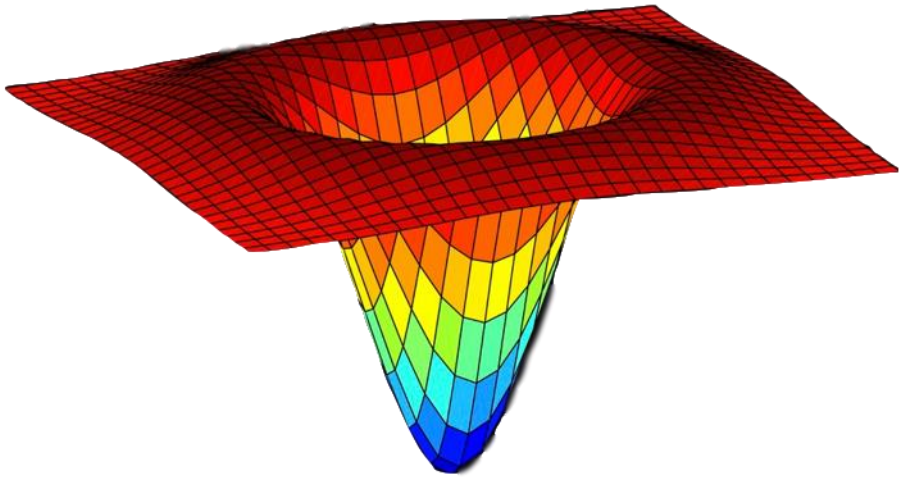
拉普拉斯算子



Laplacian of Gaussian (LoG)

高斯-拉普拉斯算子

$$L(x, y; \sigma) = \sigma^2 \left(G_{xx}(x, y; \sigma) + G_{yy}(x, y; \sigma) \right)$$

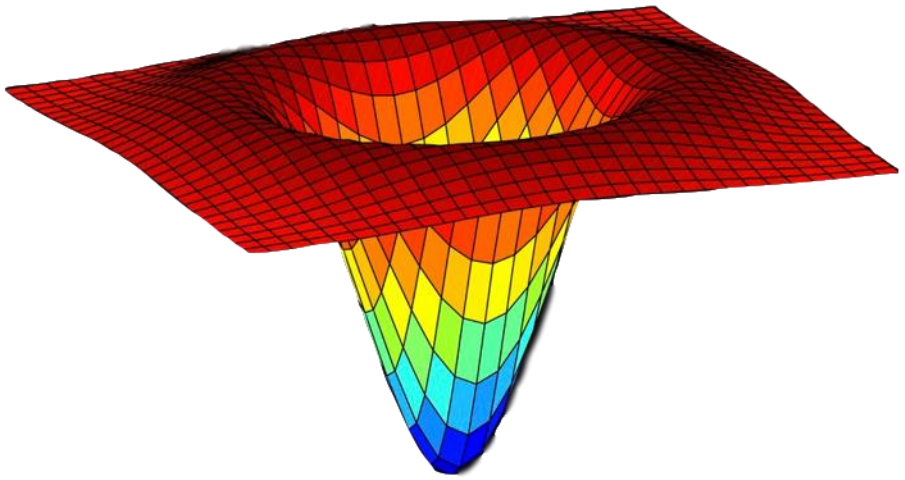


Laplacian of Gaussian (LoG)

高斯-拉普拉斯算子

$$L(x, y; \sigma) = \sigma^2 \left(G_{xx}(x, y; \sigma) + G_{yy}(x, y; \sigma) \right)$$

$$\approx G(x, y; k\sigma) - G(x, y; \sigma)$$



Laplacian of Gaussian (LoG)

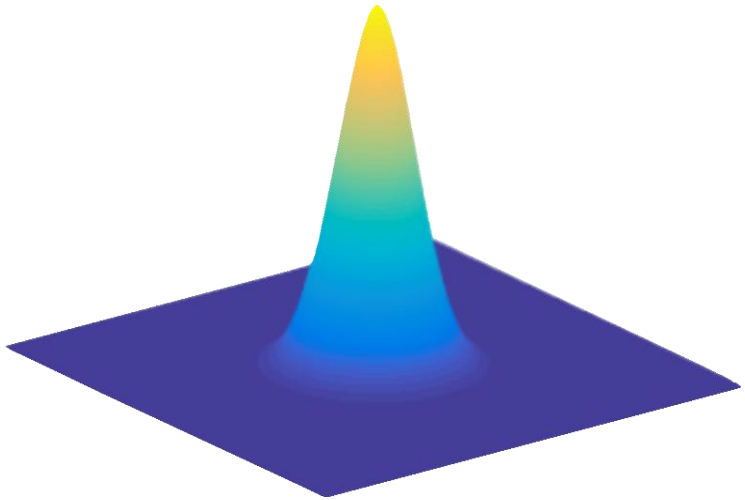
高斯-拉普拉斯算子

$$L(x, y; \sigma) = \sigma^2 \left(G_{xx}(x, y; \sigma) + G_{yy}(x, y; \sigma) \right)$$

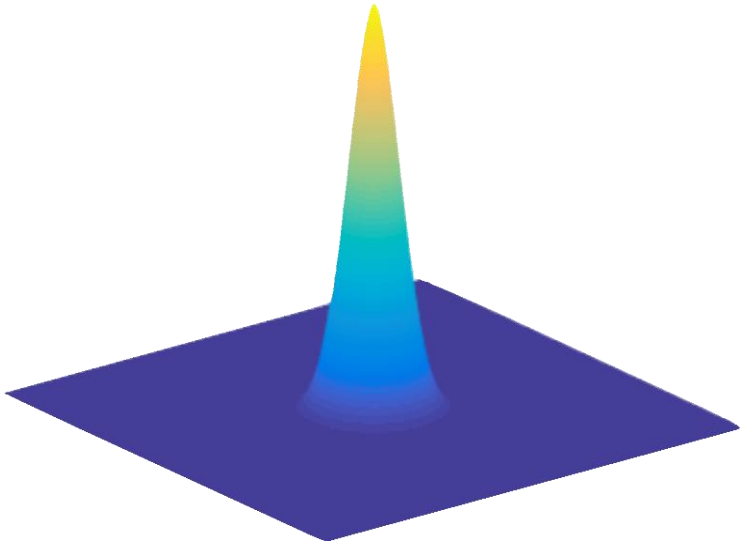
$$\approx G(x, y; k\sigma) - G(x, y; \sigma)$$

Difference of Gaussians (DoG)

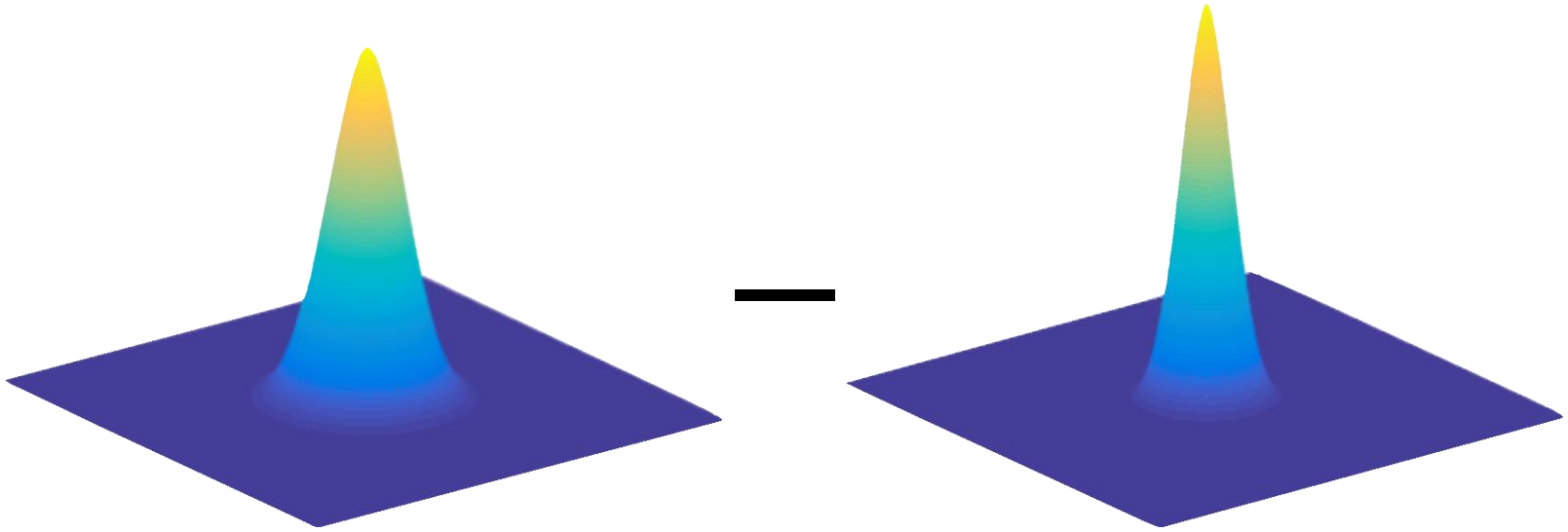
高斯差算子



σ =1.2

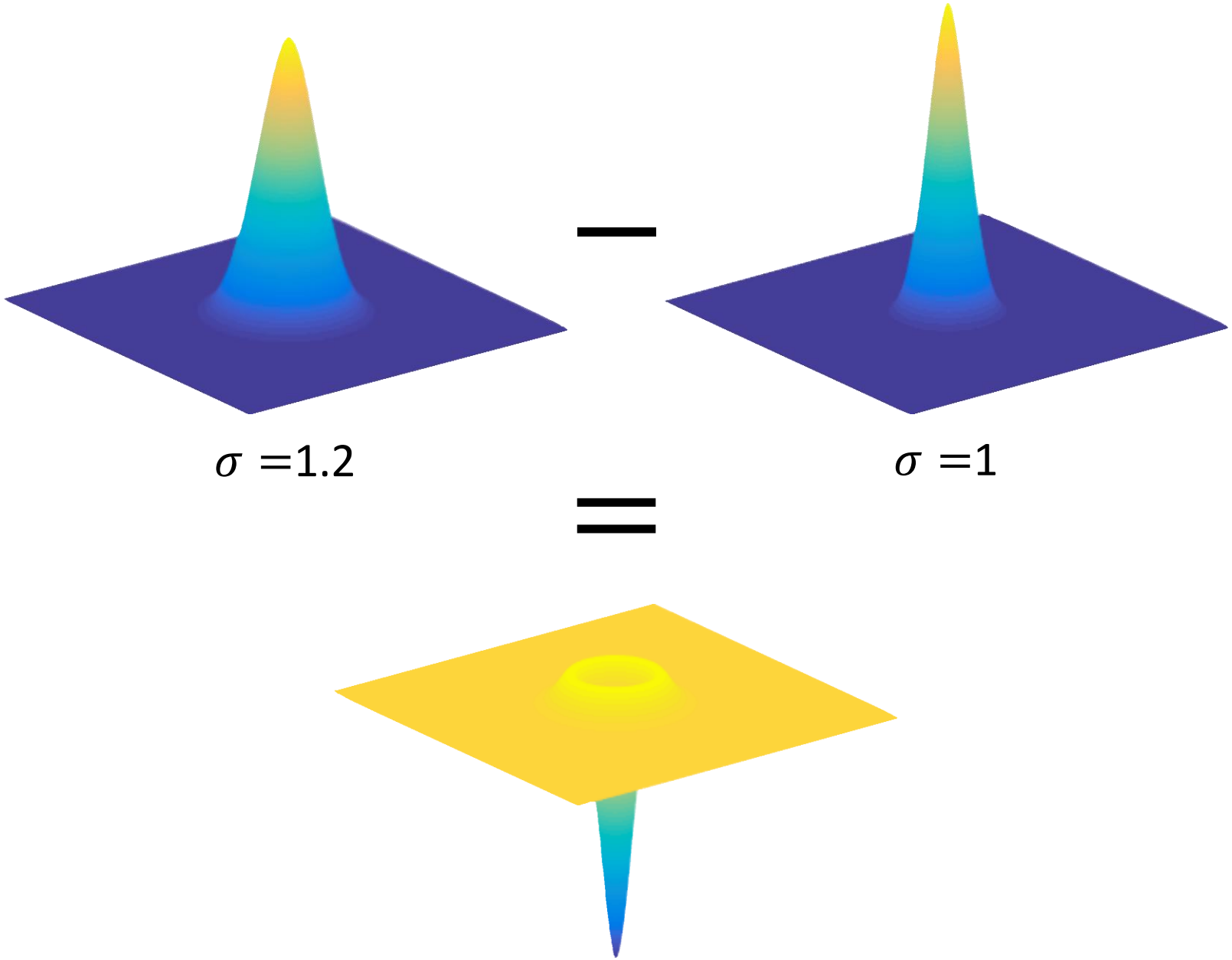


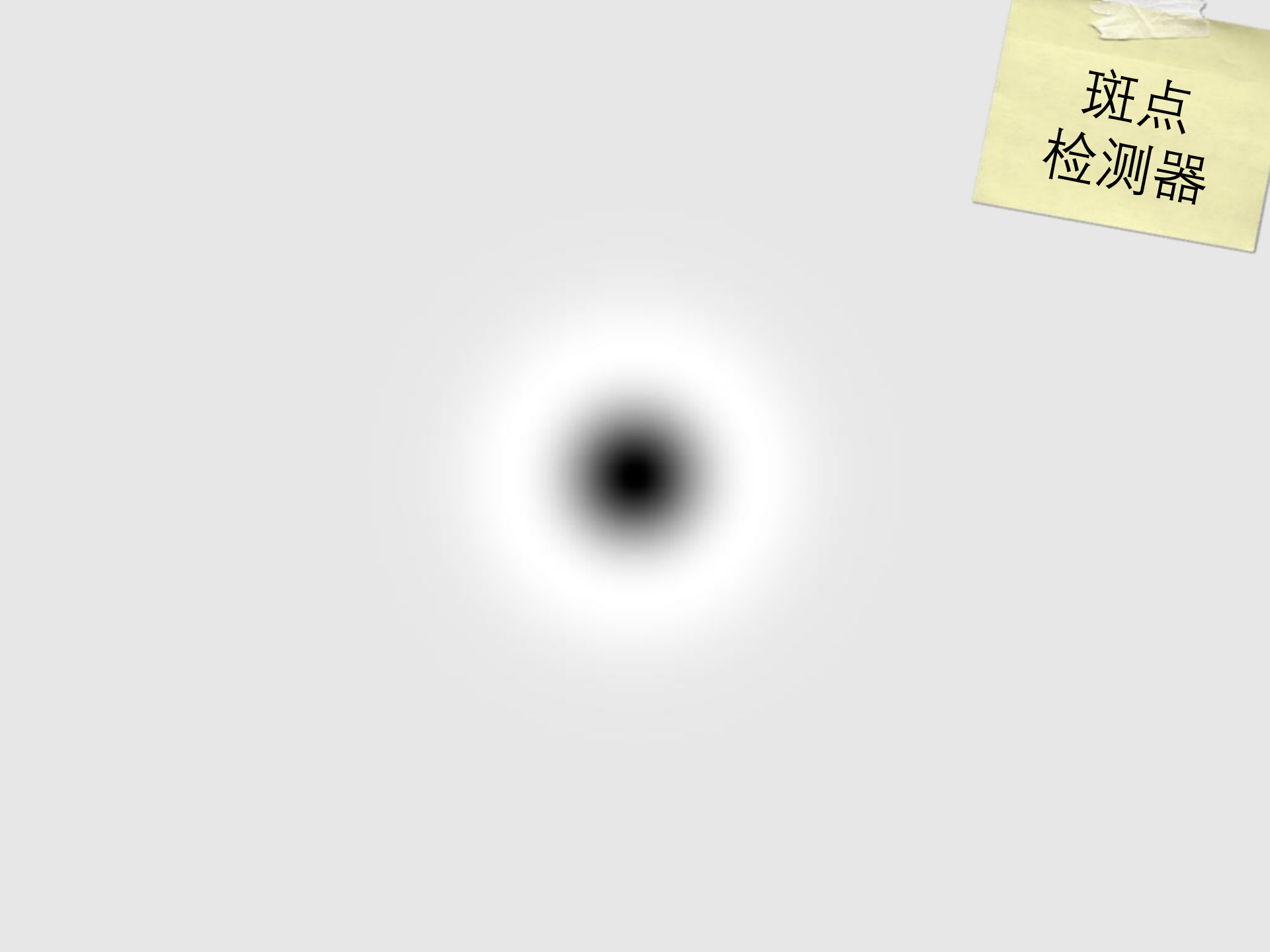
σ =1



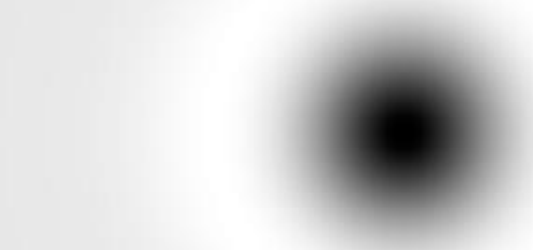
σ =1.2

σ =1

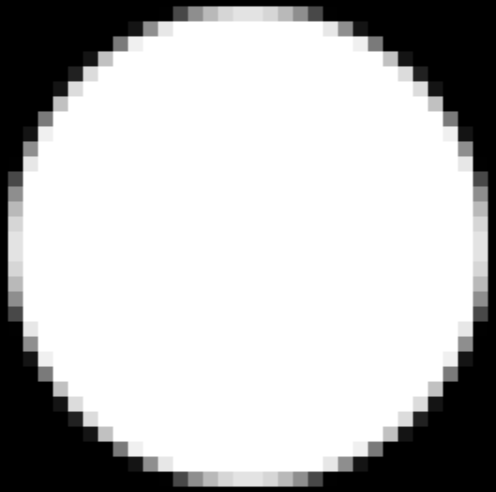

−
σ =1.2
σ =1
=


斑点
检测器

斑点
检测器



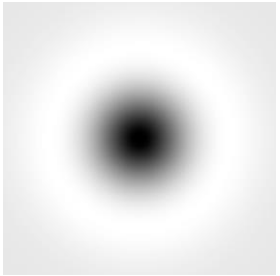
使用多个尺度的“斑点”滤波器与图像卷积
并找到极值

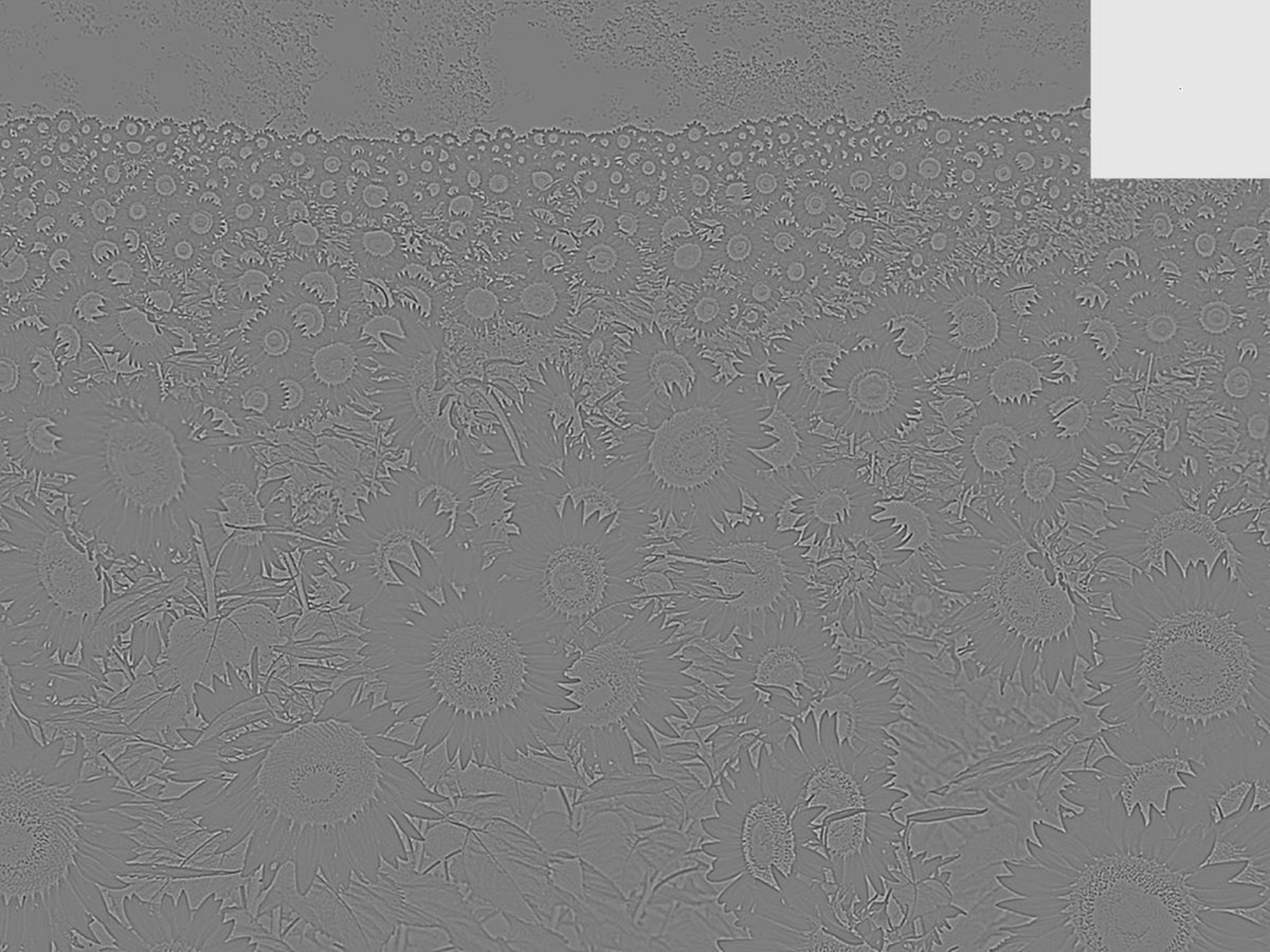


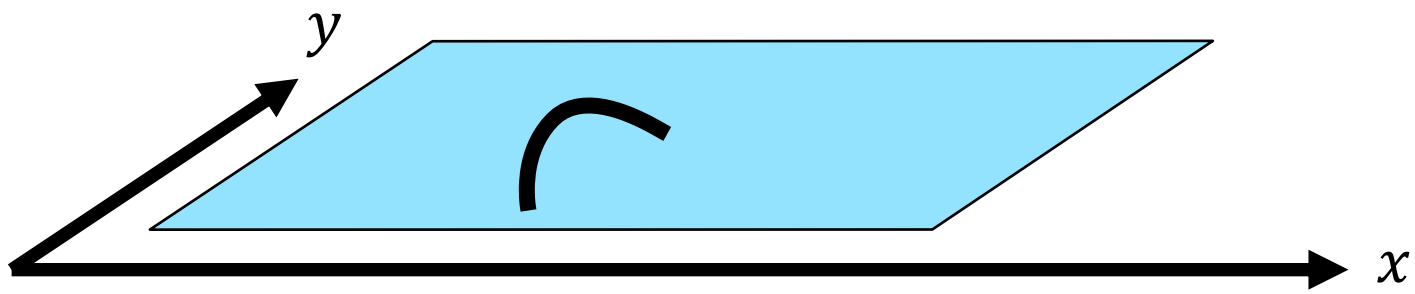
拉普拉斯极值响应位于$\sigma = \text{radius}/\sqrt{2}$

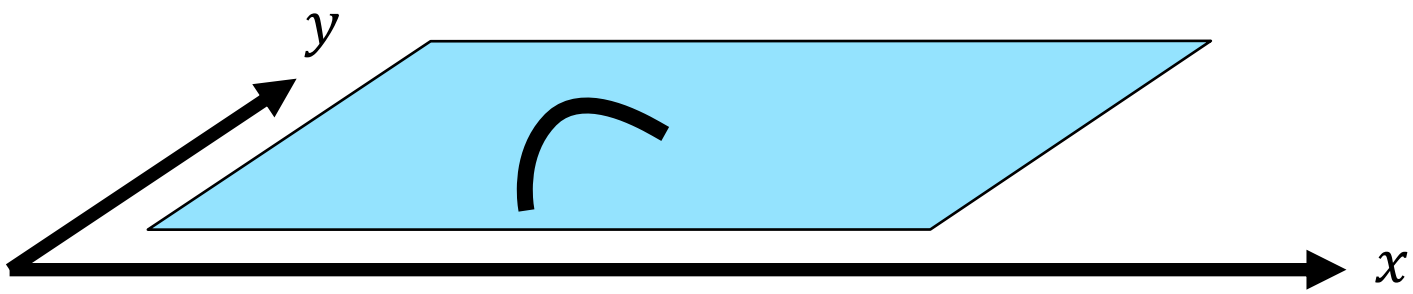
拉普拉斯极值响应位于$\sigma = \text{radius}/\sqrt{2}$

*

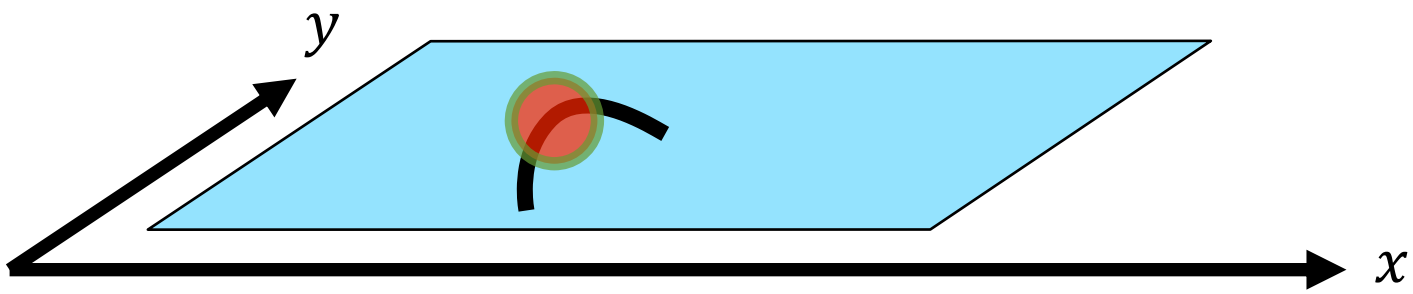




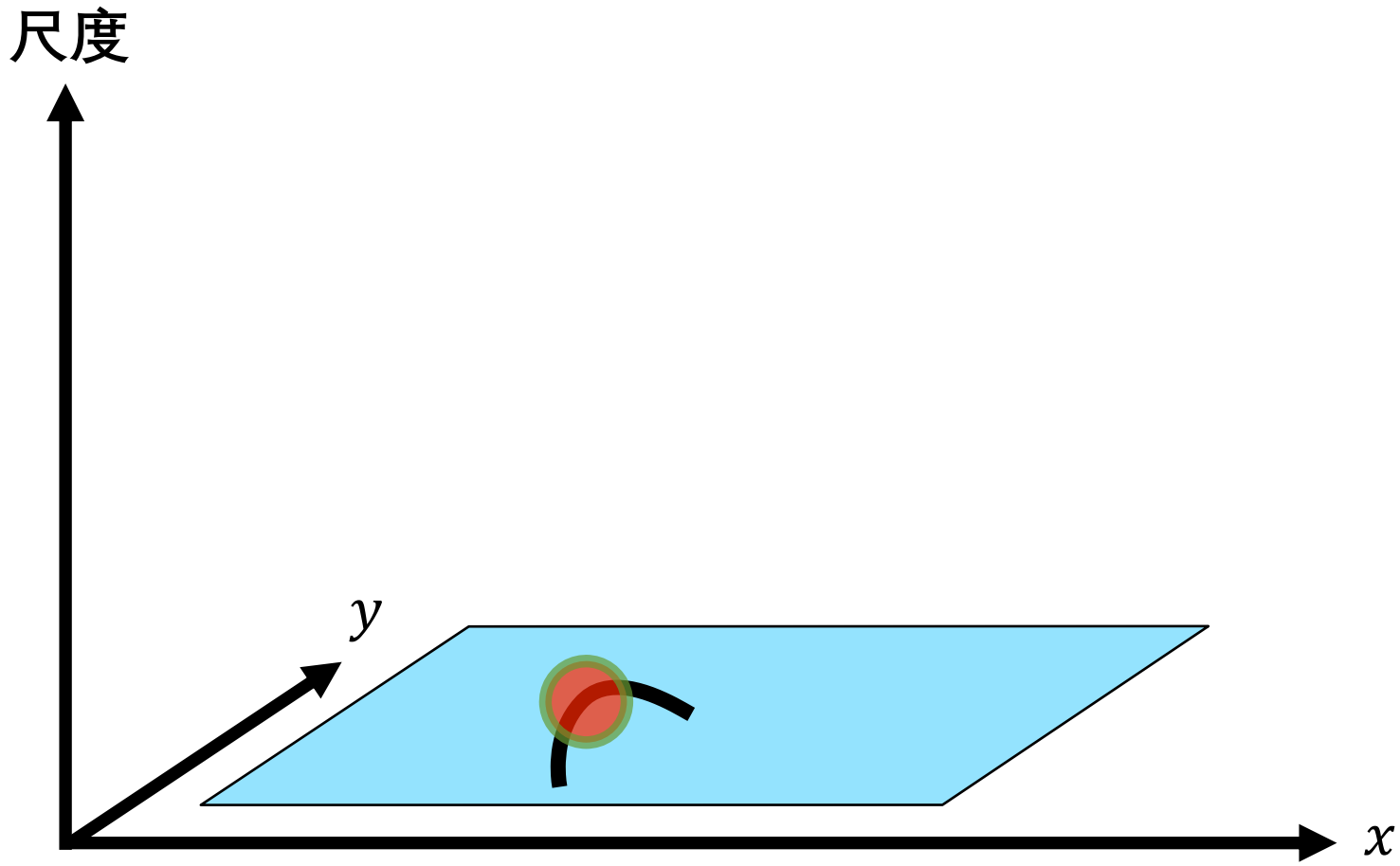

y
x



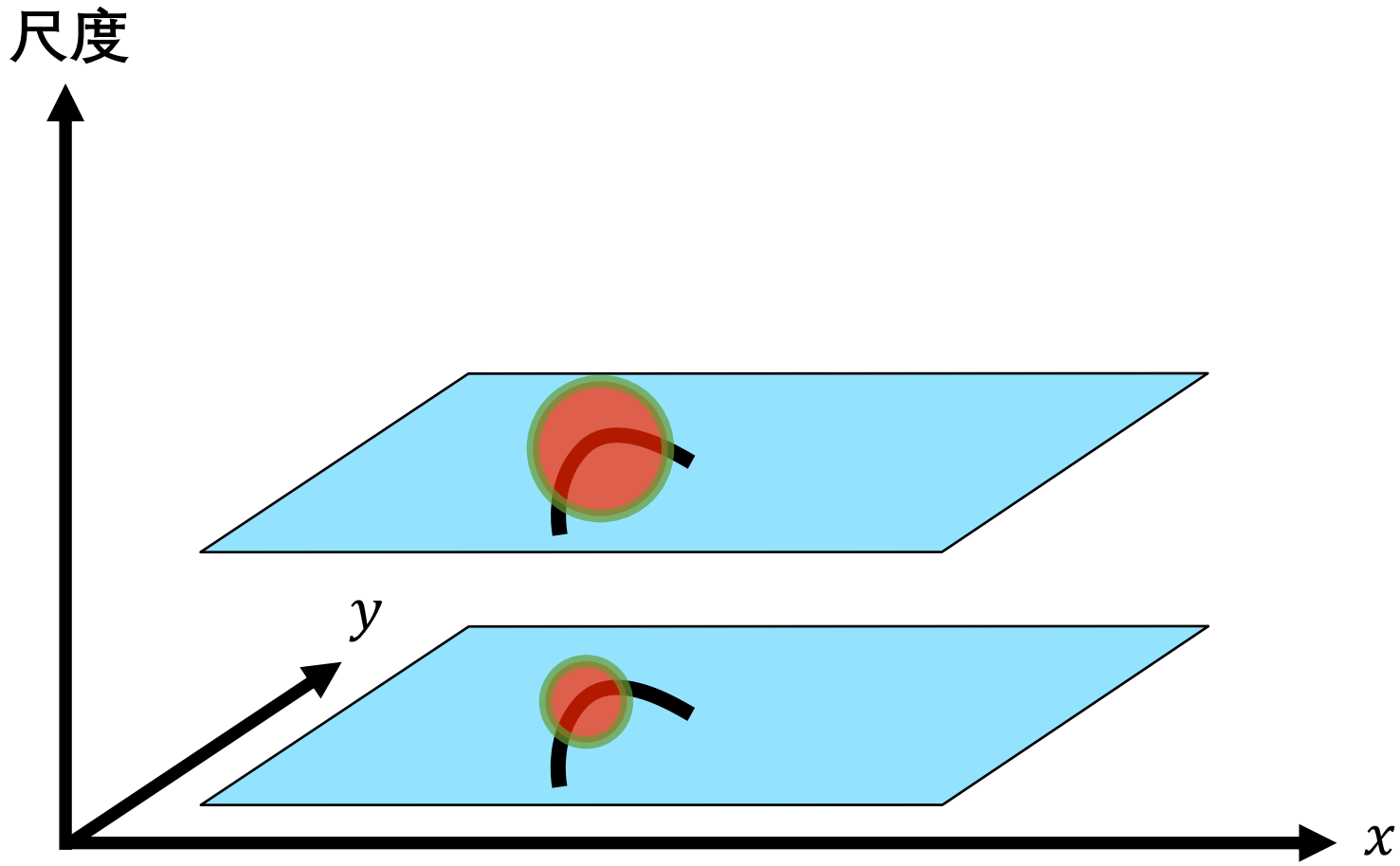
跨越多个尺度空间与DoG卷积



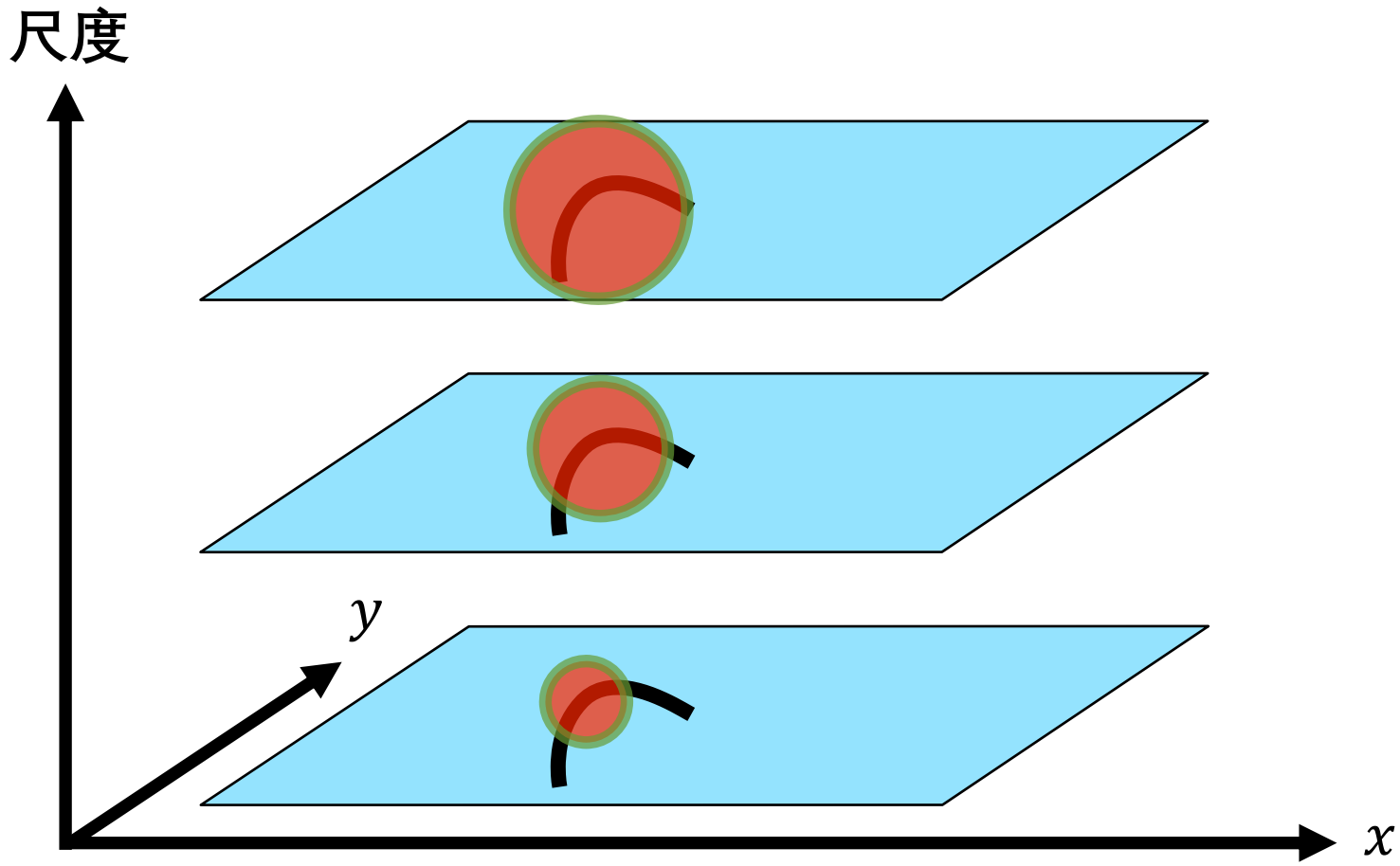
跨越多个尺度空间与DoG卷积



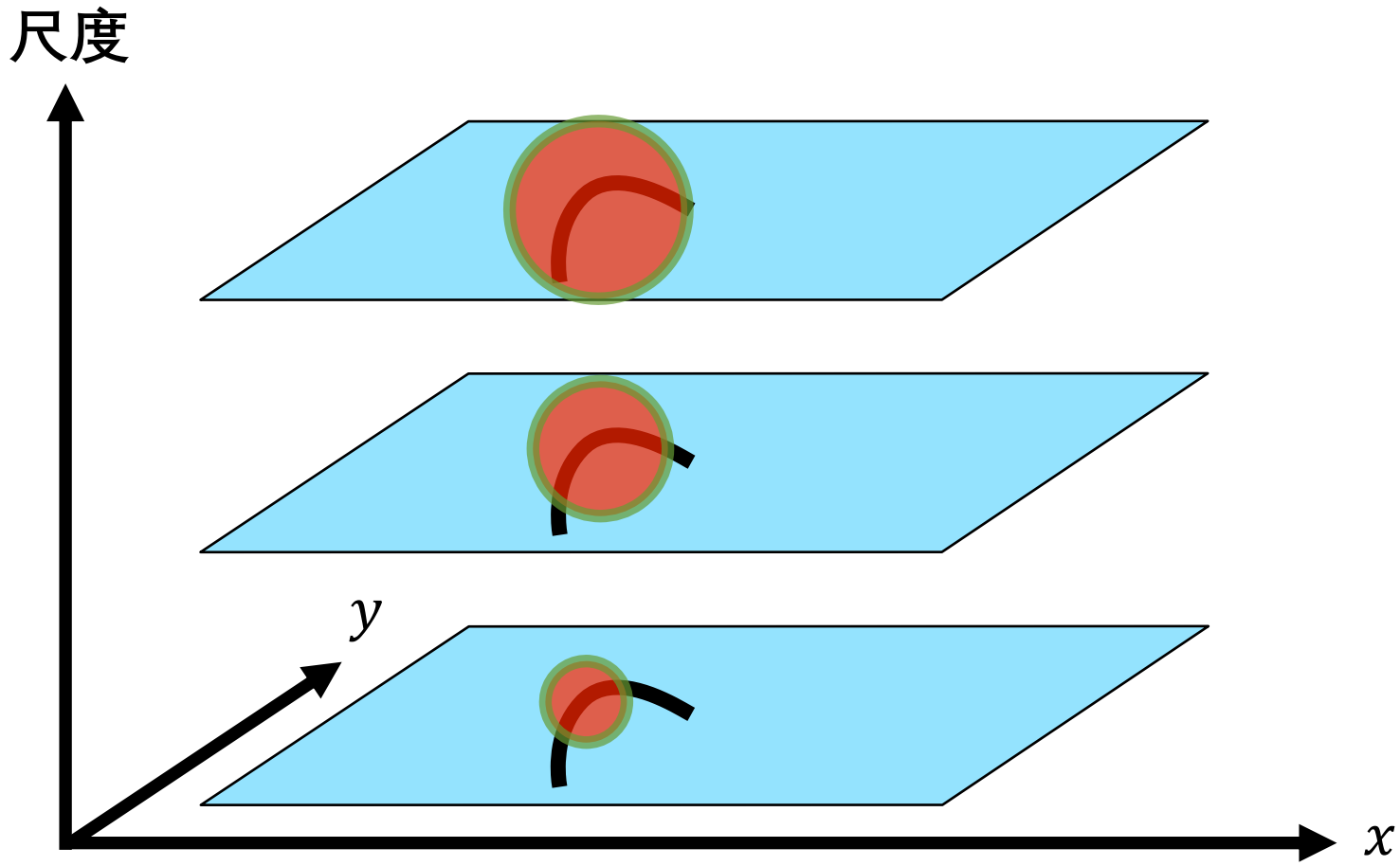
跨越多个尺度空间与DoG卷积



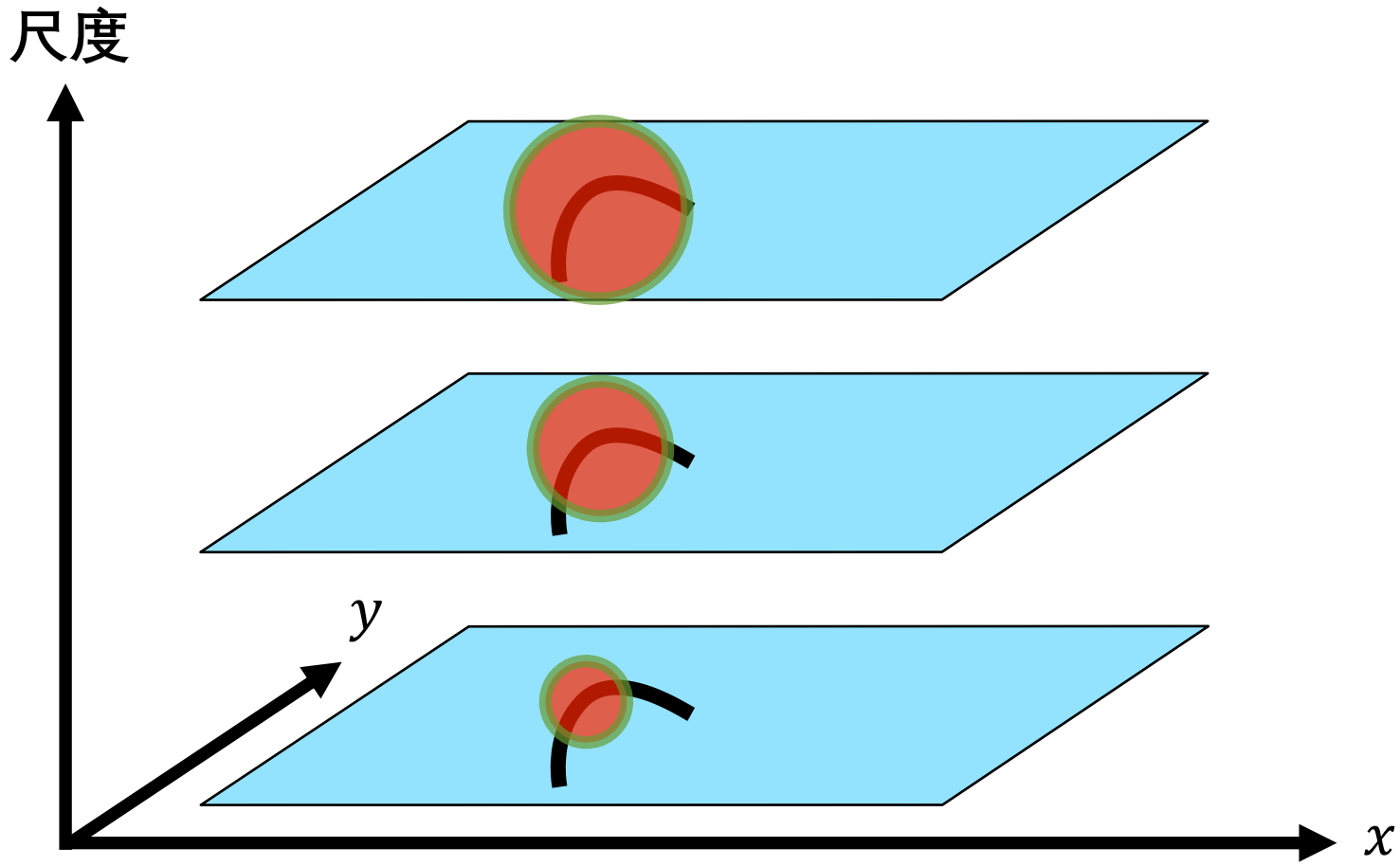
跨越多个尺度空间与DoG卷积



跨越多个尺度空间与DoG卷积

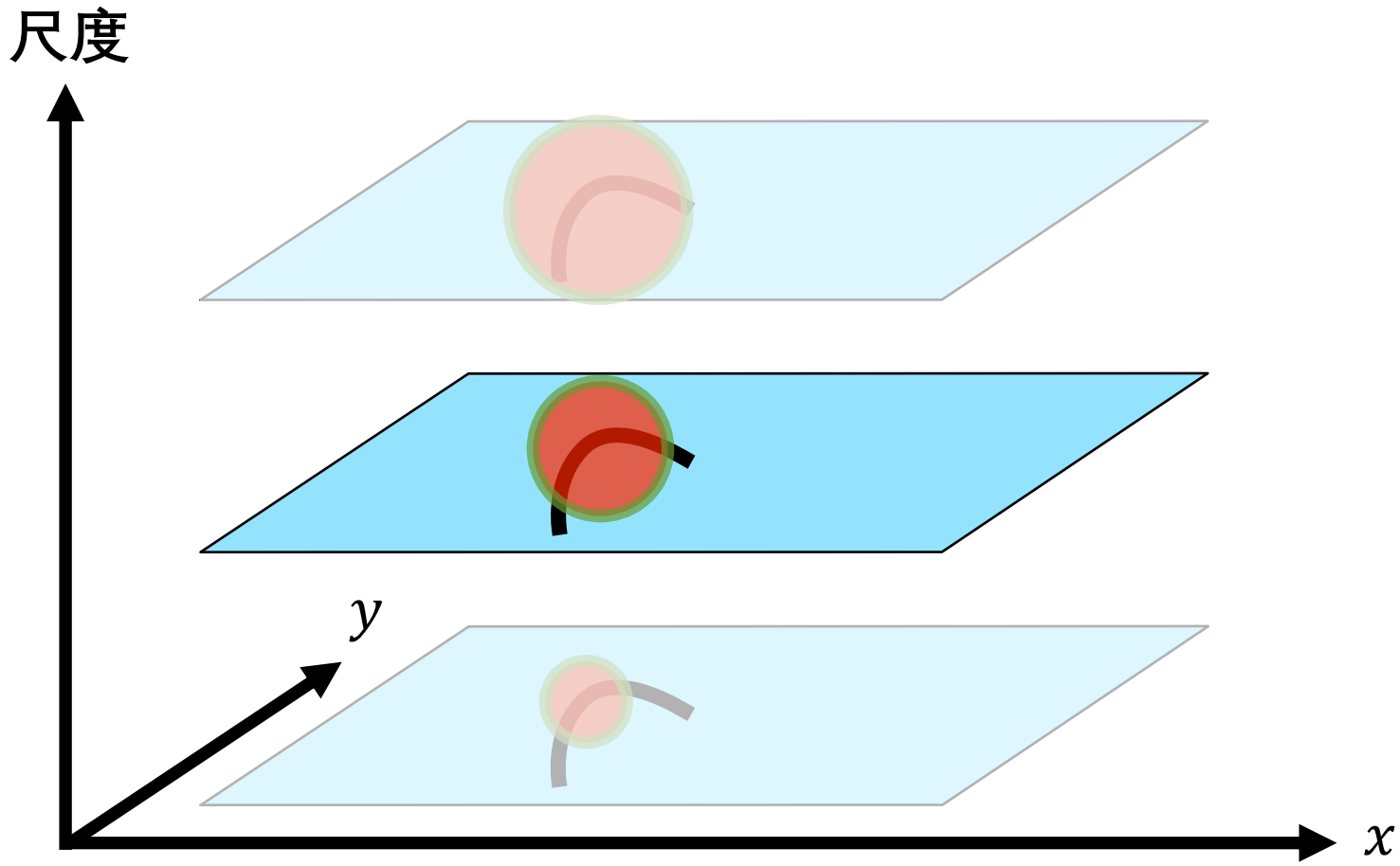


在空间中寻找DoG的局部最大值



在空间中寻找DoG的局部最大值

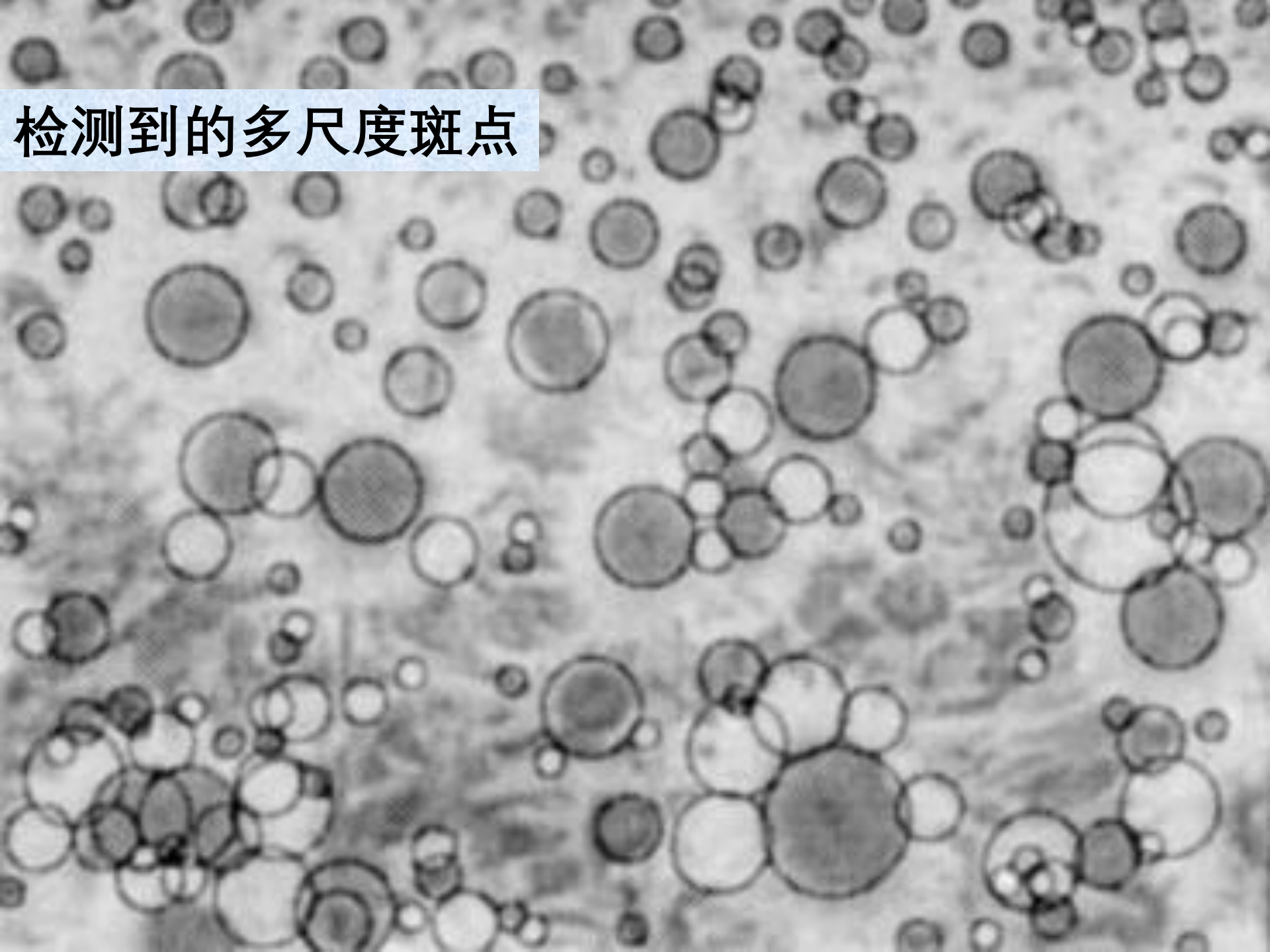
和尺度

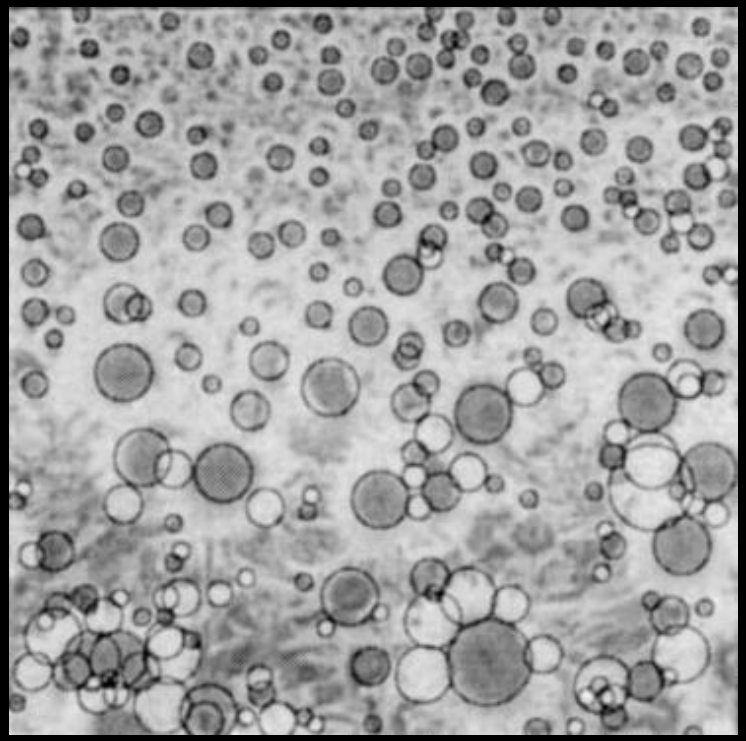


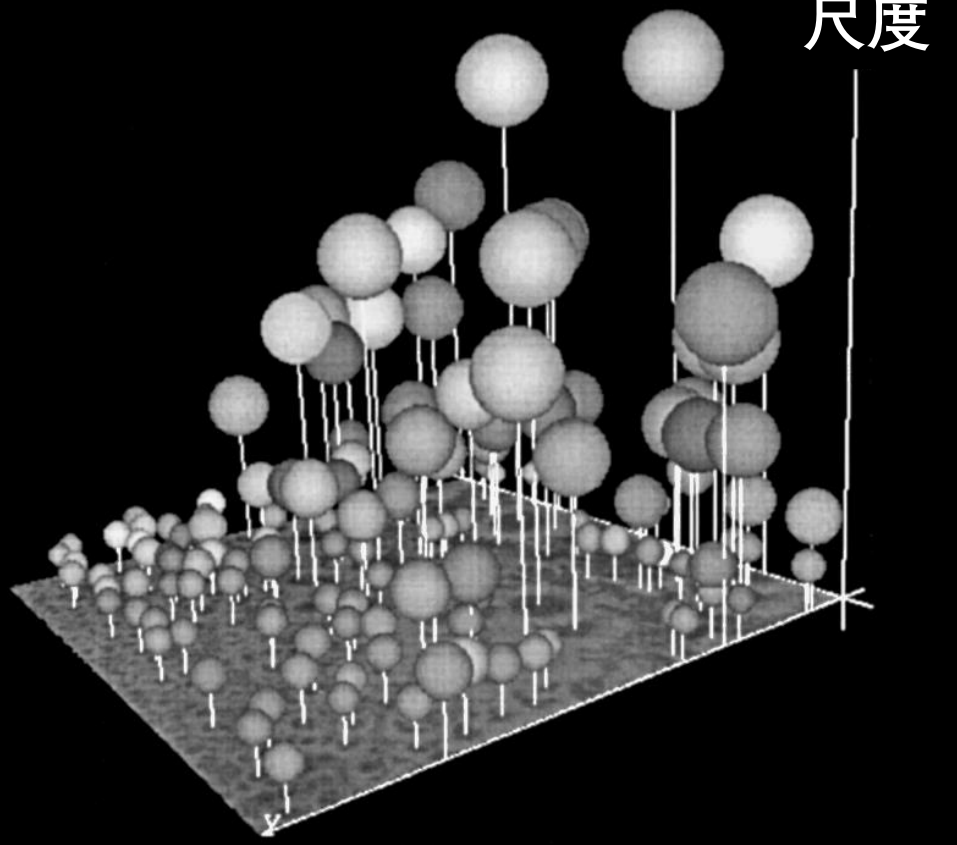
在空间中寻找DoG的局部最大值

和尺度


输入图像


检测到的多尺度斑点

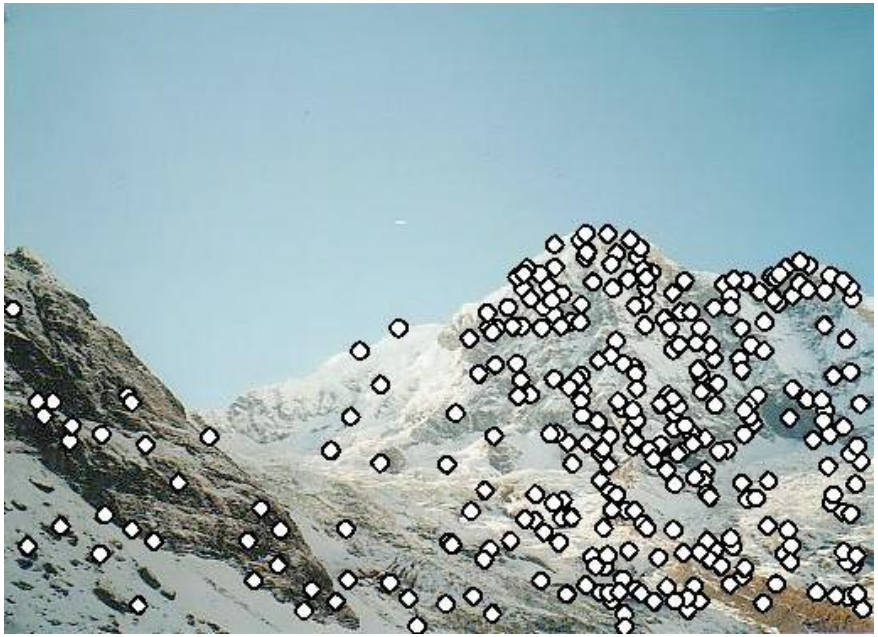


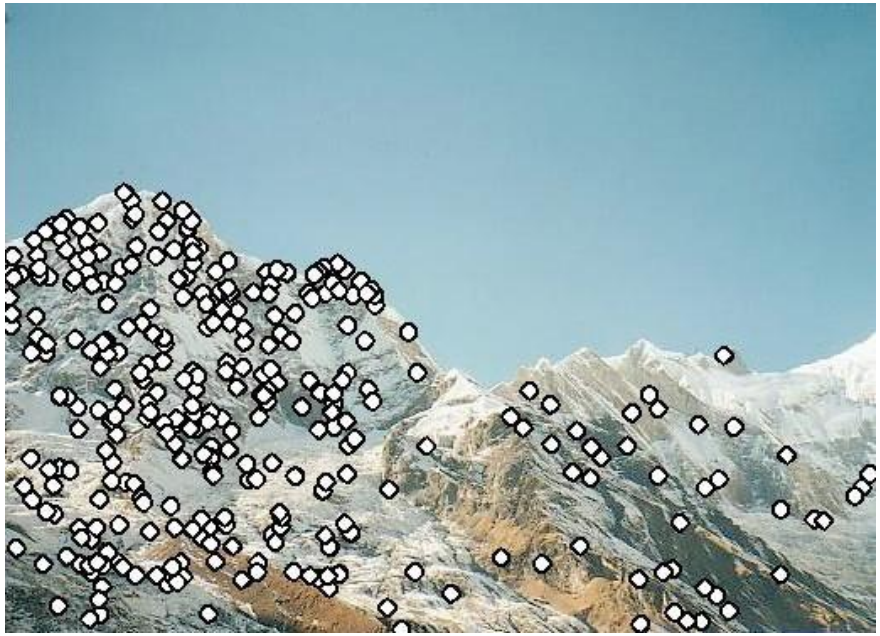


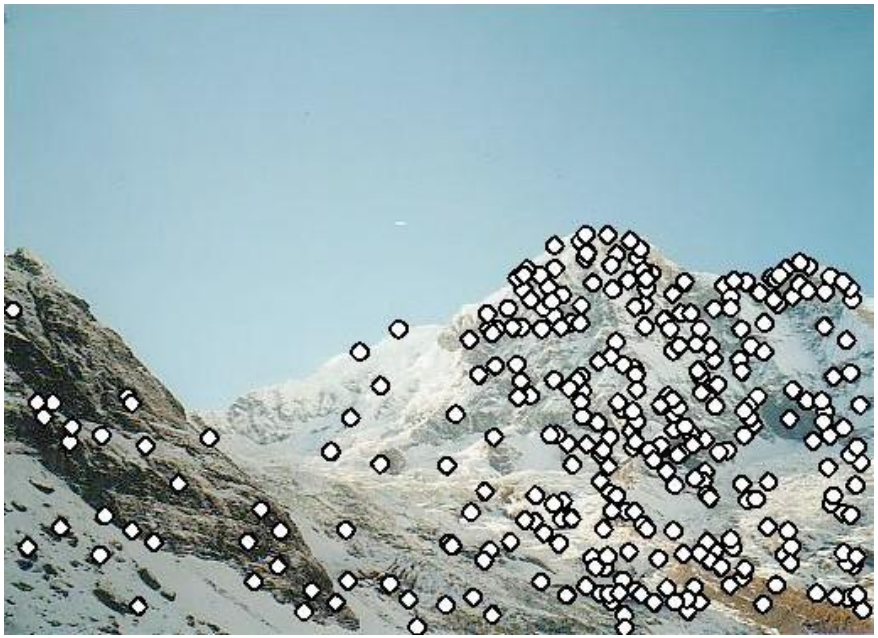
Lindeberg, Feature Detection with Automatic Scale Selection, IJCV, 1998

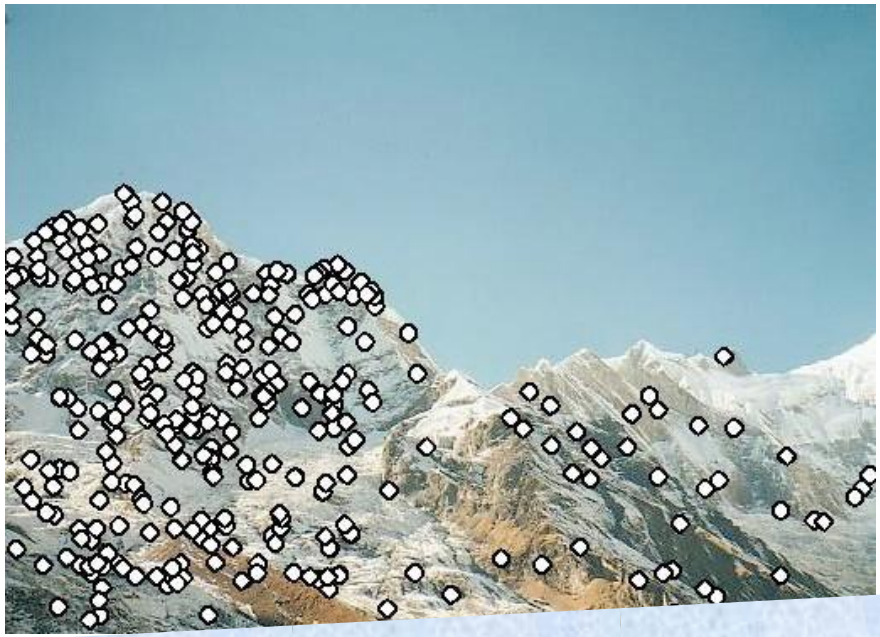
SIFT

特征描述子

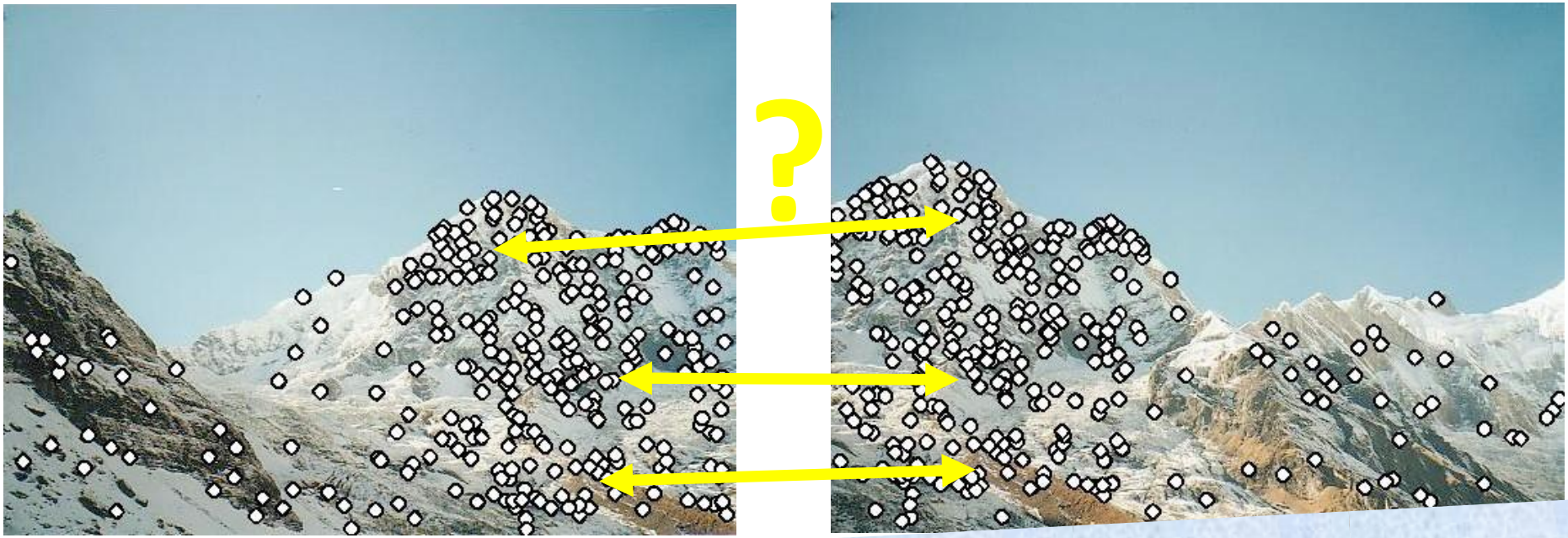








如何在图像间匹配特征?



如何在图像间匹配特征?

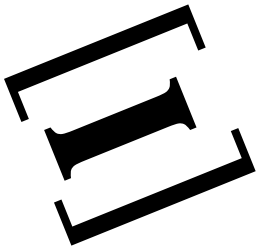
旋转不变描述

旋转不变描述

步骤1：找出图像块的主导方向

旋转不变描述

步骤1：找出图像块的主导方向

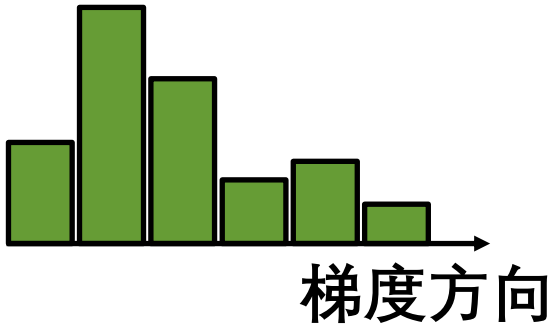


旋转不变描述

步骤1：找出图像块的主导方向

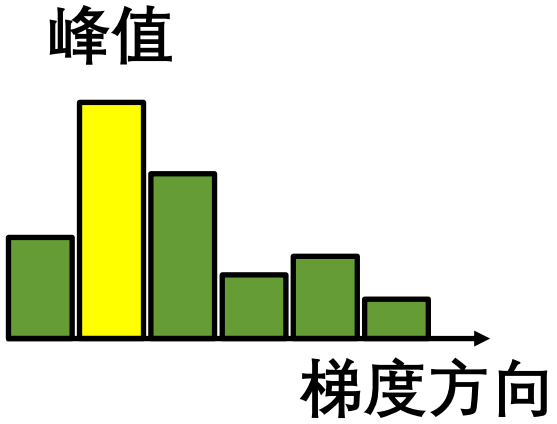
旋转不变描述

步骤1：找出图像块的主导方向



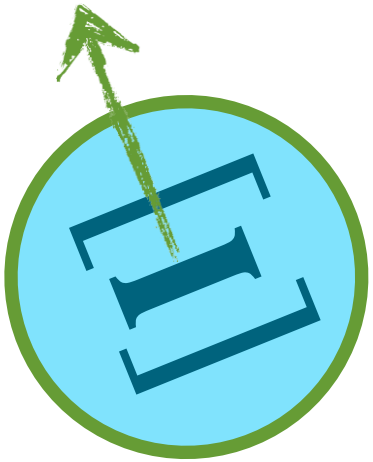
旋转不变描述

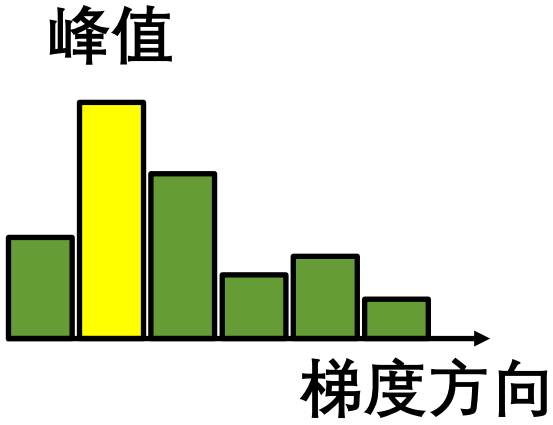
步骤1：找出图像块的主导方向



旋转不变描述

步骤1：找出图像块的主导方向

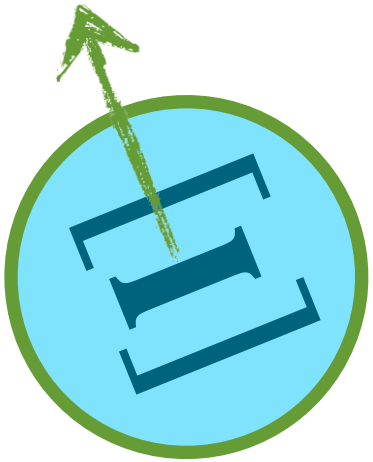


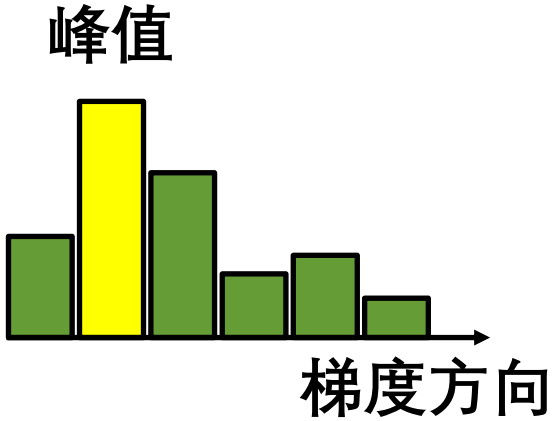


旋转不变描述

步骤1：找出图像块的主导方向

步骤2：旋转图像块指向x轴正方向

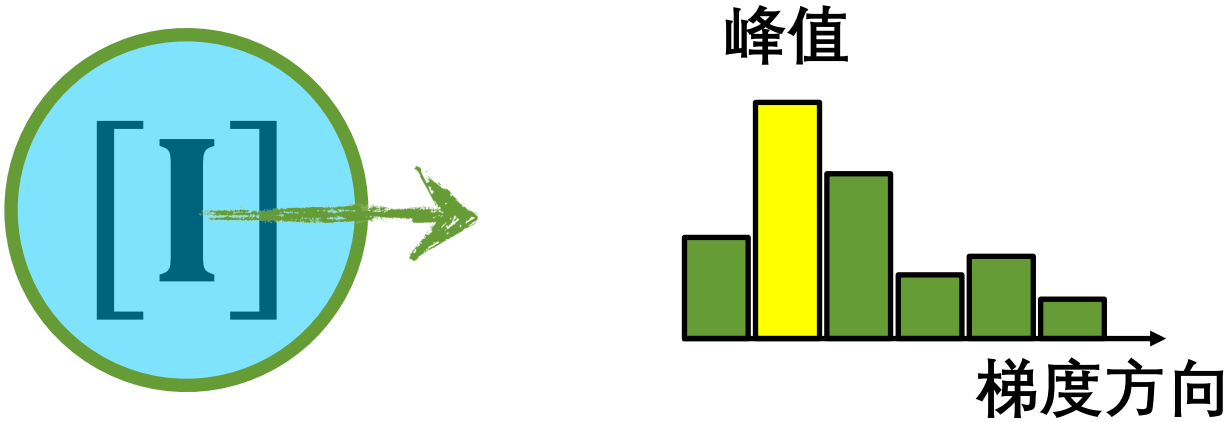




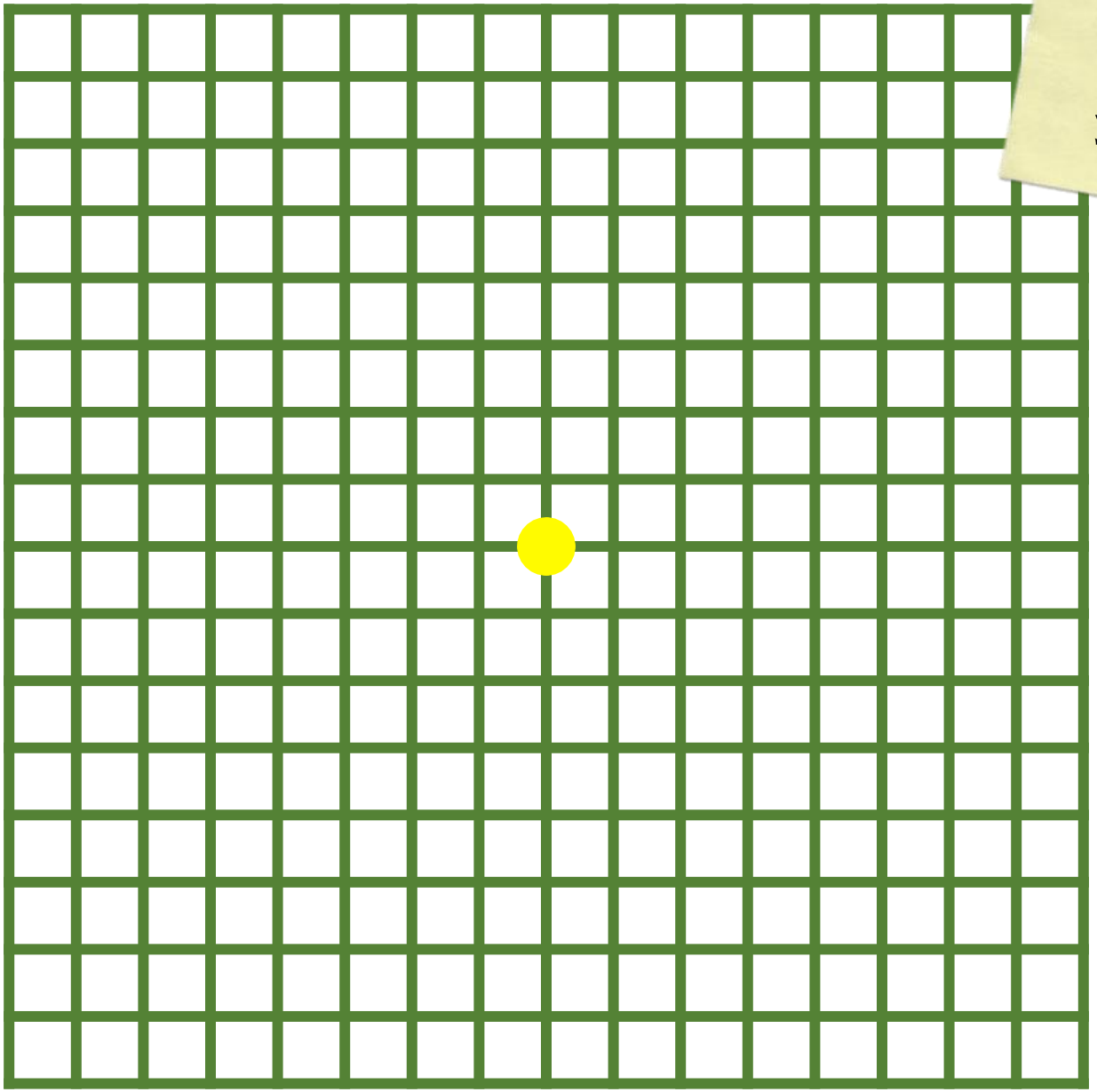
旋转不变描述

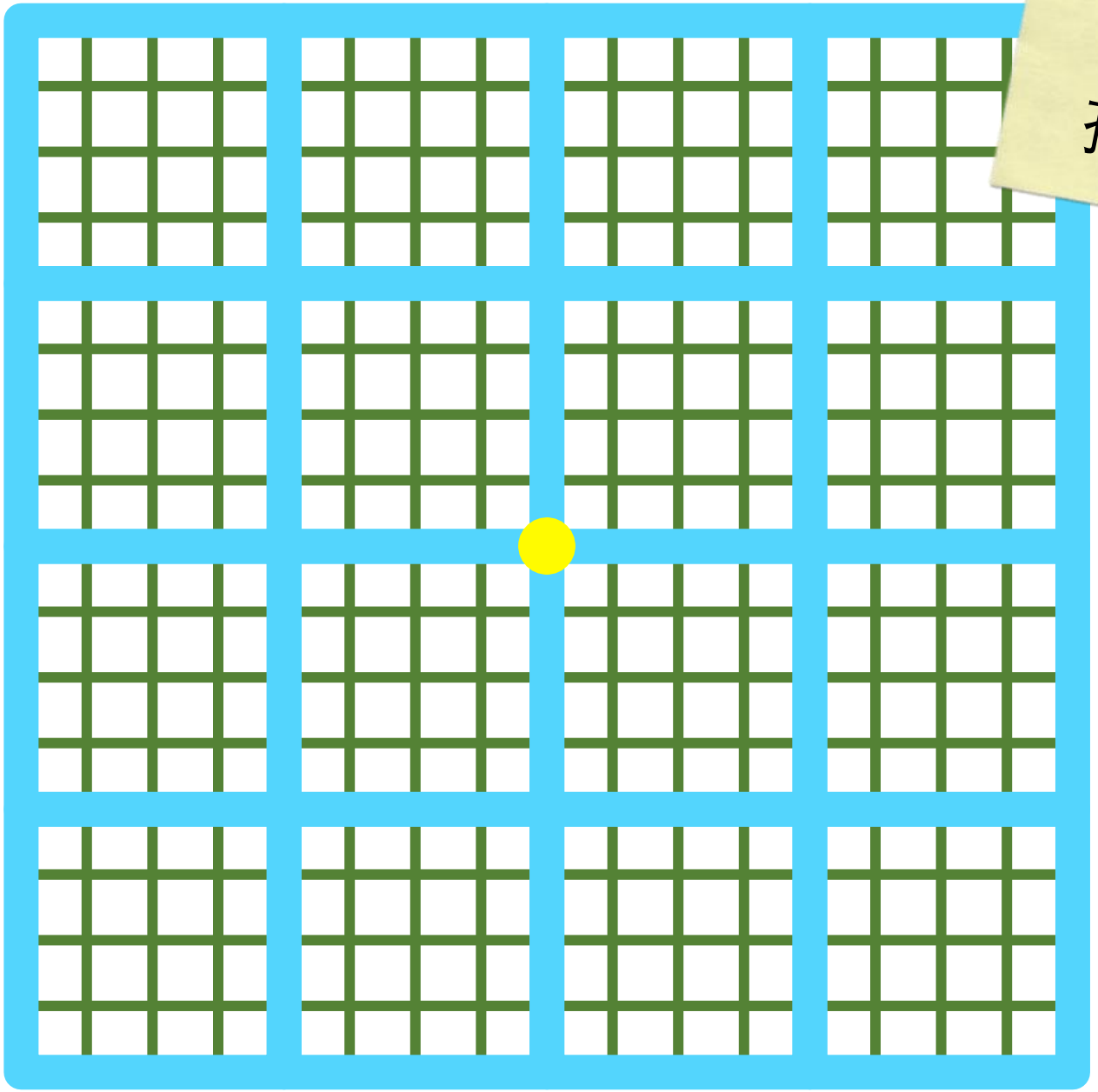
步骤1：找出图像块的主导方向

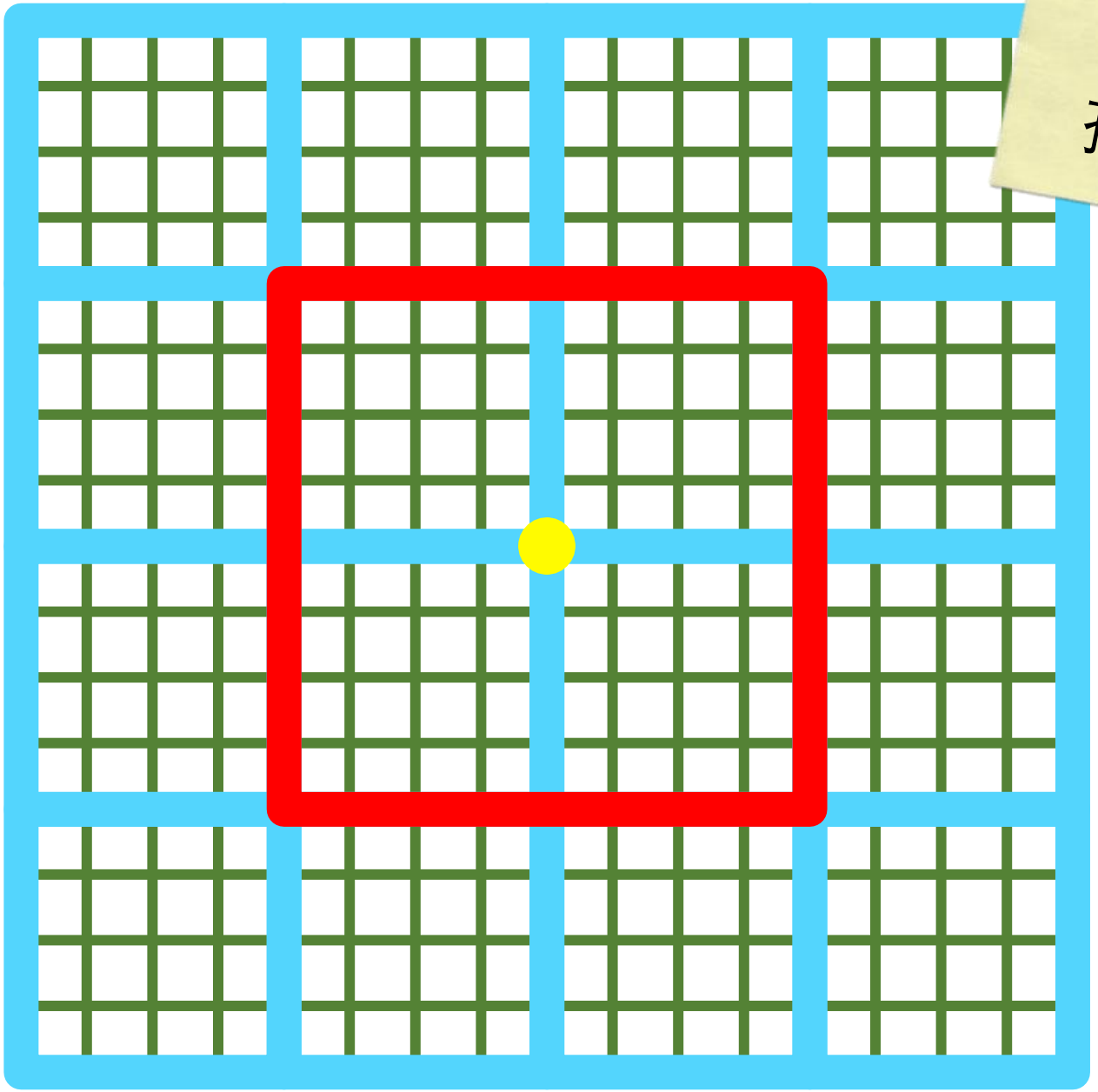
步骤2：旋转图像块指向x轴正方向



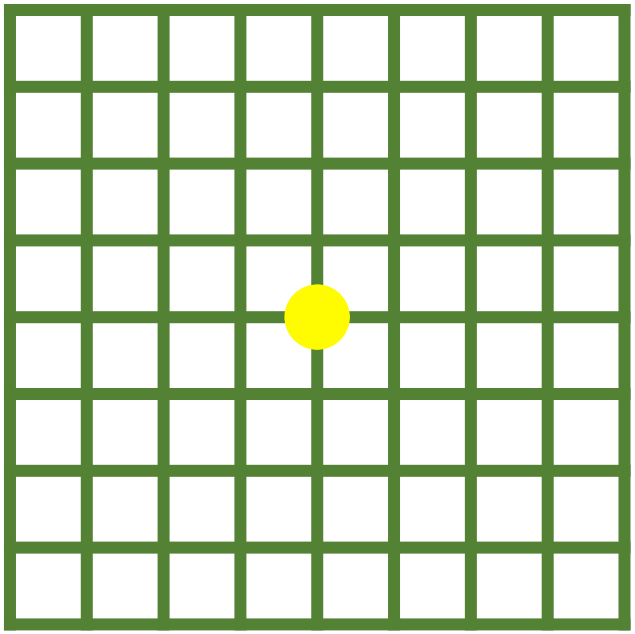
SIFT
描述子


SIFT
描述子

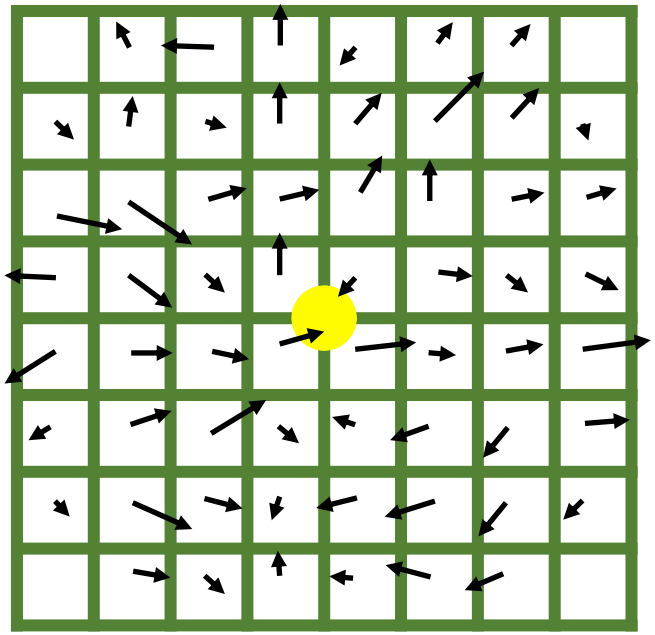

SIFT
描述子


SIFT
描述子

SIFT
描述子

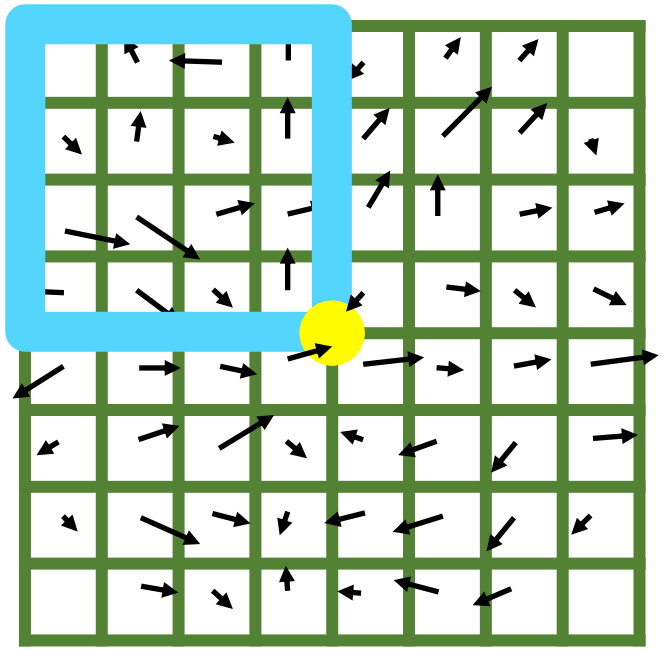


SIFT
描述子



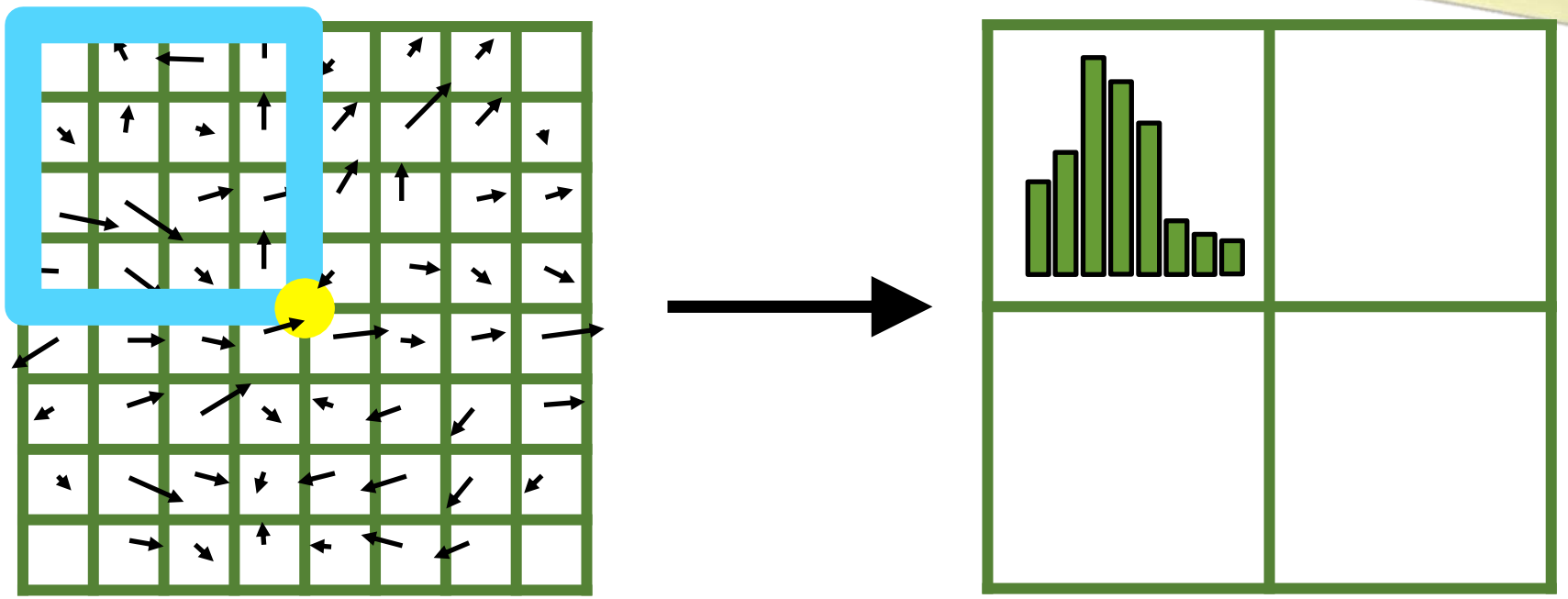
图像梯度

SIFT
描述子



图像梯度

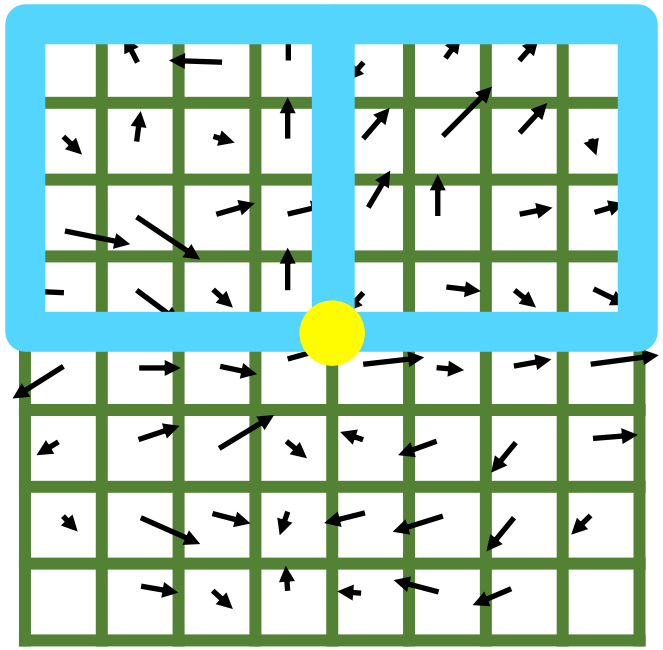
SIFT
描述子

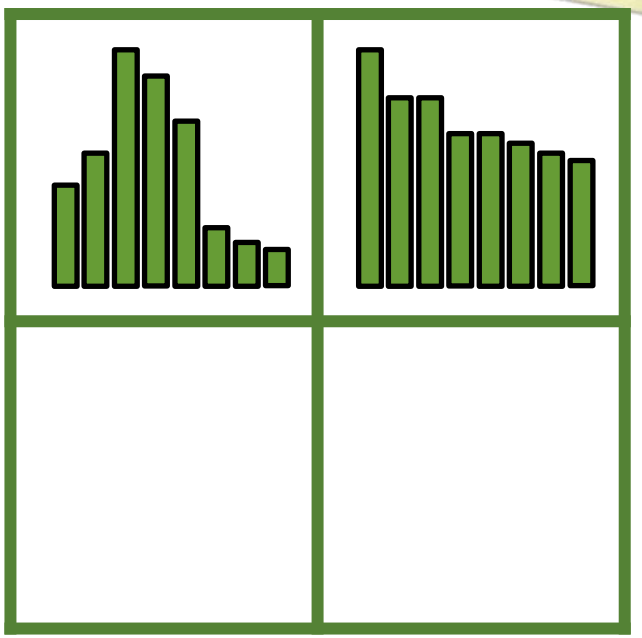


图像梯度

关键点描述子

SIFT
描述子

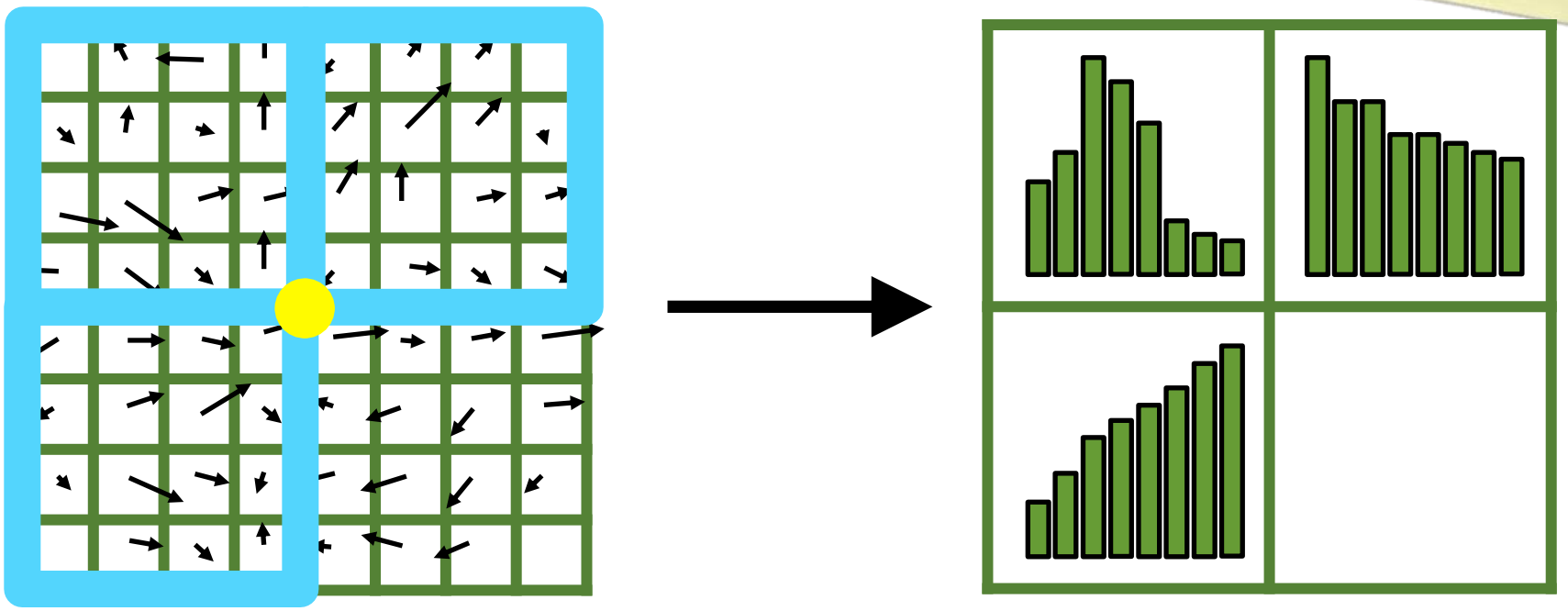




图像梯度

关键点描述子

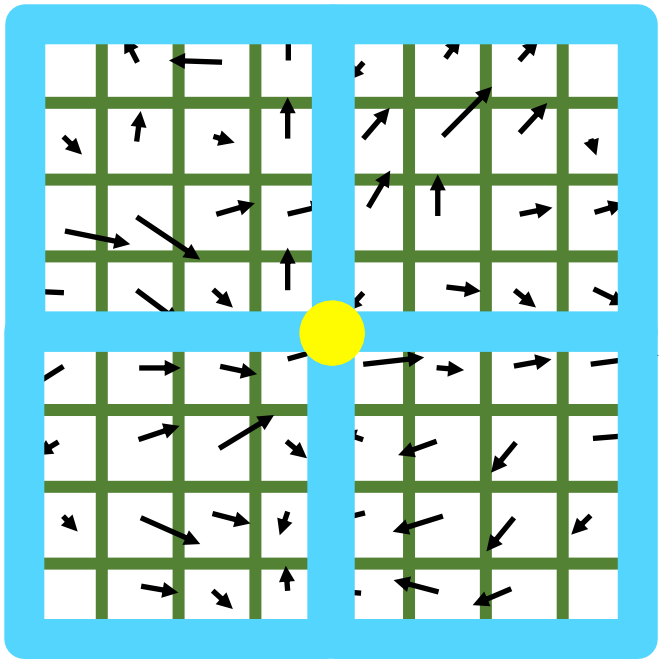
SIFT
描述子

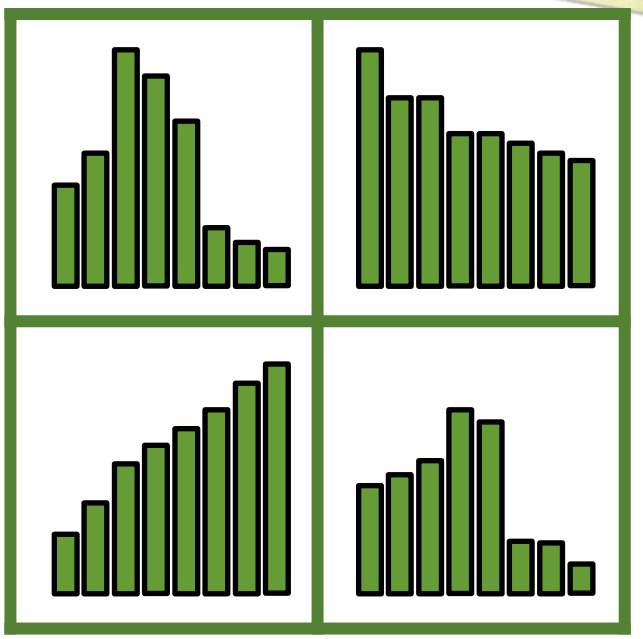


图像梯度

关键点描述子

SIFT
描述子

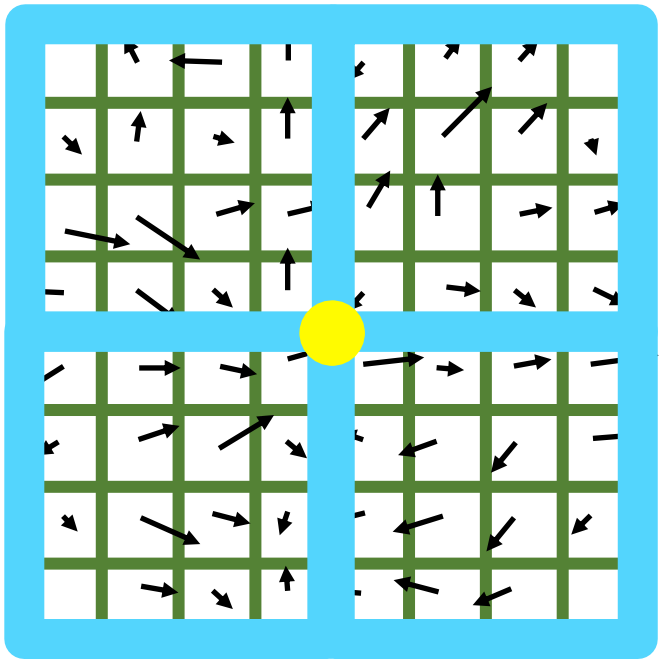


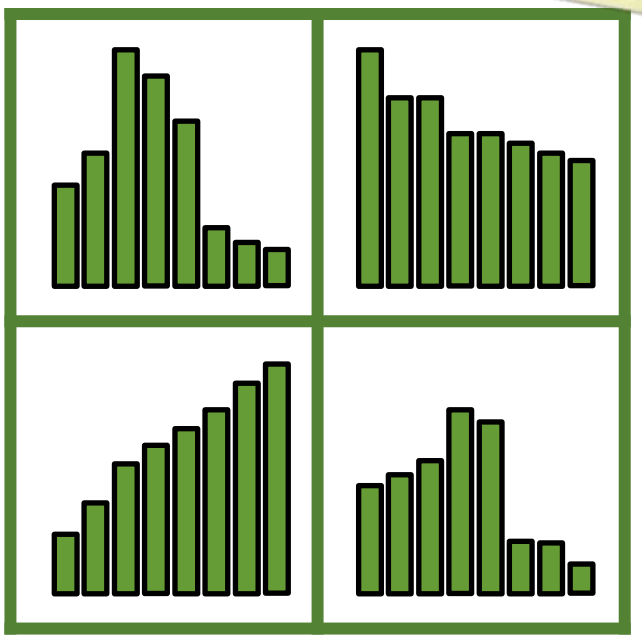


图像梯度

关键点描述子

SIFT
描述子





图像梯度

关键点描述子

16个子图像×8个方向=128维描述子


Recreation
& Athletics

SIFT性质

SIFT性质

对图像旋转、缩放和强度变化具有鲁棒性

SIFT性质

对图像旋转、缩放和强度变化具有鲁棒性

对适度的平面外旋转具有鲁棒性

SIFT性质

对图像旋转、缩放和强度变化具有鲁棒性

对适度的平面外旋转具有鲁棒性

快速且高效

太长不看?

SIFT总结

SIFT总结

1. 关键点检测

SIFT总结

1. 关键点检测

跨图像位置和尺度搜索特征响应极值

SIFT总结

1. 关键点检测

跨图像位置和尺度搜索特征响应极值

2. 关键点方向对齐

1. 关键点检测

跨图像位置和尺度搜索特征响应极值

2. 关键点方向对齐

确定每个关键点的最佳方向

1. 关键点检测

跨图像位置和尺度搜索特征响应极值

2. 关键点方向对齐

确定每个关键点的最佳方向

3. 关键点描述

SIFT总结

1. 关键点检测

   跨图像位置和尺度搜索特征响应极值

2. 关键点方向对齐

   确定每个关键点的最佳方向

3. **关键点描述**

   在选定的尺度和旋转下，用区域图像梯度描述关键点

SIFT总结

1. 关键点检测

跨图像位置和尺度搜索特征响应极值

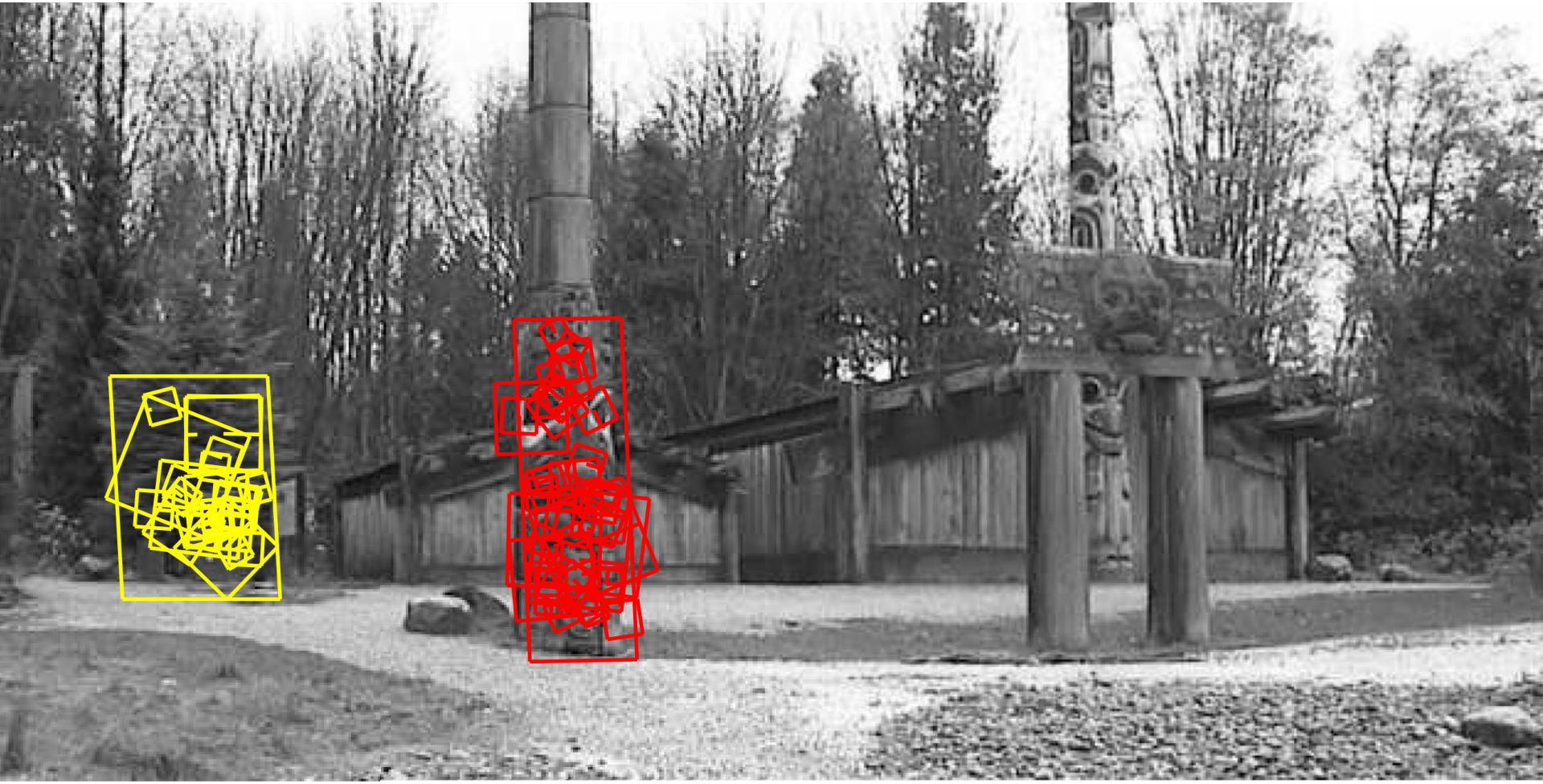
2. 关键点方向对齐

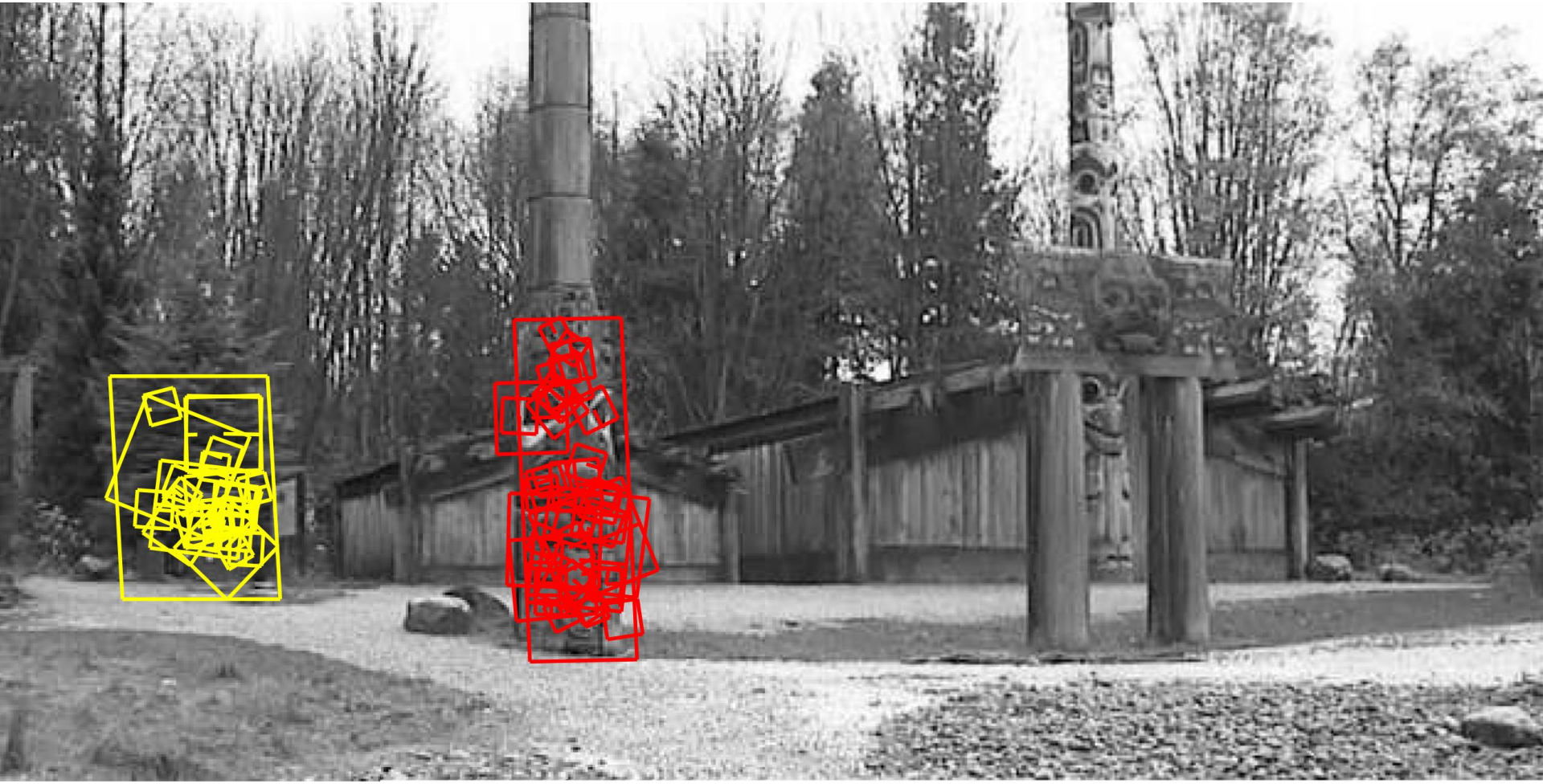
确定每个关键点的最佳方向

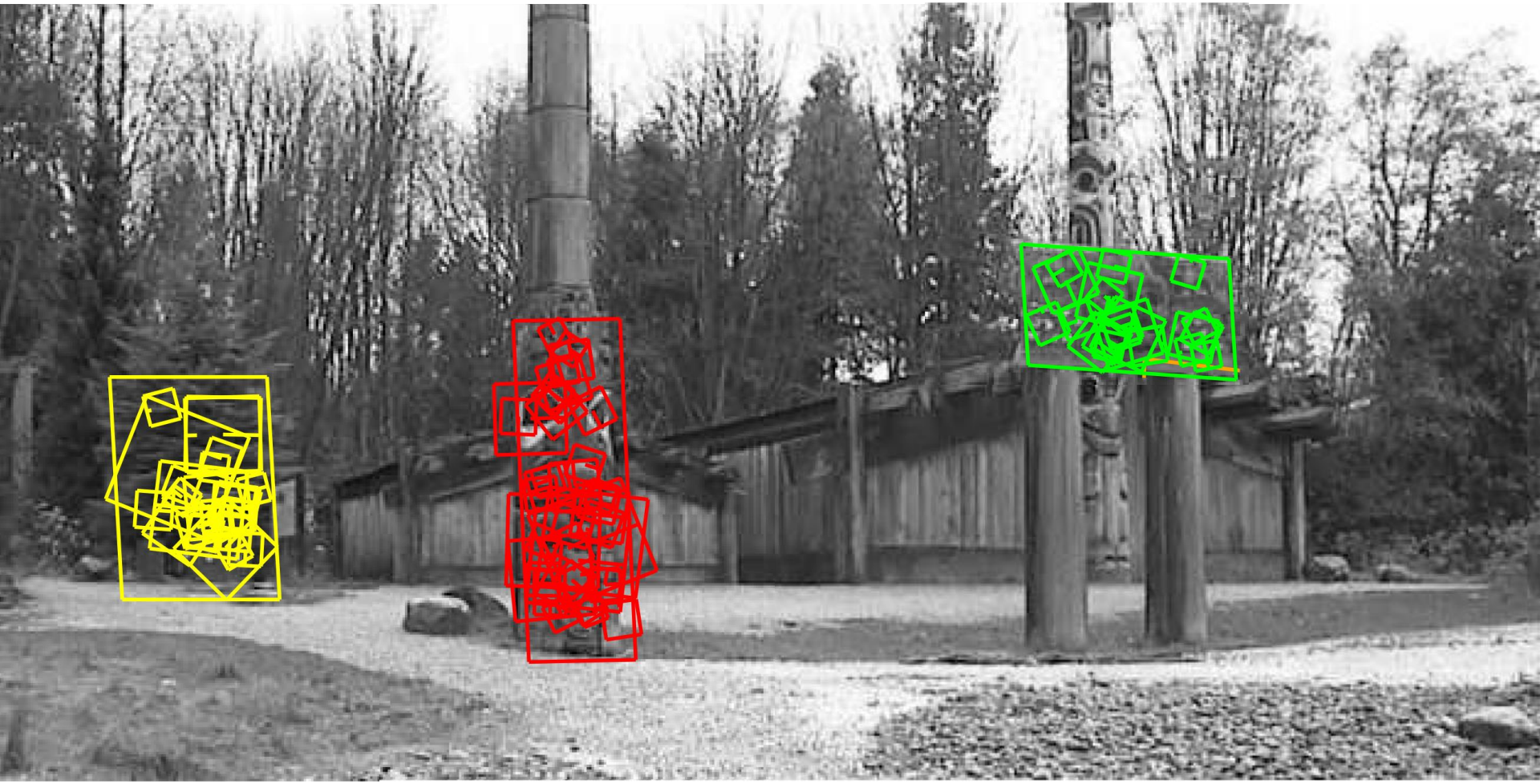
3. 关键点描述

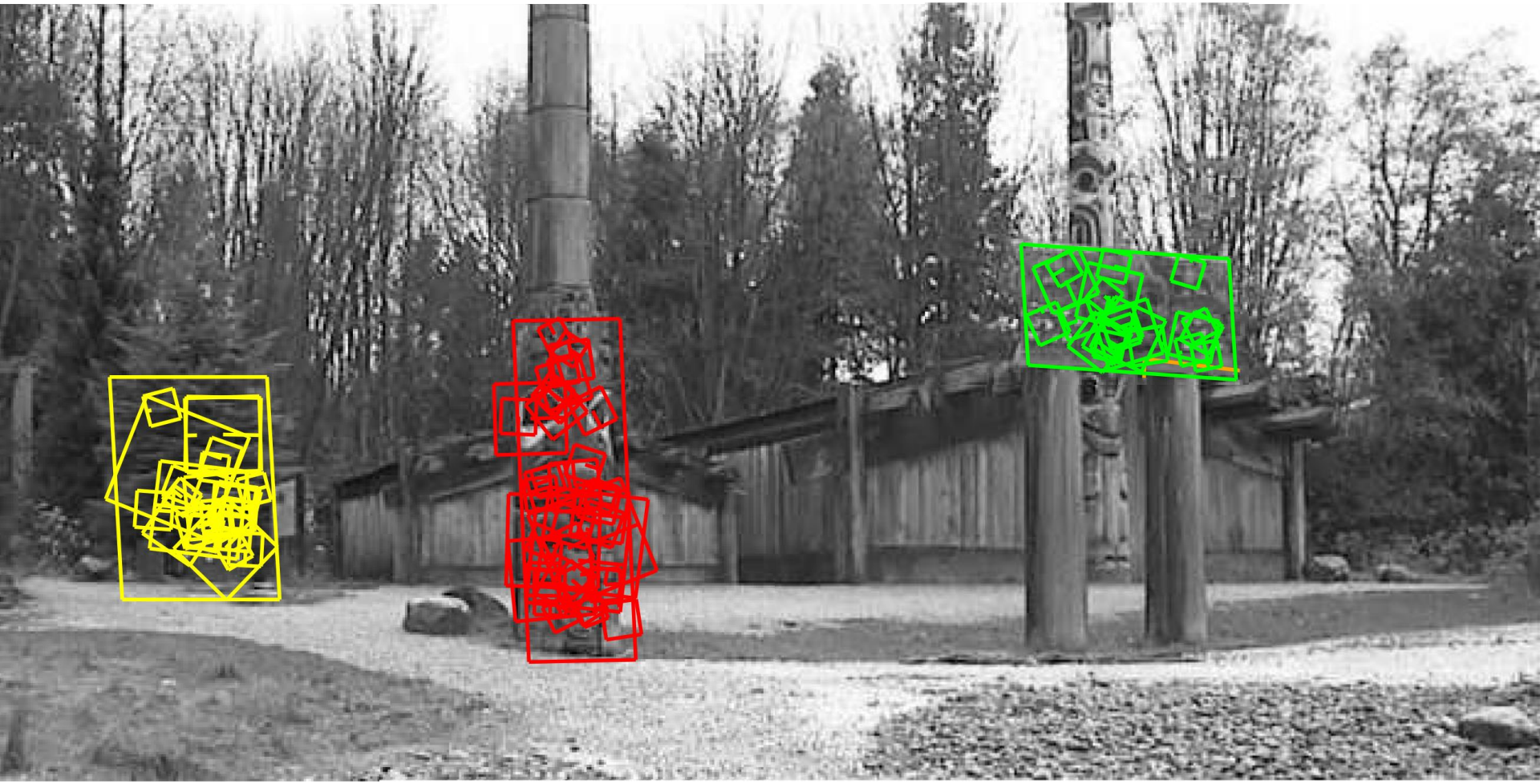
在选定的尺度和旋转下，用区域图像梯度描述关键点

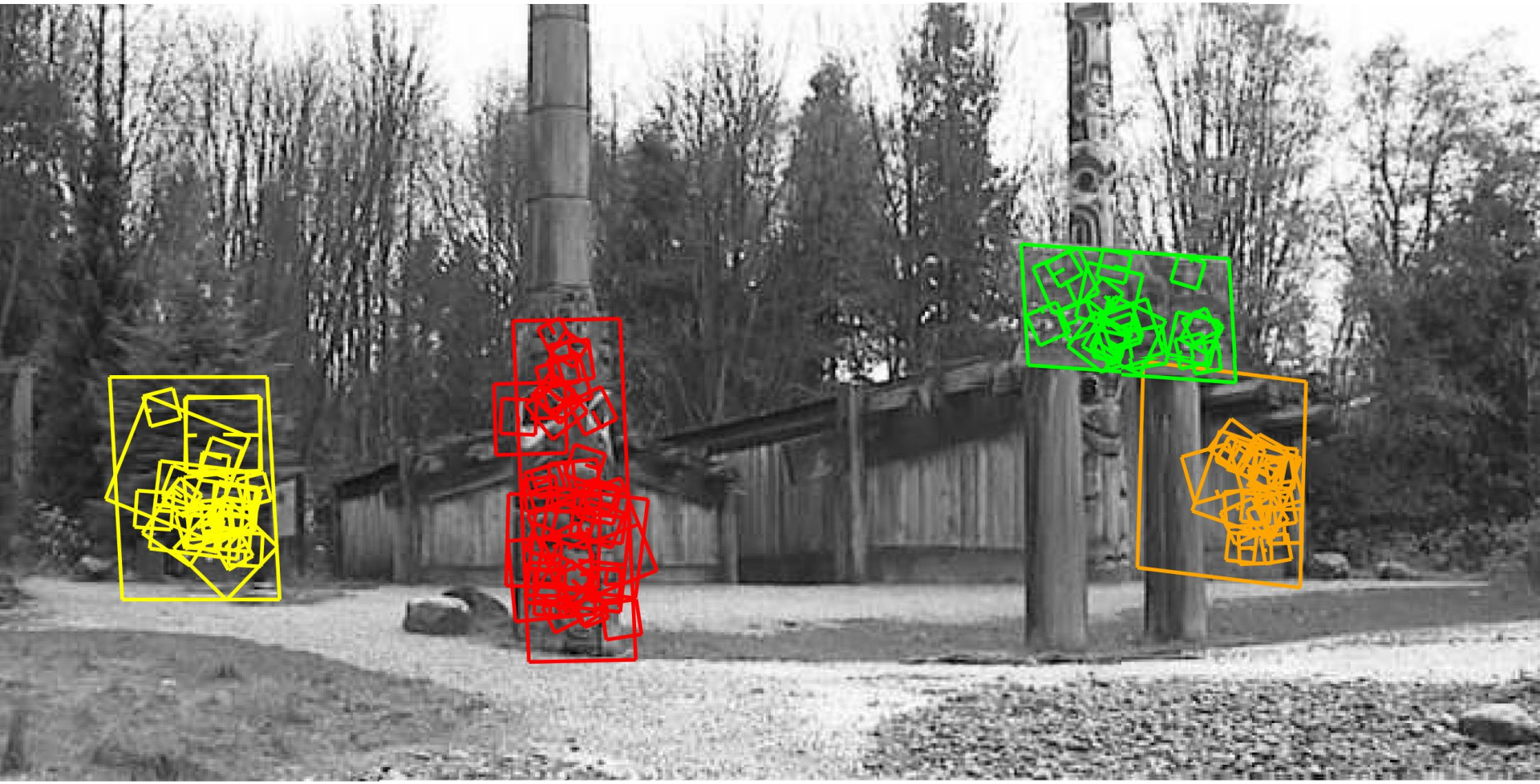
挑战：找到下面这些图像块











SIFT

Scale-Invariant Feature Transform

尺度不变特征变换

SIFT

Scale-Invariant Feature Transform

尺度不变特征变换

SIFT

Scale-Invariant Feature Transform

尺度不变特征变换

Introduction to SIFT (Scale-Invariant Feature Transform)

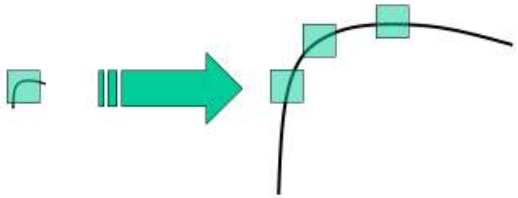
Goal

In this chapter,

- We will learn about the concepts of SIFT algorithm
- We will learn to find SIFT Keypoints and Descriptors.

Theory

In last couple of chapters, we saw some corner detectors like Harris etc. They are rotation-invariant, which means, even if the image is rotated, we can find the same corners. It is obvious because corners remain corners in rotated image also. But what about scaling? A corner may not be a corner if the image is scaled. For example, check a simple image below. A corner in a small image within a small window is flat when it is zoomed in the same window. So Harris corner is not scale invariant.



image

In 2004, **D.Lowe**, University of British Columbia, came up with a new algorithm, Scale Invariant Feature Transform (SIFT) in his paper, **Distinctive Image Features from Scale-Invariant Keypoints**, which extract keypoints and compute its descriptors. *(This paper is easy to understand and considered to be best material available on SIFT. This explanation is just a short summary of this paper)*.

There are mainly four steps involved in SIFT algorithm. We will see them one-by-one.

1. Scale-space Extrema Detection

From the image above, it is obvious that we can't use the same window to detect keypoints with different scale. It is OK with small corner. But to detect larger corners we need larger windows. For this, scale-space filtering is used. In it, Laplacian of Gaussian is found for the image with various σ values. LoG acts as

docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html

Introduction to SIFT (Scale-Invariant Feature Transform)

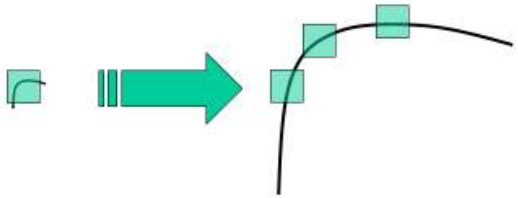
Goal

In this chapter,

- We will learn about the concepts of SIFT algorithm
- We will learn to find SIFT Keypoints and Descriptors.

Theory

In last couple of chapters, we saw some corner detectors like Harris etc. They are rotation-invariant, which means, even if the image is rotated, we can find the same corners. It is obvious because corners remain corners in rotated image also. But what about scaling? A corner may not be a corner if the image is scaled. For example, check a simple image below. A corner in a small image within a small window is flat when it is zoomed in the same window. So Harris corner is not scale invariant.



image

In 2004, **D.Lowe**, University of British Columbia, came up with a new algorithm, Scale Invariant Feature Transform (SIFT) in his paper, **Distinctive Image Features from Scale-Invariant Keypoints**, which extract keypoints and compute its descriptors. *(This paper is easy to understand and considered to be best material available on SIFT. This explanation is just a short summary of this paper)*.

There are mainly four steps involved in SIFT algorithm. We will see them one-by-one.

1. Scale-space Extrema Detection

From the image above, it is obvious that we can't use the same window to detect keypoints with different scale. It is OK with small corner. But to detect larger corners we need larger windows. For this, scale-space filtering is used. In it, Laplacian of Gaussian is found for the image with various σ values. LoG acts as

还有一件事…

机器学习

LIFT: Learned Invariant Feature Transform

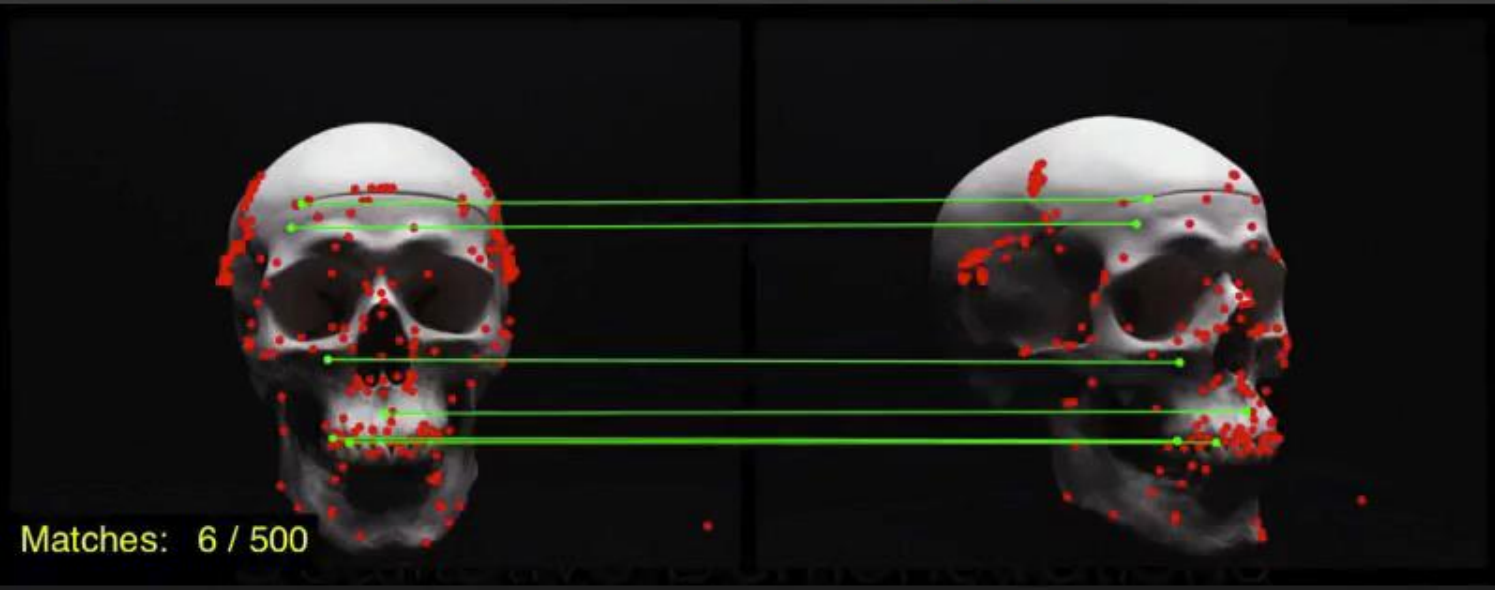
Kwang Moo Yi*,[1], Eduard Trulls*,[1], Vincent Lepetit[2], Pascal Fua[1]

[1]Computer Vision Laboratory, Ecole Polytechnique Fédérale de Lausanne (EPFL)
[2]Institute for Computer Graphics and Vision, Graz University of Technology
{kwang.yi, eduard.trulls, pascal.fua}@epfl.ch, lepetit@icg.tugraz.at

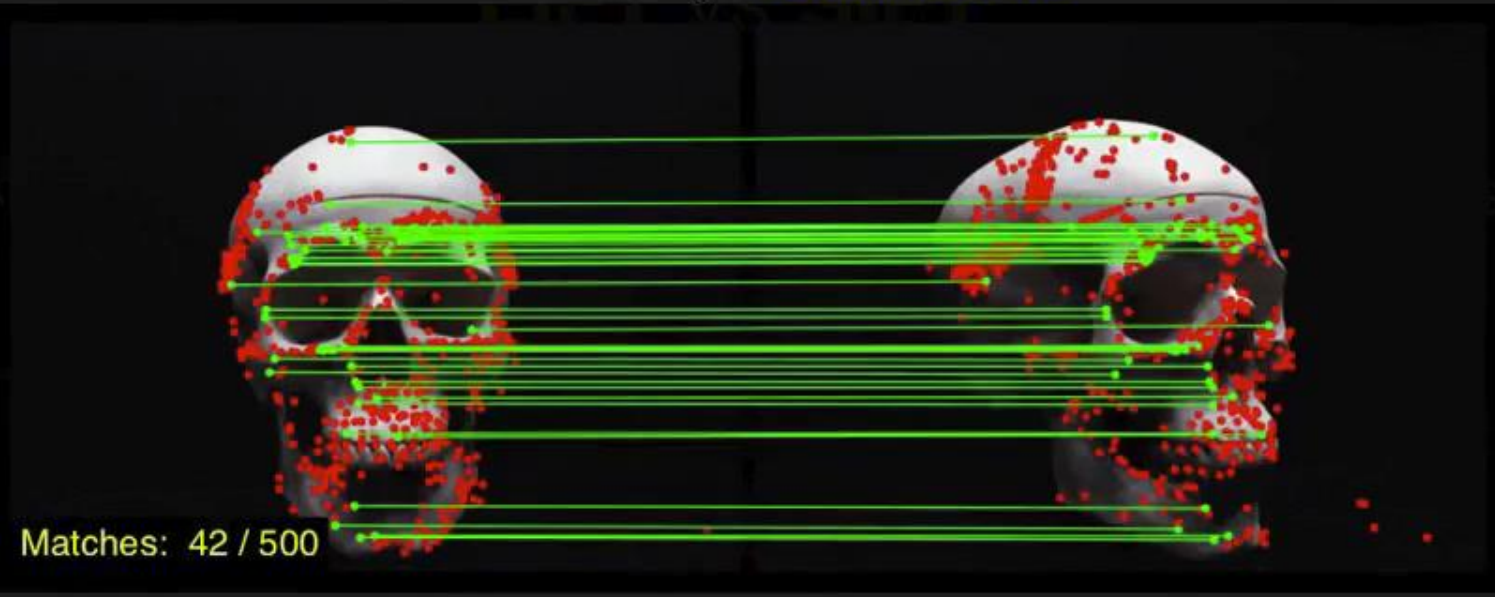
Abstract. We introduce a novel Deep Network architecture that implements the full feature point handling pipeline, that is, detection, orientation estimation, and feature description. While previous works have successfully tackled each one of these problems individually, we show how to learn to do all three in a unified manner while preserving end-to-end differentiability. We then demonstrate that our Deep pipeline outperforms state-of-the-art methods on a number of benchmark datasets, without the need of retraining.

European Conference on Computer Vision (ECCV) 2016

Matching features on **'DTU'**, sequence **#19**.
Correct matches shown with **green** lines.



SIFT. Average: **34.1** matches



LIFT (Ours). Average: **98.5** matches