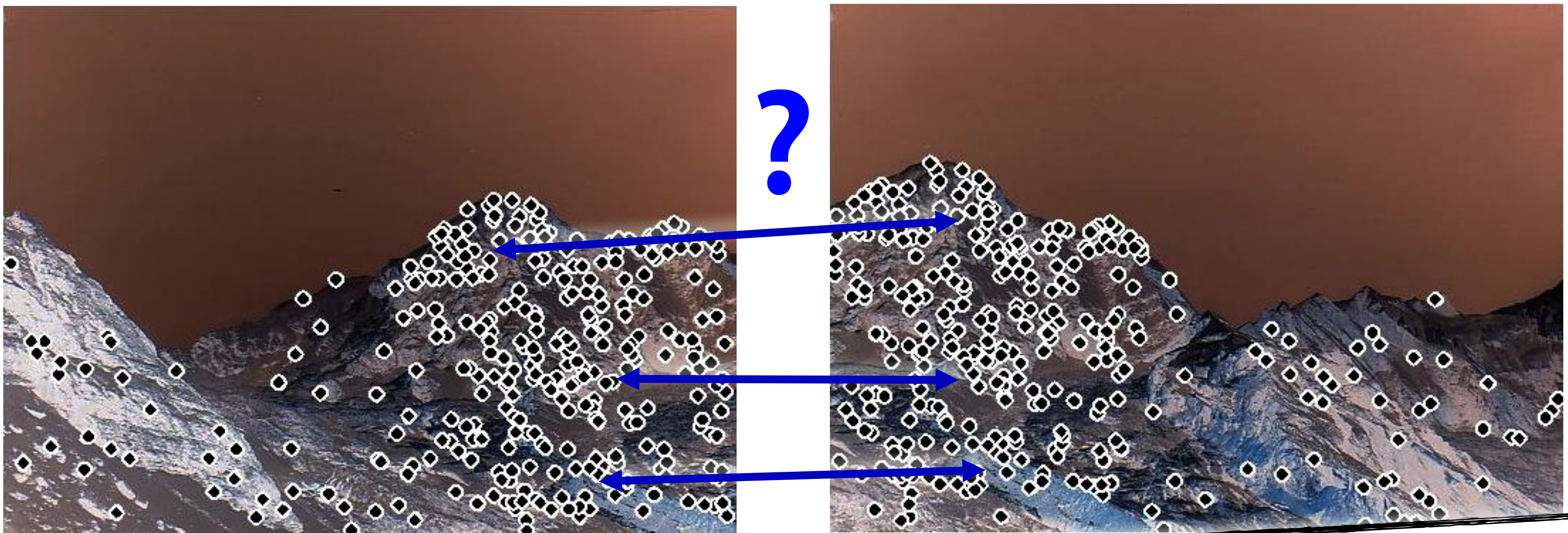
Intro to

Computer Vision

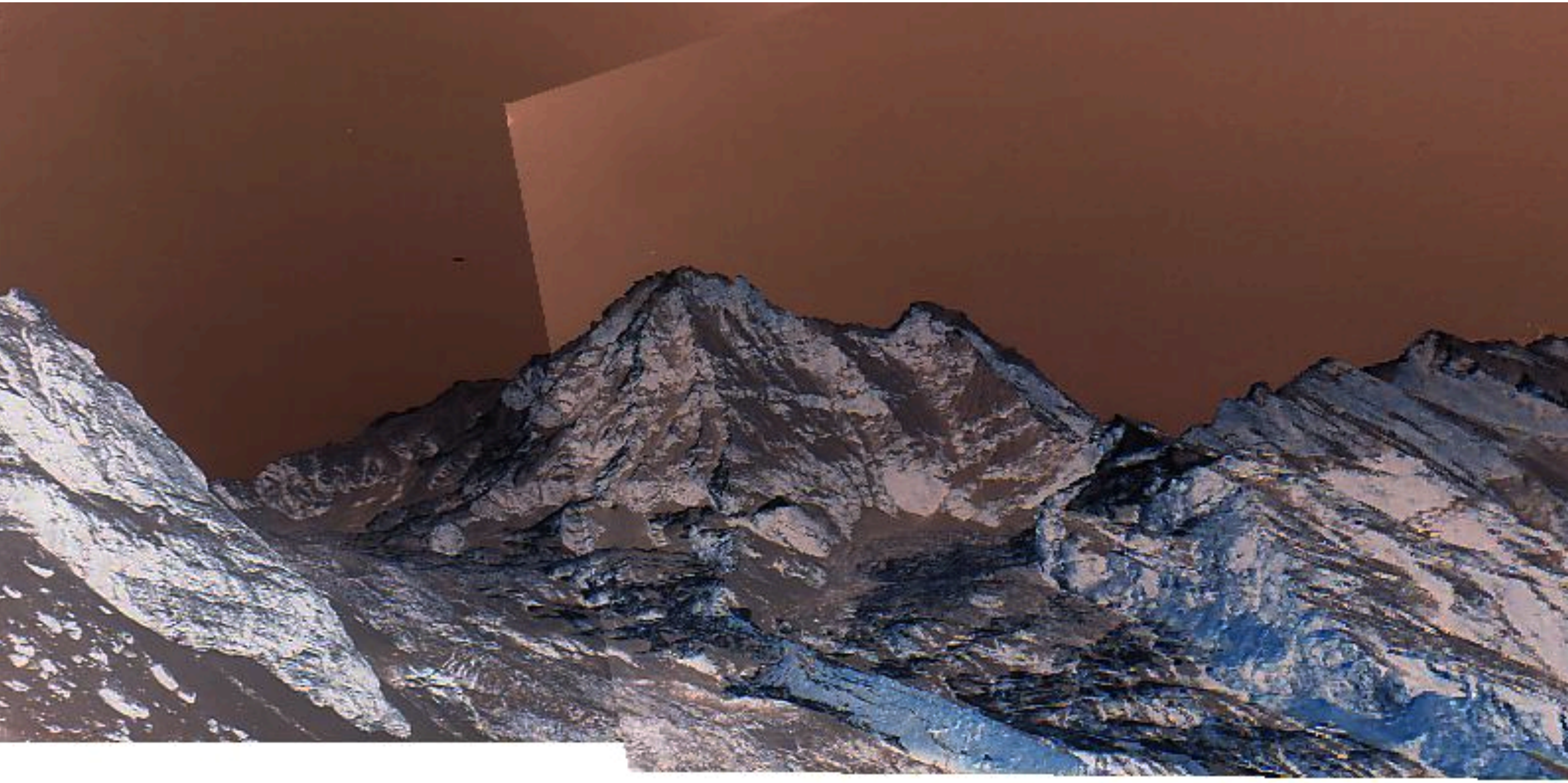
with Prof. Kosta Derpanis

Feature detection



What stuff in an image matches with stuff in another?

Find matching pairs and align



Rome dataset

74,394 images

Schönberger and Frahm, Structure from Motion Revisited, CVPR, 2016

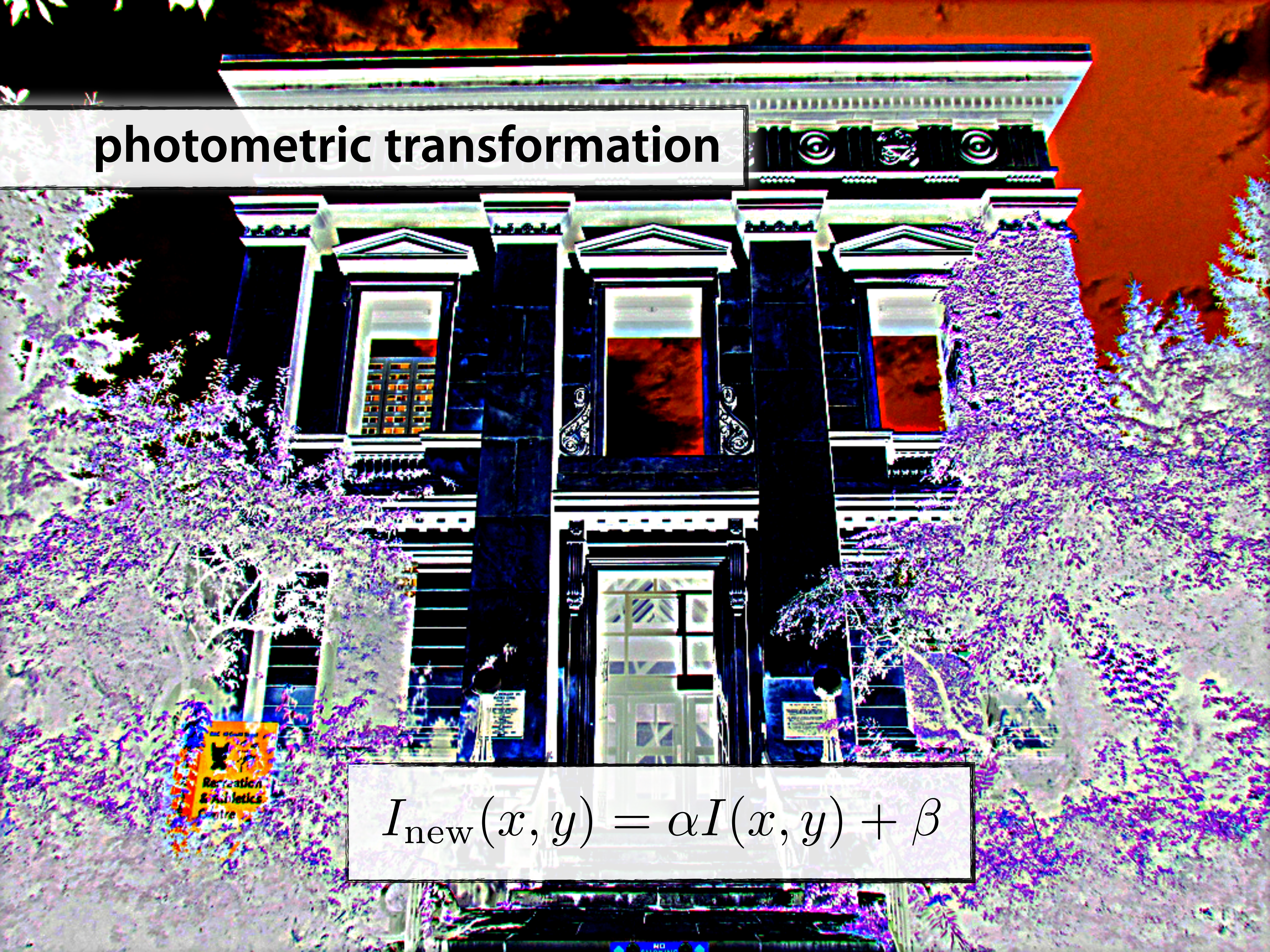
What makes a “good” feature?

REPEATABILITY

Same feature can be found in other images despite geometric/photometric remappings


input image

photometric transformation



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

geometric transformation

REPEATABILITY

Same feature can be found in other images despite geometric/photometric remappings

REPEATABILITY

Same feature can be found in other images despite geometric/photometric remappings

SALIENCY

Each feature is distinctive

REPEATABILITY

Same feature can be found in other images despite geometric/photometric remappings

SALIENCY

Each feature is distinctive

COMPACTNESS AND EFFICIENCY

Many fewer features than image pixels

REPEATABILITY

Same feature can be found in other images despite geometric/photometric remappings

SALIENCY

Each feature is distinctive

COMPACTNESS AND EFFICIENCY

Many fewer features than image pixels

LOCALITY

Feature occupies a relatively small area of the image

Why is locality important?

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

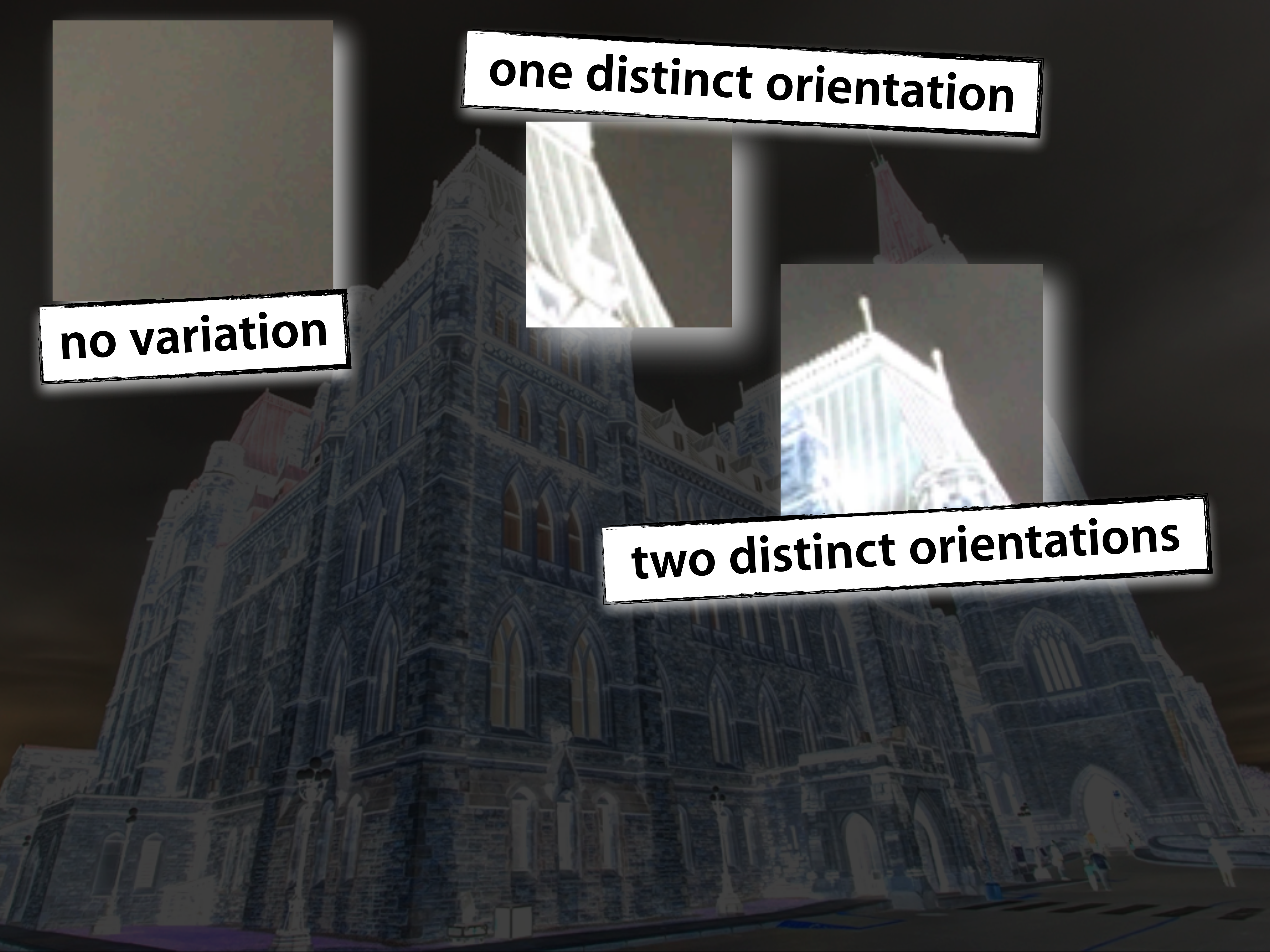
Laplacian DoG

MSER

Harris-/Hessian-Laplace

What is a corner?

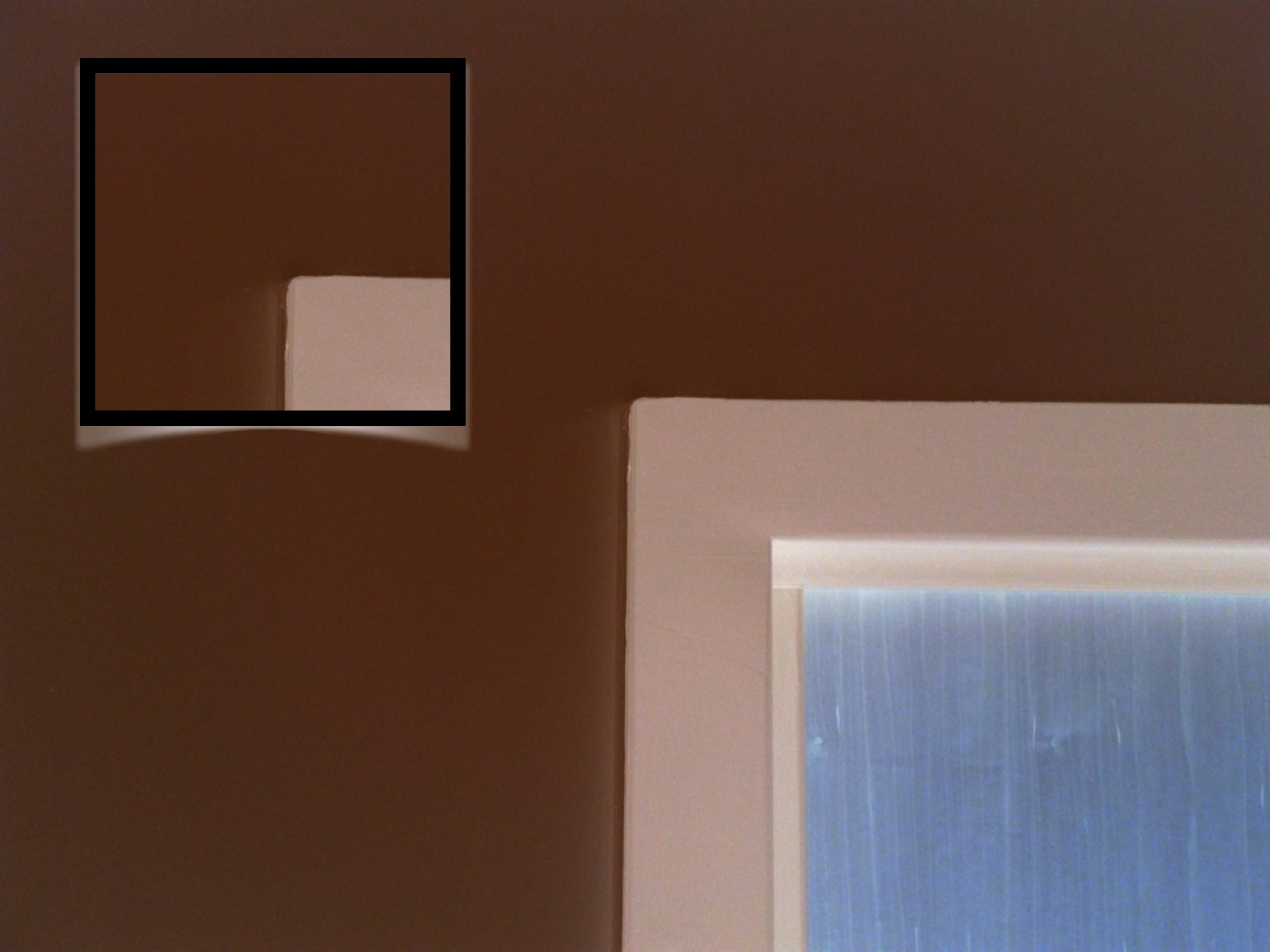
locations where two distinct
oriented structures in the image are present



no variation
one distinct orientation
two distinct orientations

Why bother with corners?

localizes spatial patterns in the image



Harris Corner Detector

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].

THE EDGE TRACKING PROBLEM

Förstner Corner Detector

ISPRS Intercommission Workshop, Interlaken, June 1987

A Fast Operator for Detection and Precise Location of
Distinct Points, Corners and Centres of Circular Features

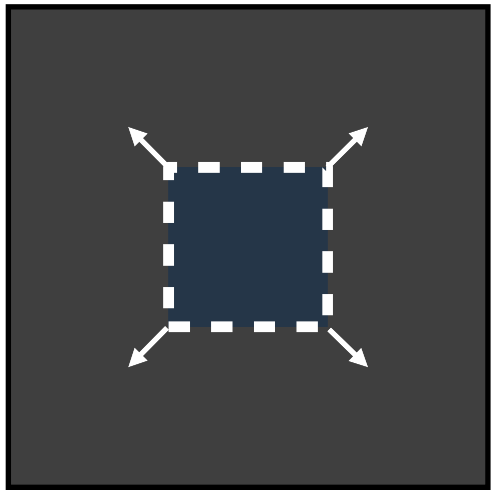
by W. Förstner and E. Gülch
Institute for Photogrammetry
Stuttgart University

Summary:

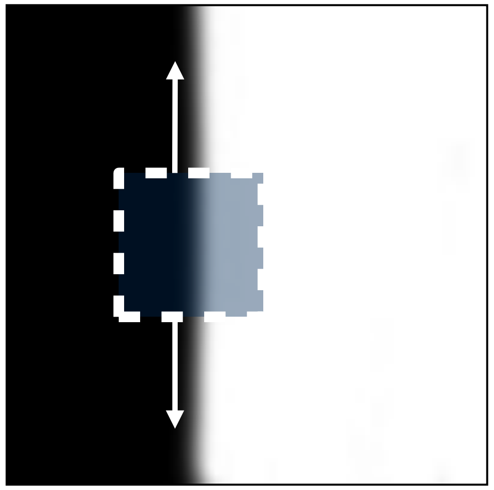
Feature extraction is a basic step for image matching and image analysis. The paper describes a fast operator for the detection and precise location of distinct points, corners and centres of circular image features. Distinct points are needed for feature based image matching or for tracking in image sequences. A special class of these distinct points are corners, which, besides edges, are the basic ele-

Intuition

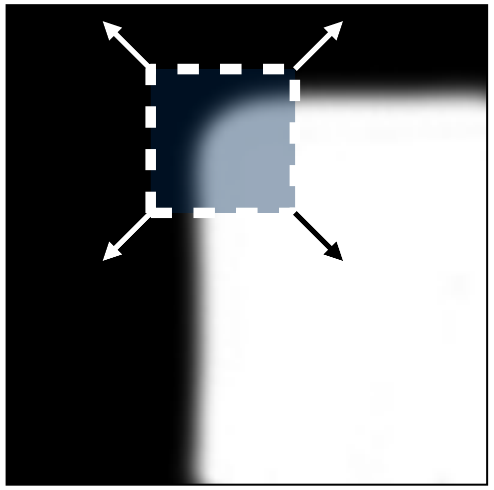
Analyze the local variation of signal



“flat” region
no change in all directions



“edge”
no change along the edge direction



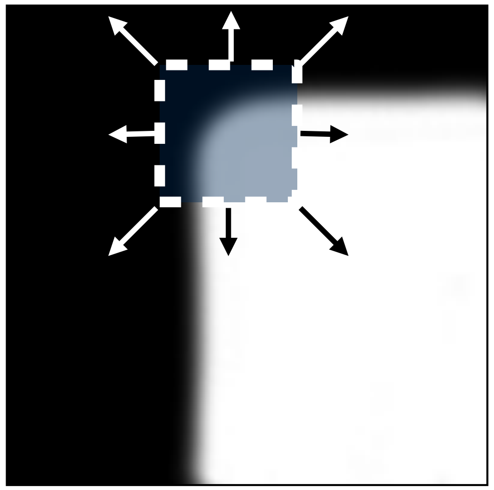
“corner”
significant change in all directions

Derivation

Variation of intensity for the shift $(\Delta x, \Delta y)$

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

image

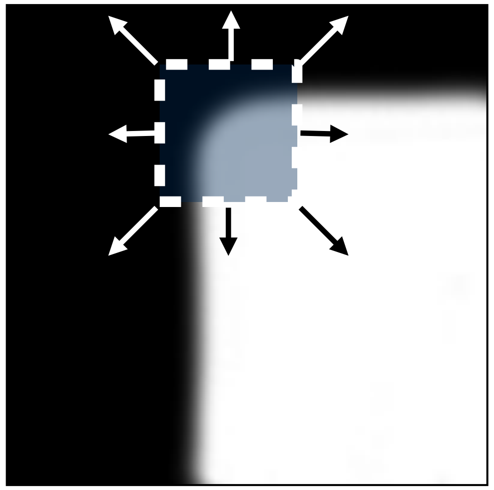


Derivation

Variation of intensity for the shift $(\Delta x, \Delta y)$

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

shifted image



Derivation

Variation of intensity for the shift $(\Delta x, \Delta y)$

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

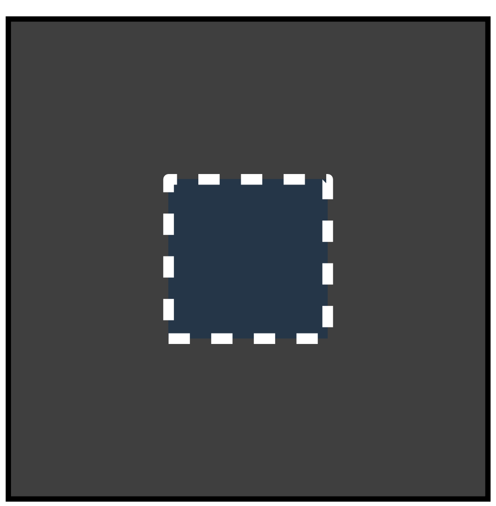
window

Possible windowing functions

Derivation

Variation of intensity for the shift $(\Delta x, \Delta y)$

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

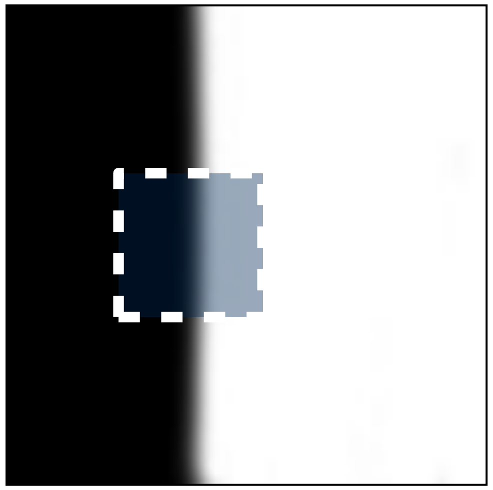


Nearly constant patches

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

Variation of intensity for the shift $(\Delta x, \Delta y)$

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



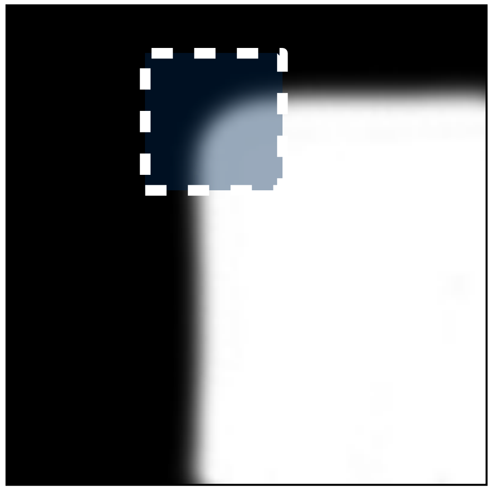
"edge" patches

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

large variation along one orientation

Variation of intensity for the shift $(\Delta x, \Delta y)$

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



“corner” patches

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

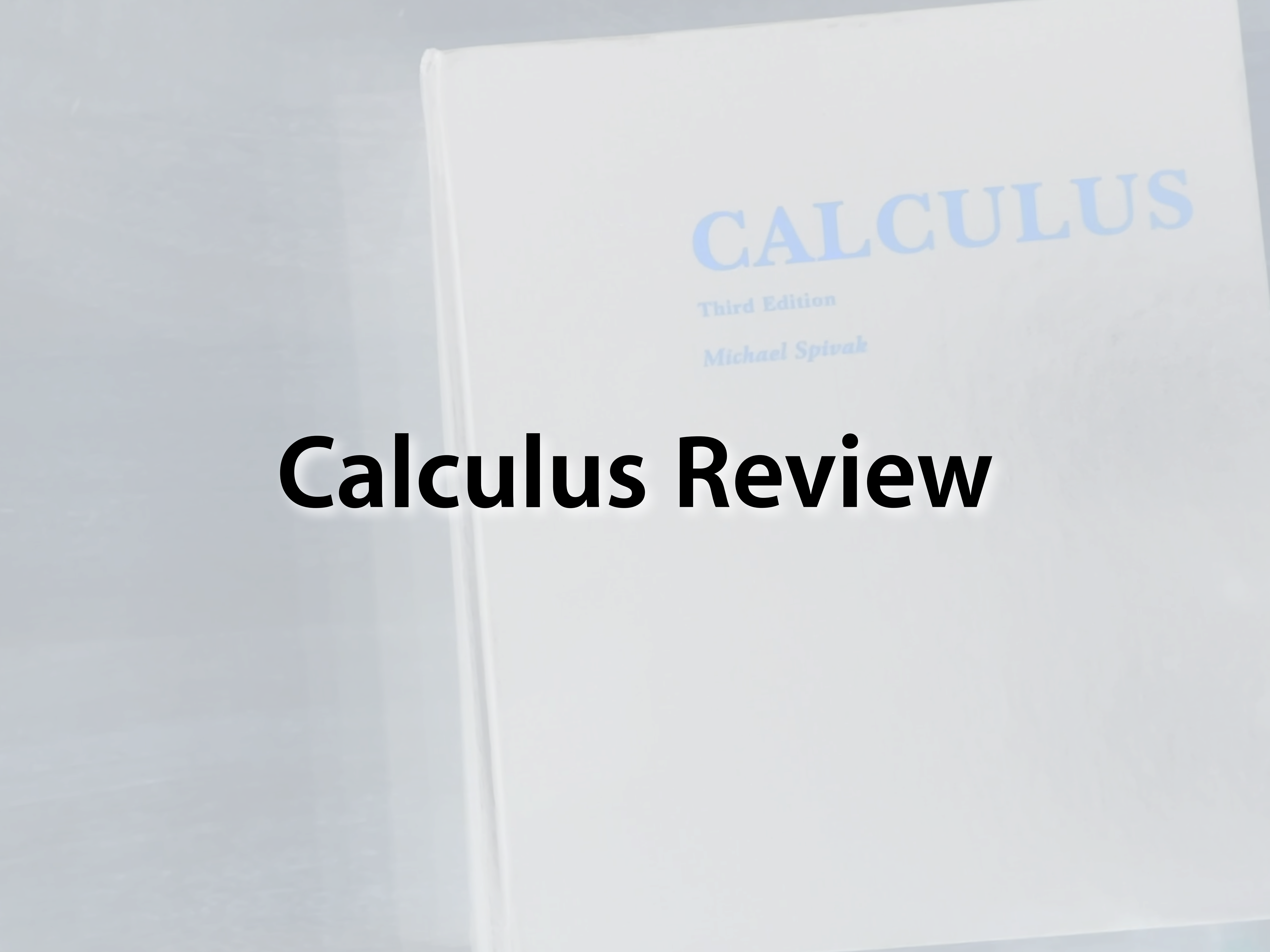
large variation along two orientations

Derivation

Variation of intensity for the shift $(\Delta x, \Delta y)$

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

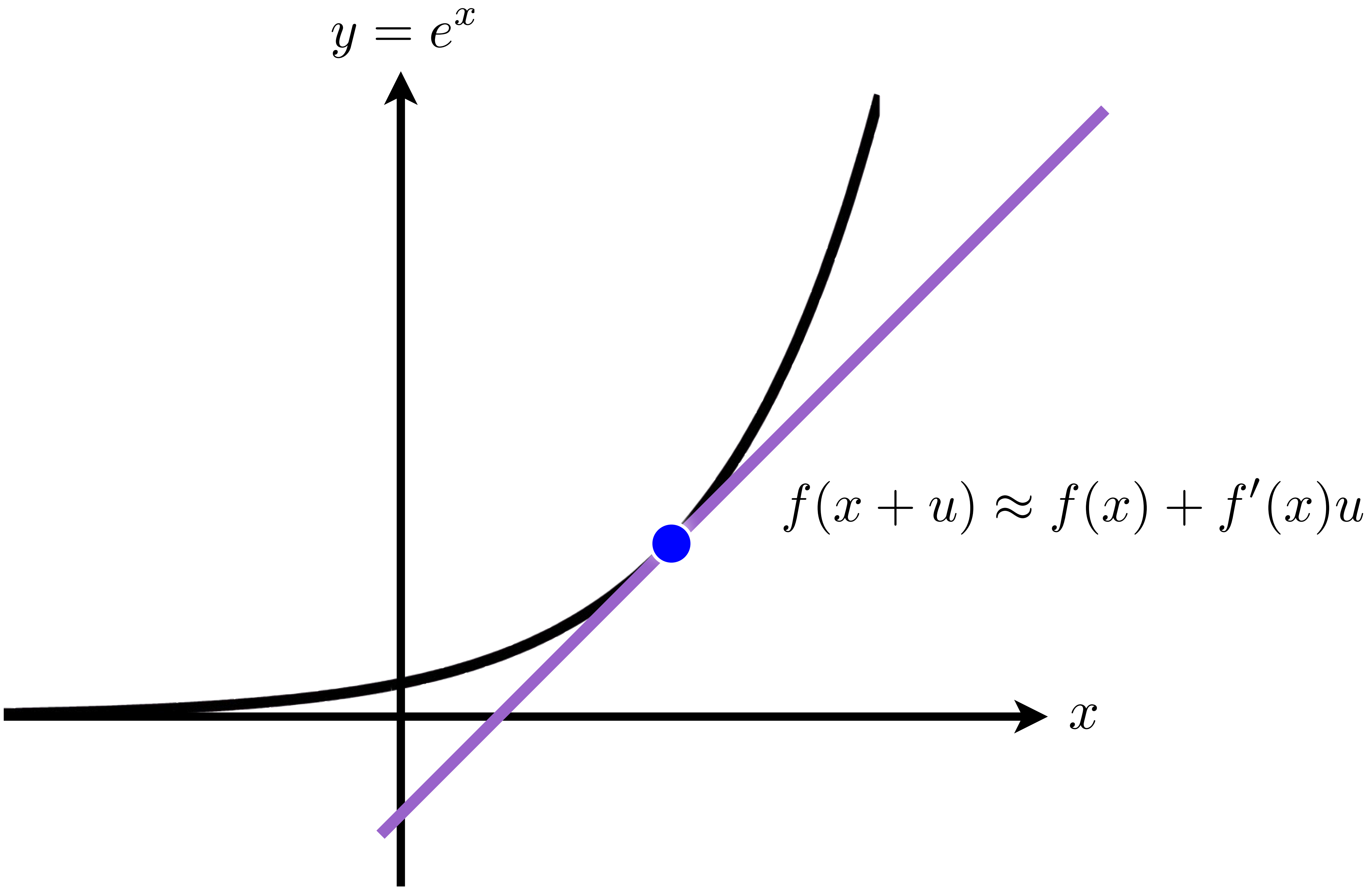
Calculus Review



1D case

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

first-order approximation



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

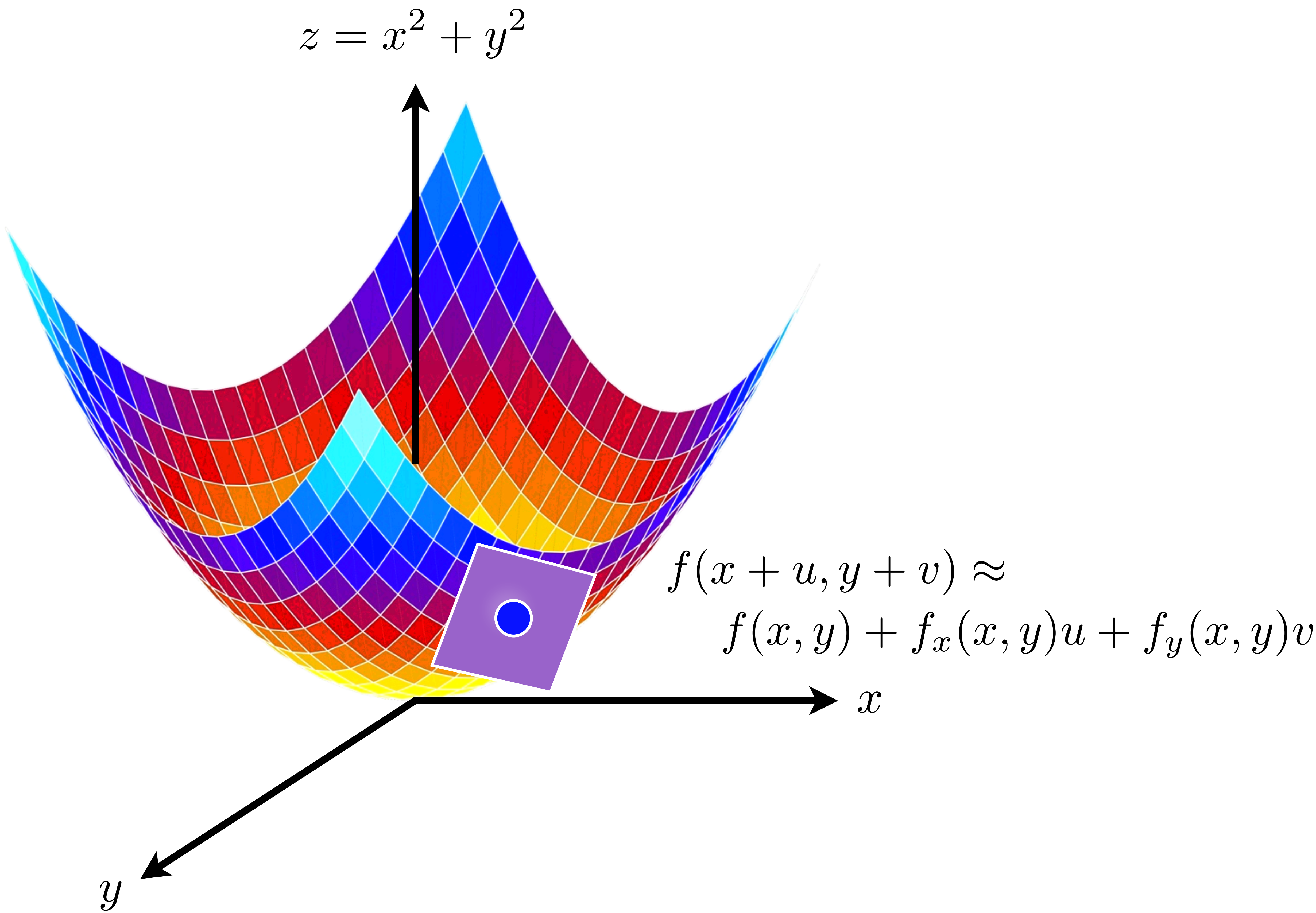
2D case

$$f(x+u, y+v) \approx f(x,y) + f_x(x,y)u + f_y(x,y)v$$

$$\cancel{+\frac{1}{2}[f_{xx}(x,y)u^2 + 2f_{xy}(x,y)uv + f_{yy}(x,y)v^2]}$$

$$\cancel{+ \text{h.o.t.}}$$

first-order approximation


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

Derivation

Variation of intensity for the shift $(\Delta x, \Delta y)$

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

Taylor series expansion

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

expand

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

rewrite as matrix

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

Taylor series expansion

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

expand

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

rewrite as matrix

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

where

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

$$\boxed{\begin{aligned} & E(\Delta x, \Delta y) = \begin{pmatrix} \Delta x & \Delta y \end{pmatrix} \mathbf{M} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \\ & \quad \text{where} \\ & \quad \mathbf{M} = \sum_{x,y} w(x,y) \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \end{aligned}}$$

Definition: A $d \times d$ matrix $\mathbf{M}$ has *eigenvalue* λ if there is a d-dimensional vector $\mathbf{u} \neq \mathbf{0}$ such that

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

This $\mathbf{u}$ is the *eigenvector* corresponding to λ.

Definition: A matrix $\mathbf{B}$ is *symmetric* if $\mathbf{B} = \mathbf{B}^\top$.

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

Spectral Decomposition

Theorem: Let $\mathbf{M}$ be a *real symmetric* $d \times d$ matrix with eigenvalues $\lambda_1, \ldots, \lambda_d$ and corresponding *orthonormal eigenvectors* $\mathbf{u}_1, \ldots, \mathbf{u}_d$. Then:

$$\mathbf{M} = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_d \end{pmatrix} \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_d \end{pmatrix} \begin{pmatrix} \mathbf{u}_1^\top \\ \mathbf{u}_2^\top \\ \vdots \\ \mathbf{u}_d^\top \end{pmatrix}$$

Theorem: Let $\mathbf{M}$ be a *real symmetric* $d \times d$ matrix with eigenvalues $\lambda_1, \ldots, \lambda_d$ and corresponding *orthonormal eigenvectors* $\mathbf{u}_1, \ldots, \mathbf{u}_d$. Then:

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

Definition: A real symmetric $d \times d$ matrix $\mathbf{M}$ **is** *positive semidefinite* if

$$\mathbf{z}^\top \mathbf{M} \mathbf{z} \geq 0$$

for all $\mathbf{z} \in \mathbb{R}^d$.

All eigenvalues of M are greater or equal to zero

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

where

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

$\mathbf{M}$ is a symmetric matrix

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

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

where

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

$\mathbf{M}$ is a symmetric matrix

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

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

where

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

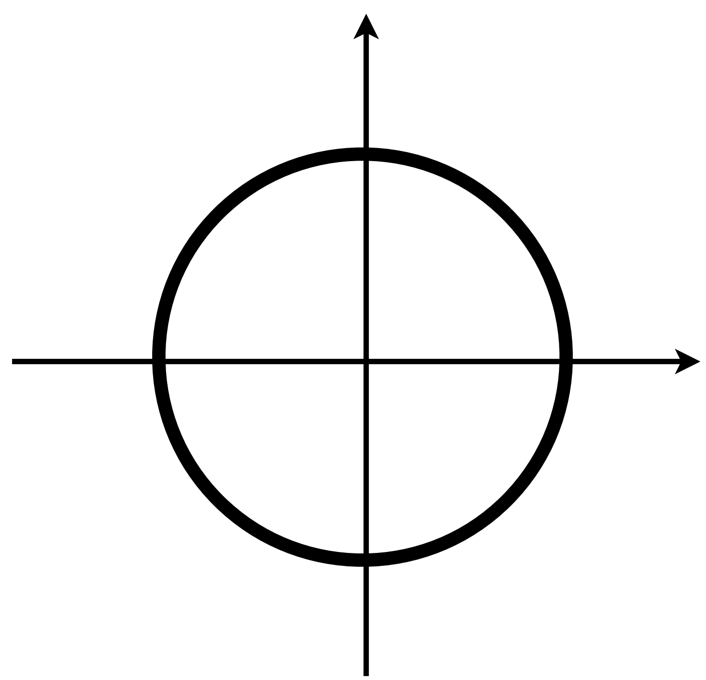
M is a symmetric matrix

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

M is positive semidefinite

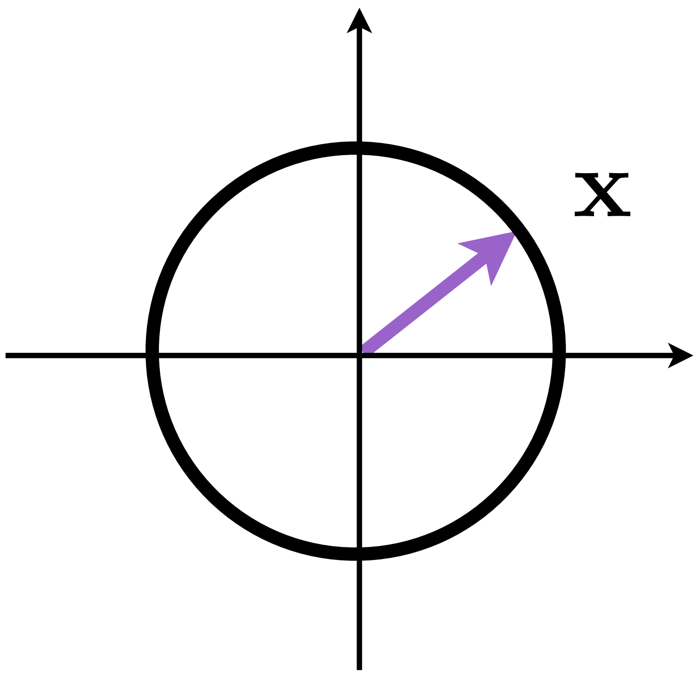
eigenvalues non-negative

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



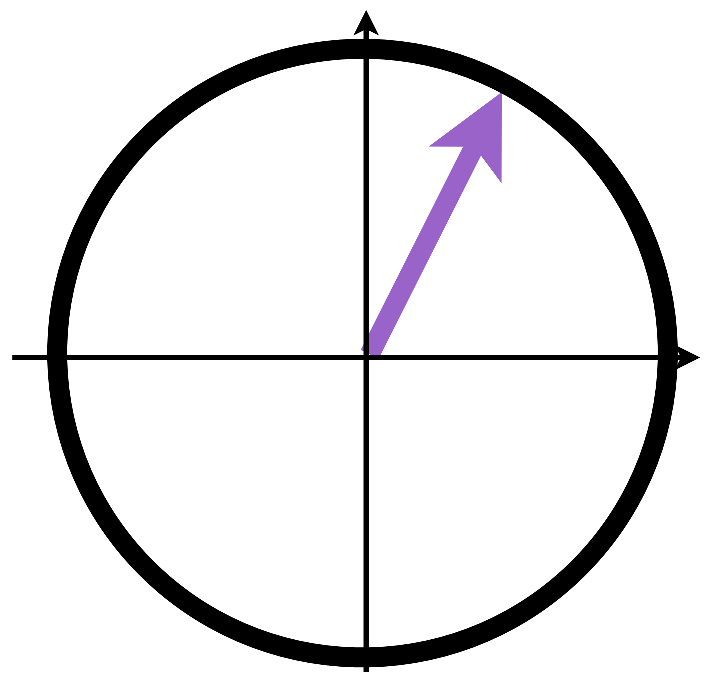
unit circle

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



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

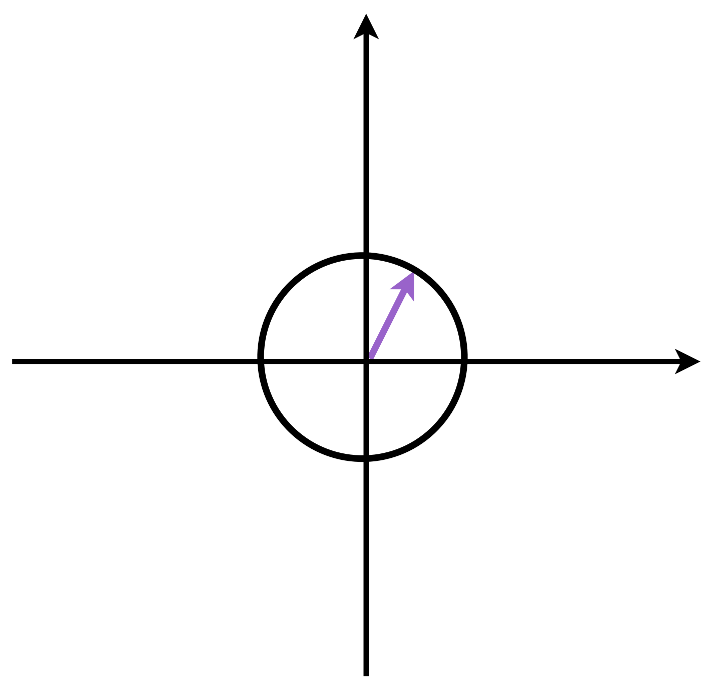
$$\mathbf{x}^\top \mathbf{R} \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \mathbf{R}^\top \mathbf{x}$$



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

if $\lambda_1 = \lambda_2$ **and large**

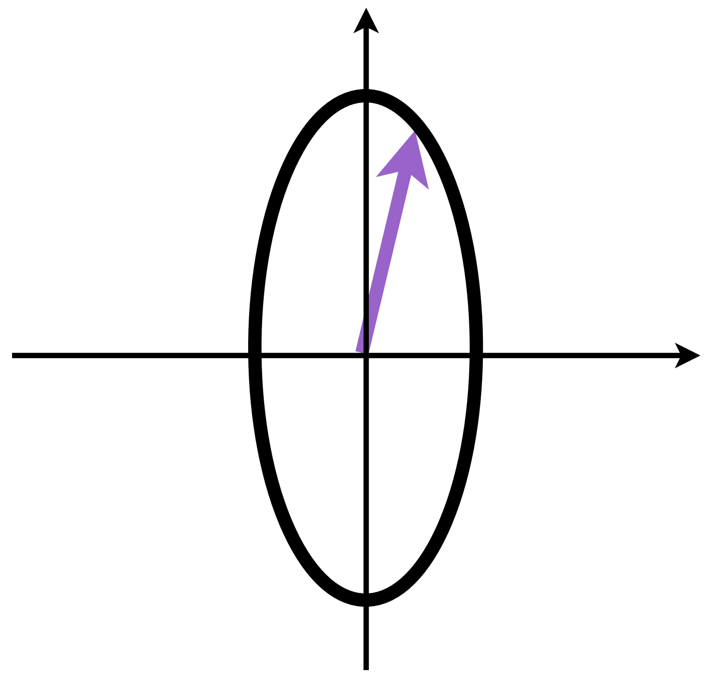
$$\mathbf{x}^\top \mathbf{R} \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \mathbf{R}^\top \mathbf{x}$$



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

if $\lambda_1 = \lambda_2$ **and small**

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



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

if $\lambda_2 \gg \lambda_1$ **and large**

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

Have we seen this before?

transpose of right side

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

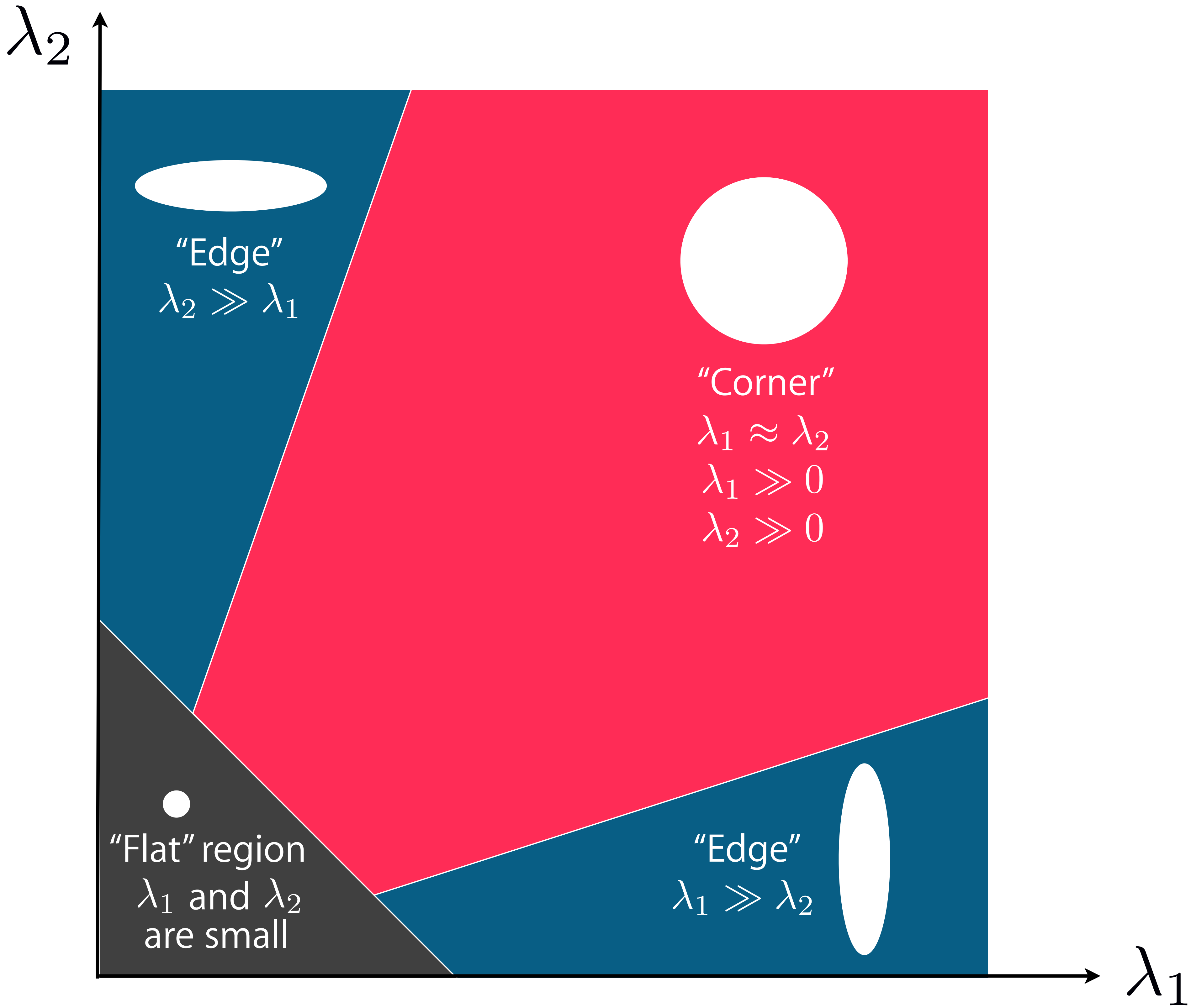
Inner product (distance squared)

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

Inner product (distance squared)

Eigenvalues indicate whether a patch is a corner

Image point classification using eigenvalues of M



Computing the eigenvalues for each image point is computationally expensive

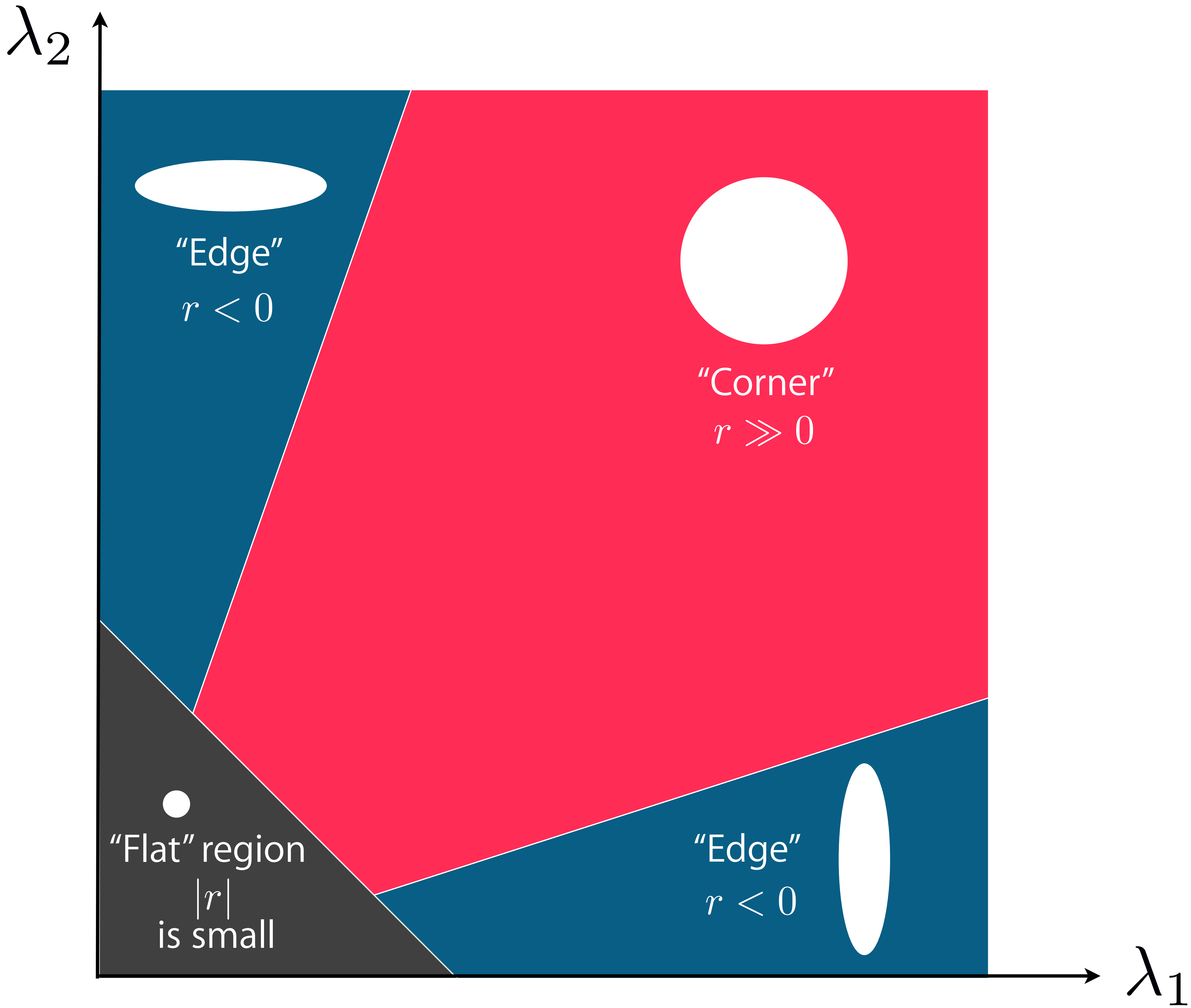
Harris Corner Response Function

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

k is empirically set constant: *0.04 - 0.06*

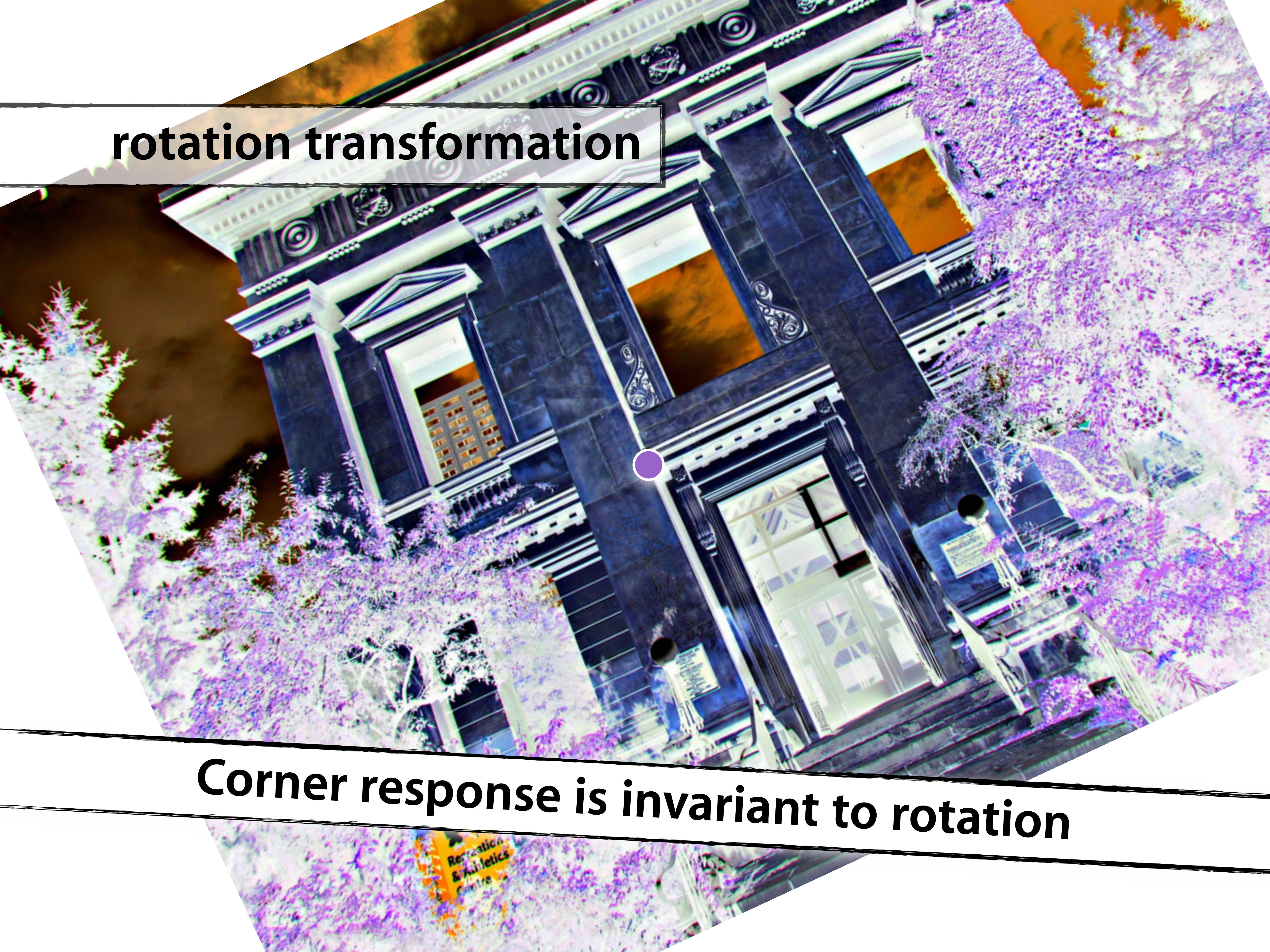
$$\det \mathbf{M} = \lambda_1 \lambda_2 \qquad \text{trace}\, \mathbf{M} = \lambda_1 + \lambda_2$$

Image point classification using r

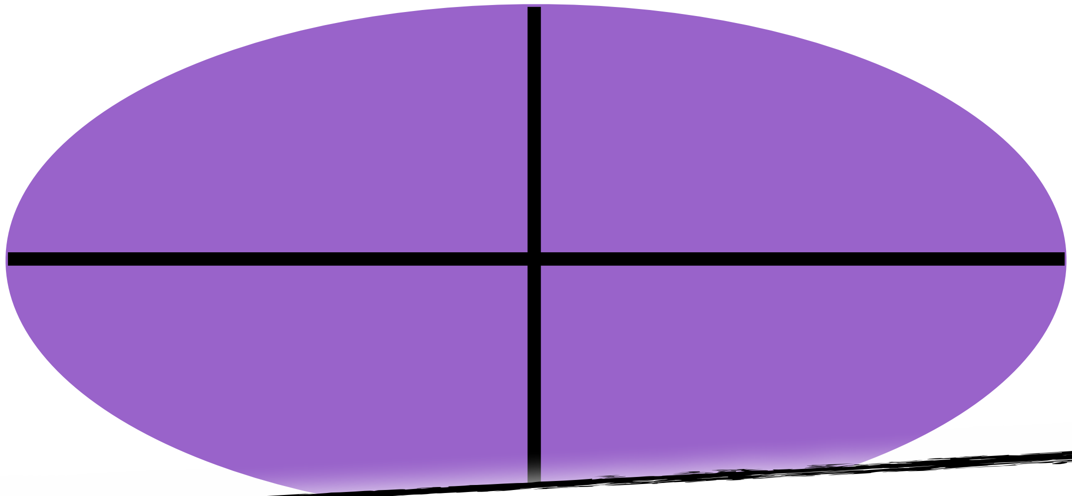


Rotation invariant

rotation transformation



Corner response is invariant to rotation



the shape remained the same

Local Intensity Variation

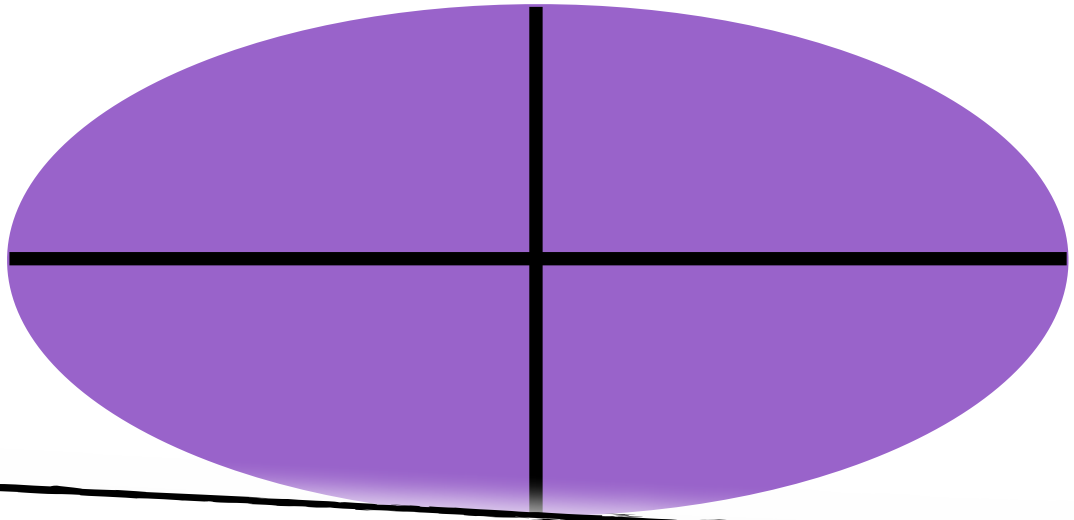
$$E(\Delta x, \Delta y) = \begin{pmatrix} \Delta x & \Delta y \end{pmatrix} \mathbf{M} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

where

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

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

Rotation Invariant?



What else remained the same?

eigenvalues

Harris Corner Properties

Rotation invariant

Partially invariant to illumination variation

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

Intensity Invariant?

Local Intensity Variation

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

where

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

image derivatives

intensity shift

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

Corner response is a function of derivatives

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

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

Invariant to intensity shift?

YES

intensity scale

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

Corner response is a function of derivatives

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

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

Invariant to intensity scale?

NO

intensity scale

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

Corner response is a function of derivatives

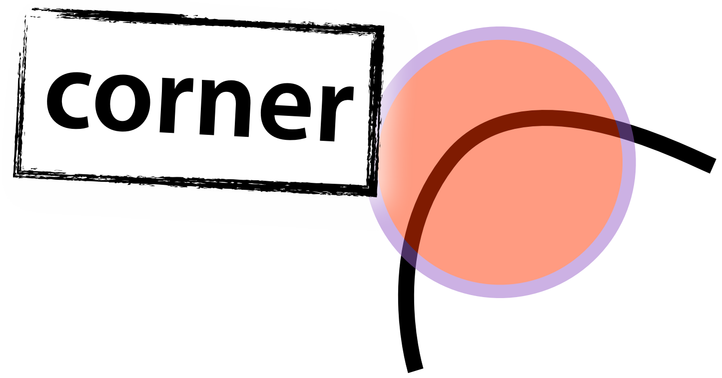
$$\frac{\partial I_{\text{new}}(x,y)}{\partial x} = \alpha \frac{\partial I(x,y)}{\partial x}$$

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

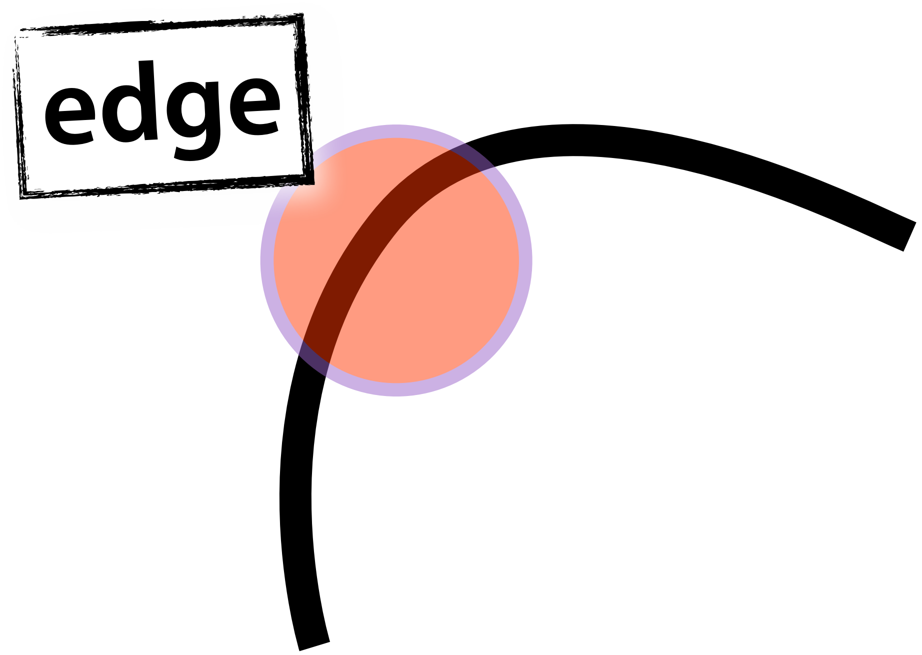
Invariant to intensity scale?

adjust corner threshold

NOT scale invariant



NOT scale invariant



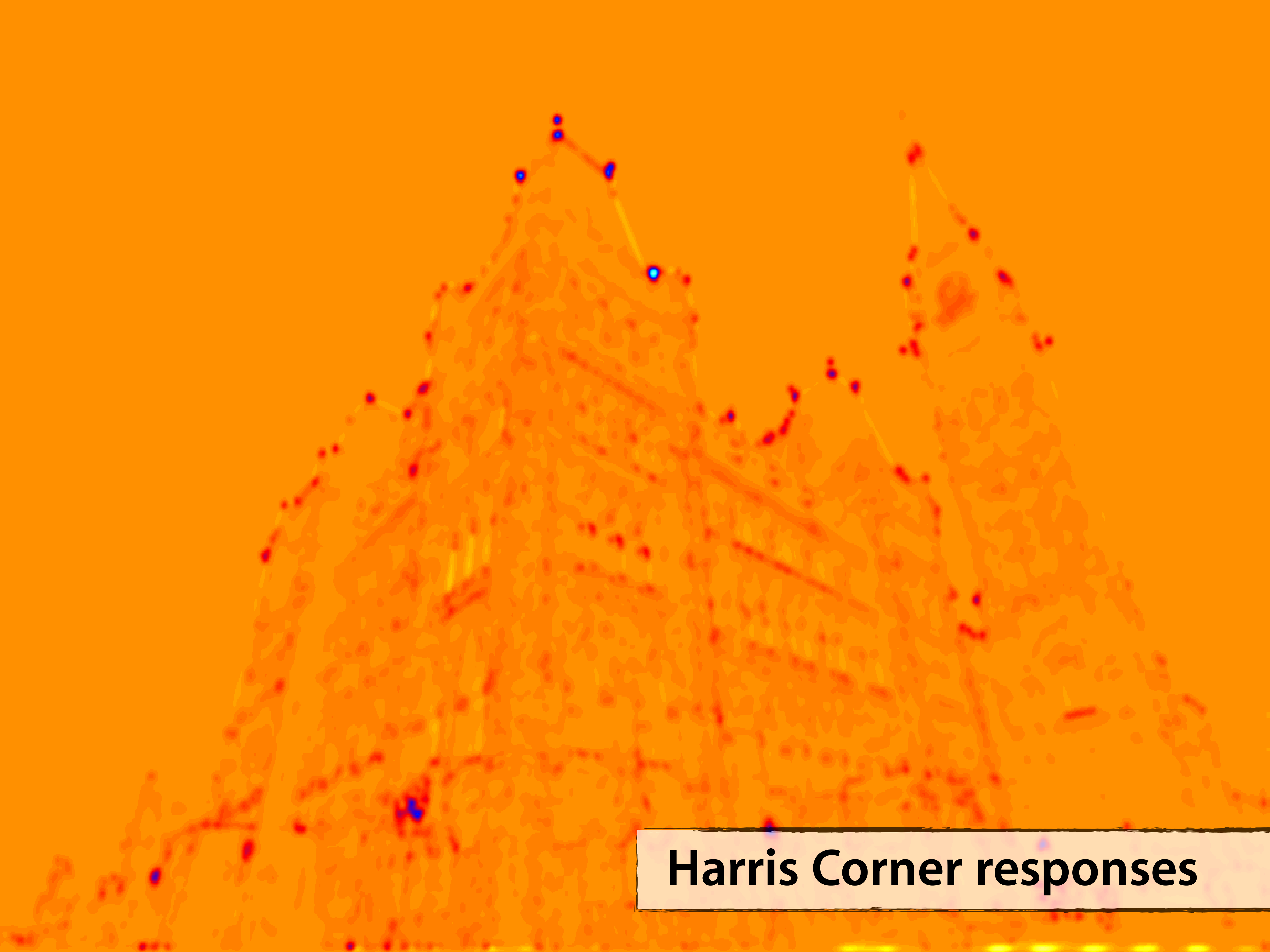
Harris Corner Properties

Rotation invariant

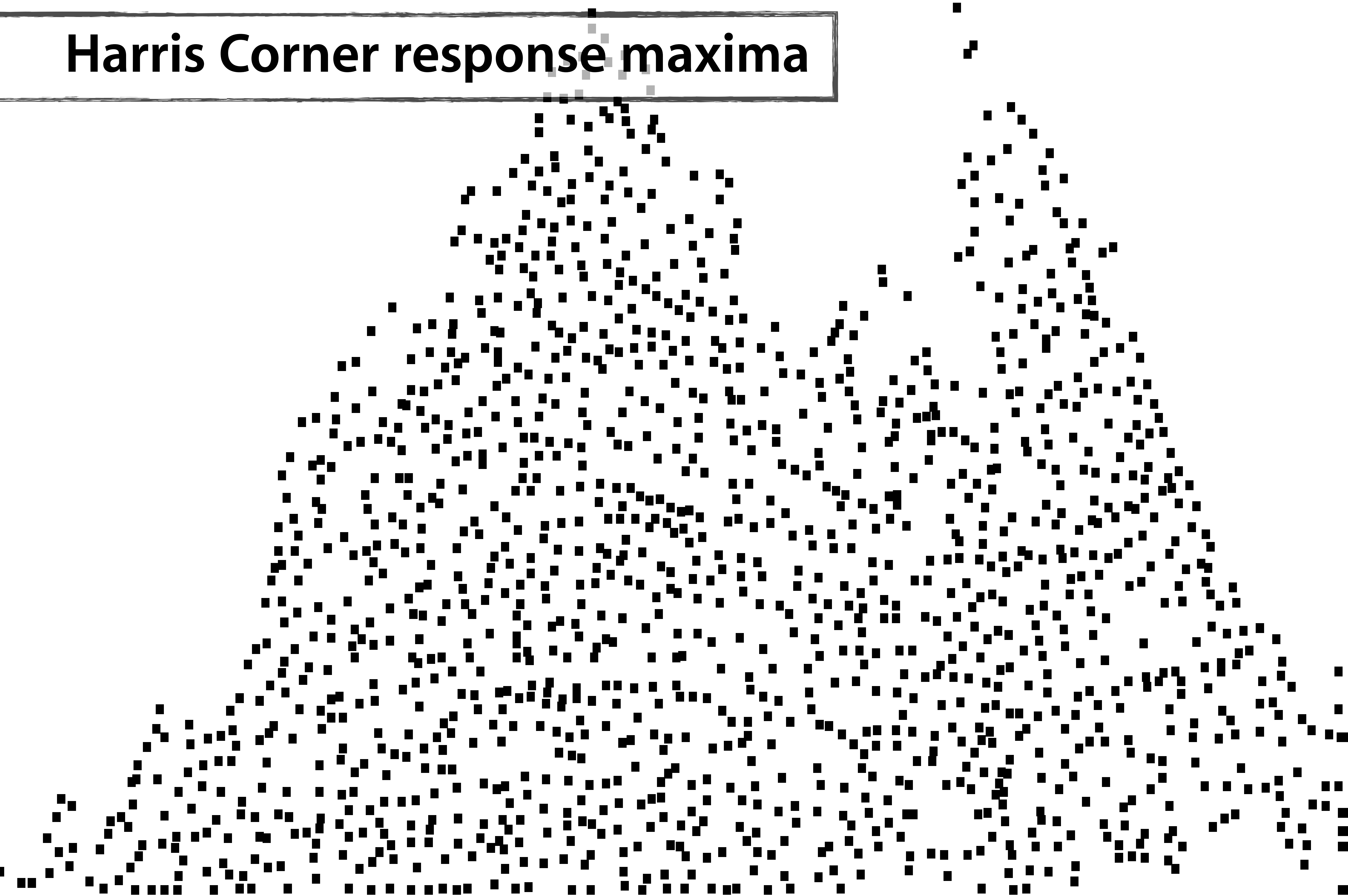
Partially invariant to illumination variation

NOT scale invariant

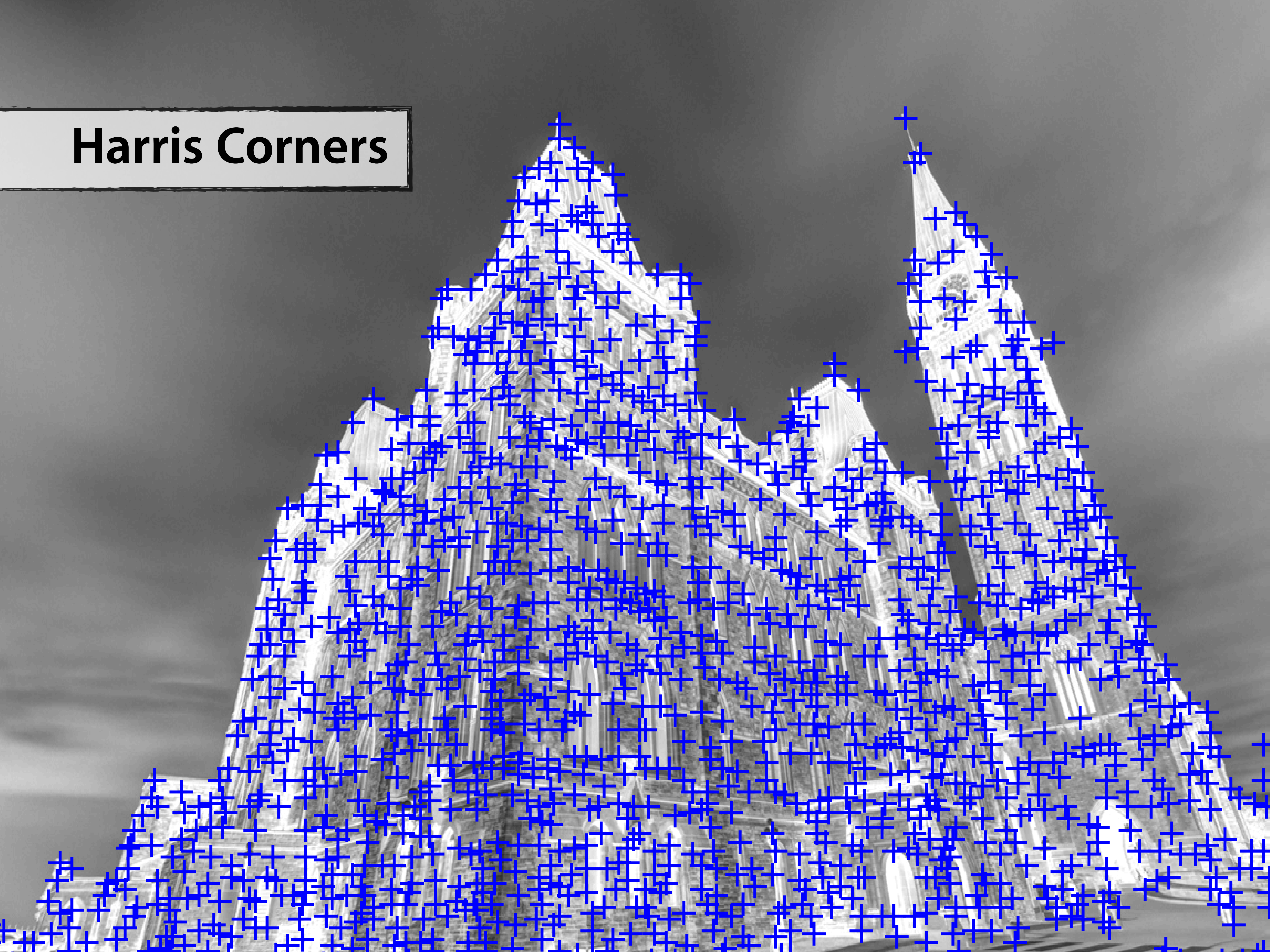

input image


Harris Corner responses

Harris Corner response maxima



Harris Corners



Harris Corner Detector

```
im = imread('parliament.jpg');
im = double(rgb2gray(im))/255;

sigma = 5;
g = fspecial('gaussian', 2*sigma*3+1, sigma);
dx = [-1 0 1;-1 0 1; -1 0 1];

Ix = imfilter(im, dx,  'symmetric', 'same');
Iy = imfilter(im, dx', 'symmetric', 'same');

Ix2 = imfilter(Ix.^2,  g, 'symmetric', 'same');
Iy2 = imfilter(Iy.^2,  g, 'symmetric', 'same');
Ixy = imfilter(Ix.*Iy, g, 'symmetric', 'same');

k = 0.04; % k is between 0.04 and 0.06
% --- r = Det(M) - kTrace(M)^2 ---
r = (Ix2.*Iy2 - Ixy.^2) - k*(Ix2 + Iy2).^2;

% --- find local maxima and threshold ---
```

```
im = imread('parliament.jpg');
im = double(rgb2gray(im))/255;

sigma = 5;
g = fspecial('gaussian', 2*sigma*3+1, sigma);
dx = [-1 0 1;-1 0 1; -1 0 1];

Ix = imfilter(im, dx,  'symmetric', 'same');
Iy = imfilter(im, dx', 'symmetric', 'same');

Ix2 = imfilter(Ix.^2,  g, 'symmetric', 'same');
Iy2 = imfilter(Iy.^2,  g, 'symmetric', 'same');
Ixy = imfilter(Ix.*Iy, g, 'symmetric', 'same');

r = (Ix2.*Iy2 - Ixy.^2) - k*(Ix2 + Iy2).^2;

% --- find local maxima and threshold ---
```

weighted sum of Harris matrix elements

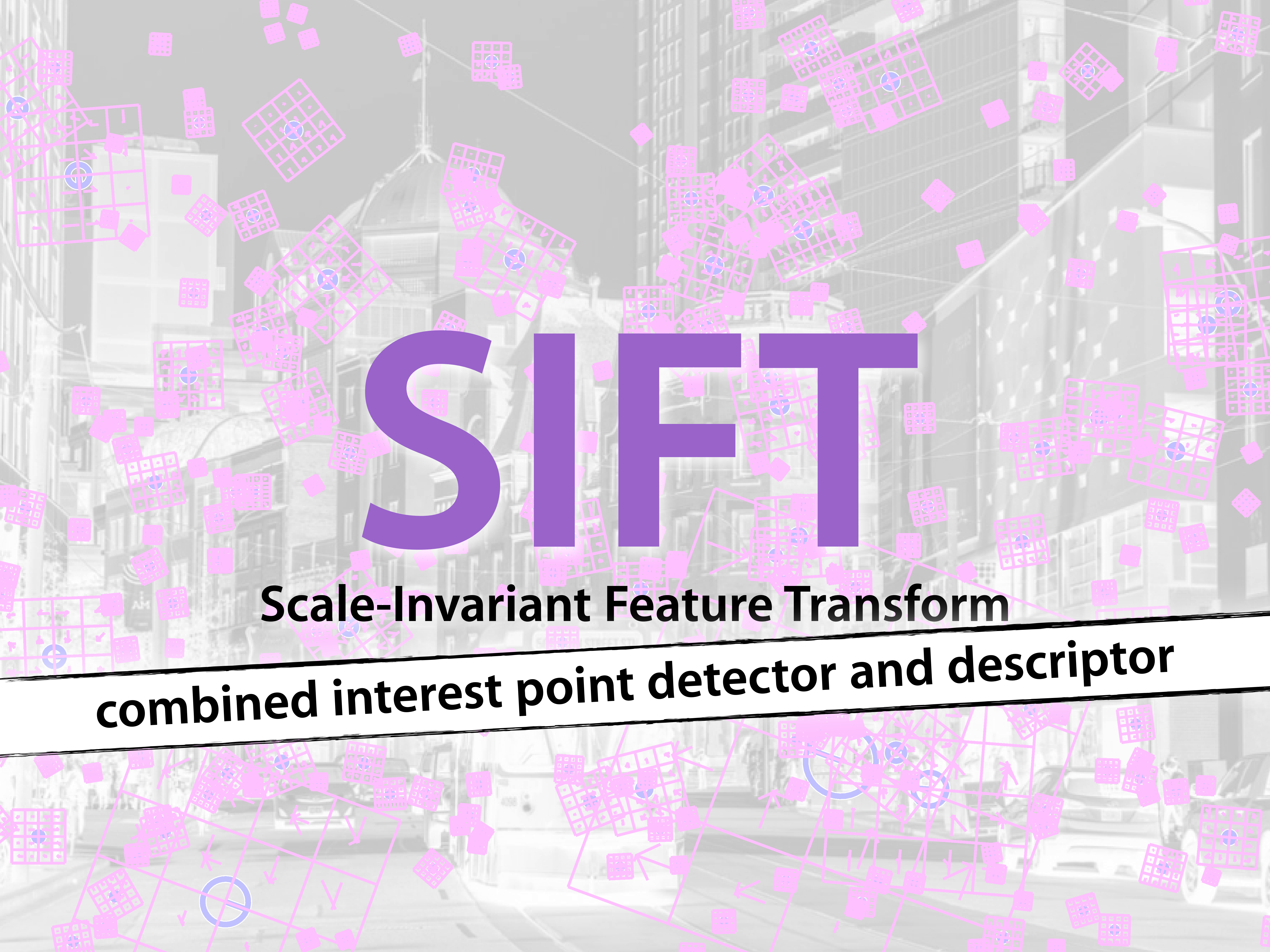
Harris Corner

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

where

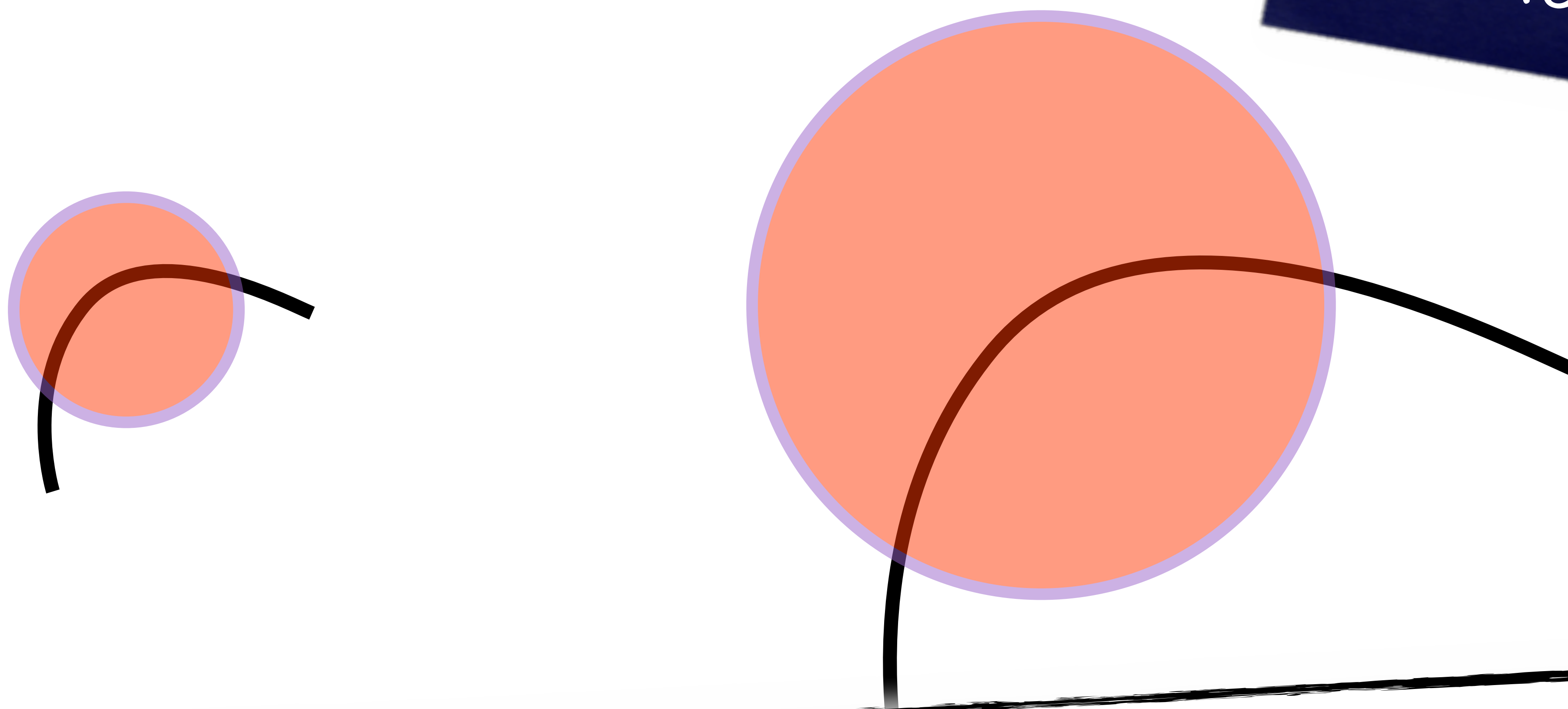
$$\mathbf{M} = \sum_{x,y} w(x,y) \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

$$\begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$


SIFT
Scale-Invariant Feature Transform
combined interest point detector and descriptor

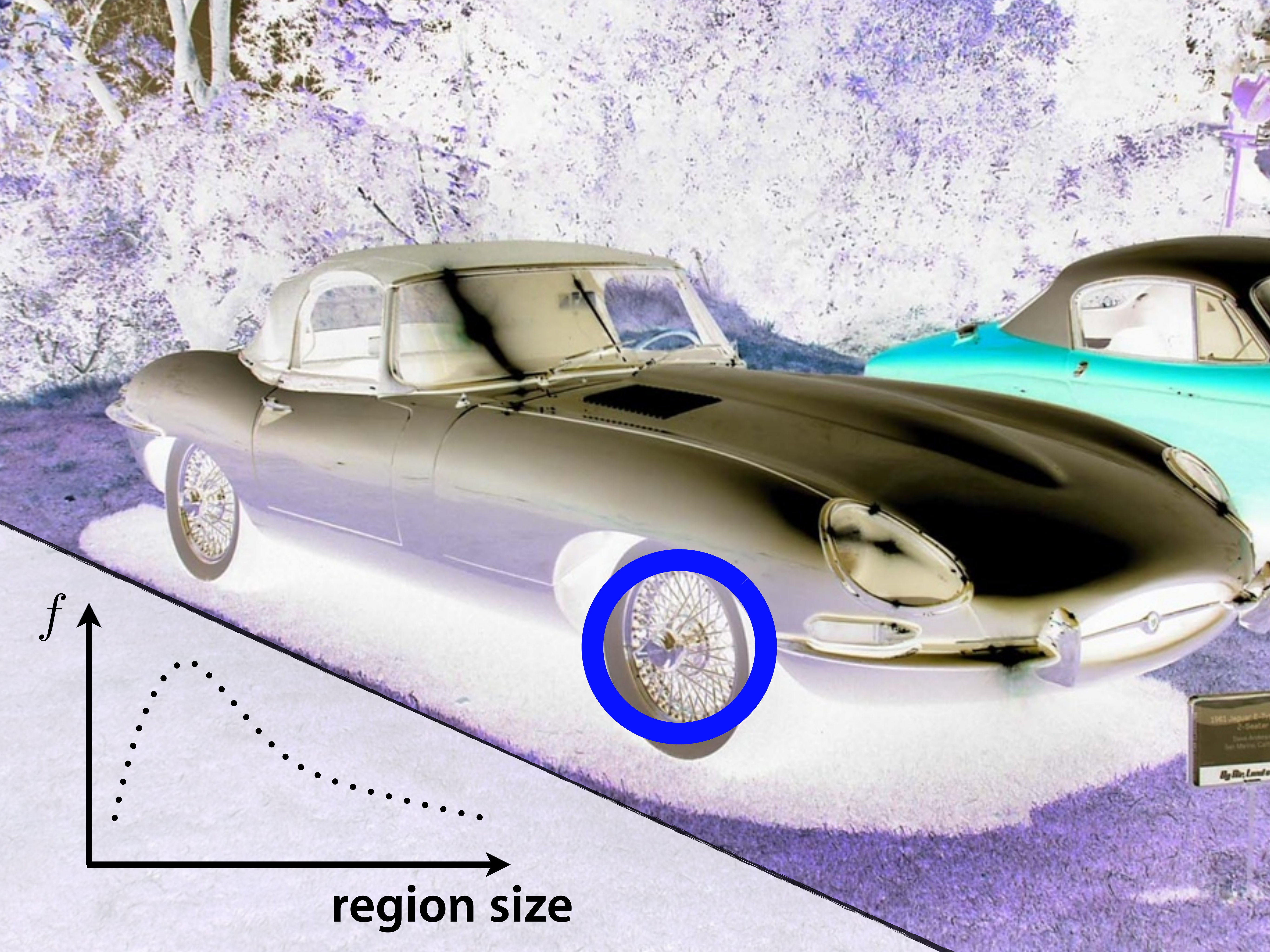

What stuff in an image matches with stuff in another?

Scale Invariant Detection

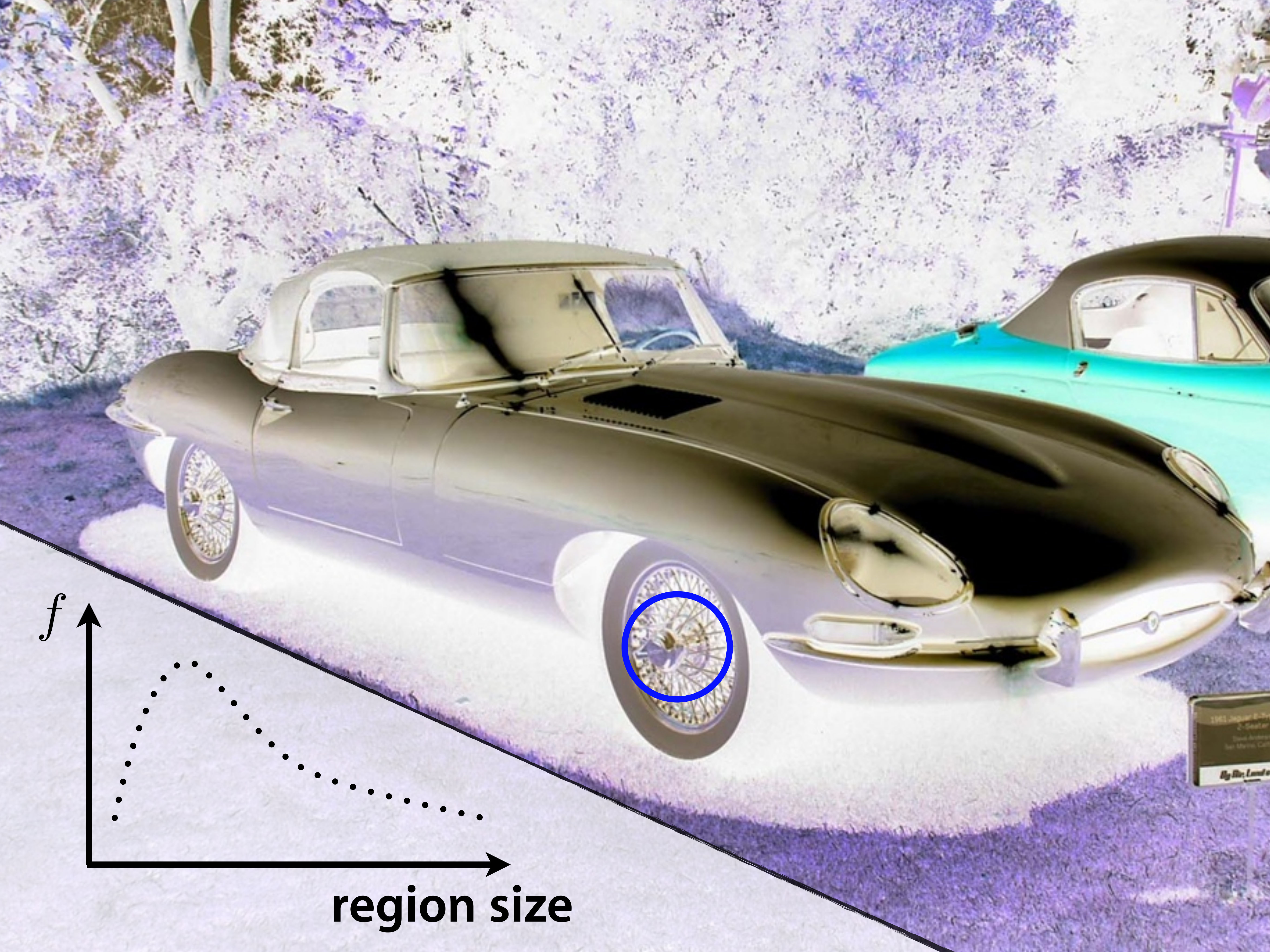


How do we choose corresponding correct region sizes **independently** in each image?

design a **scale invariant function** on the region



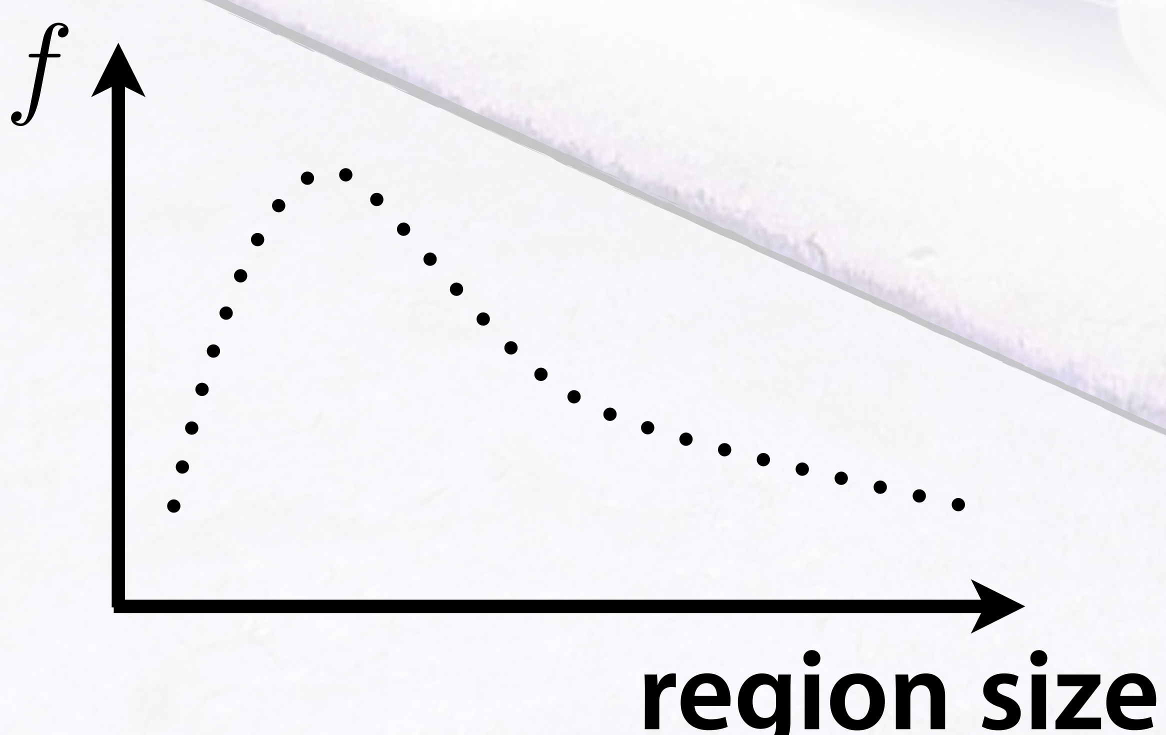
f
region size

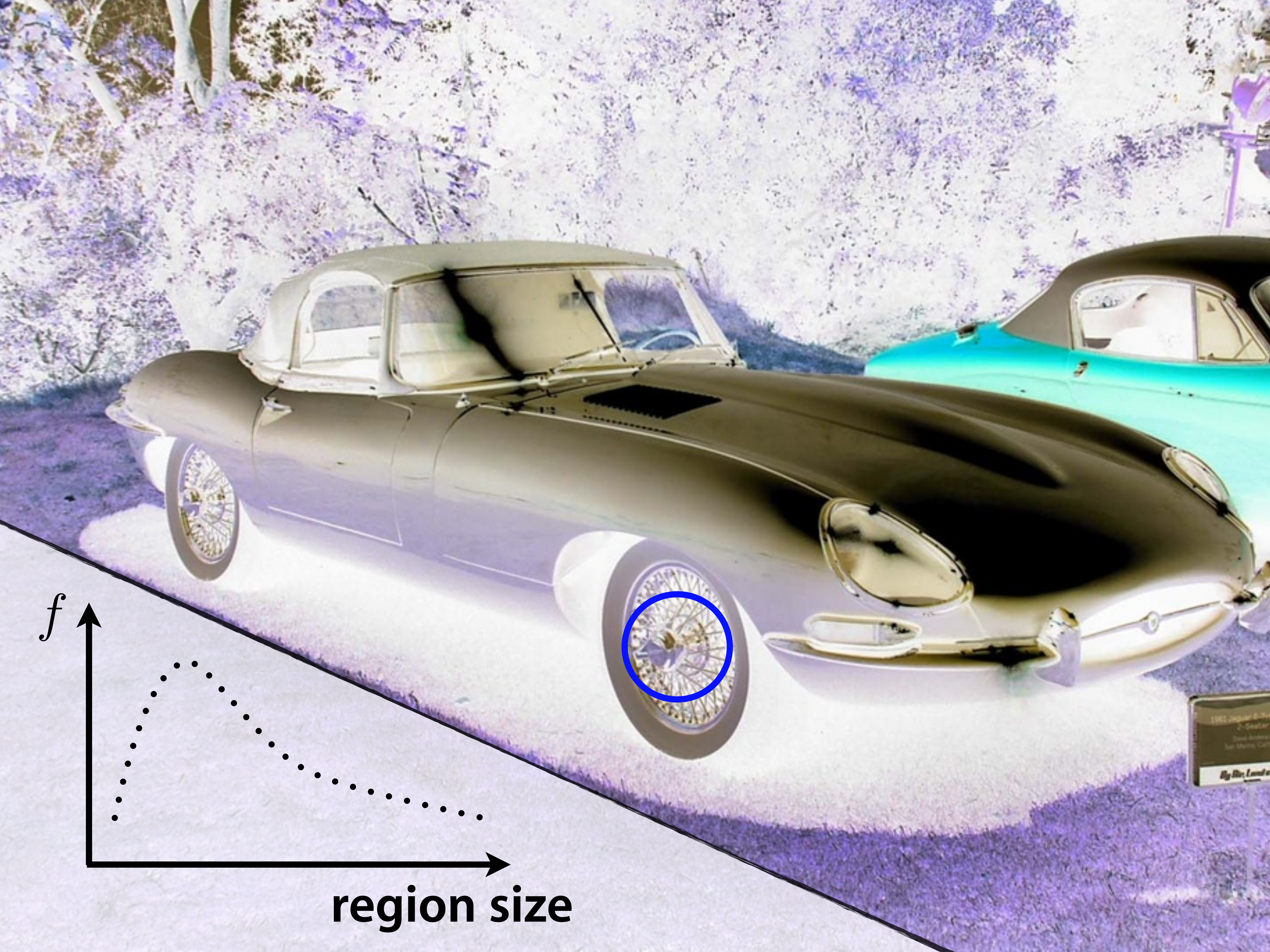


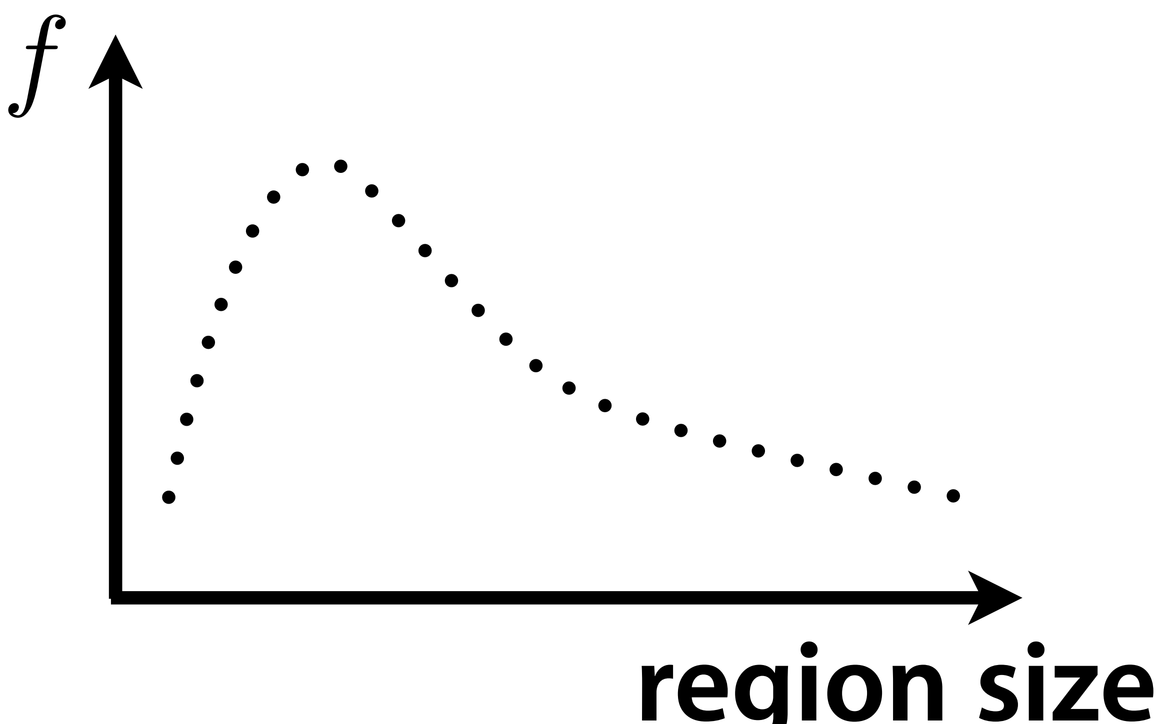
f
region size

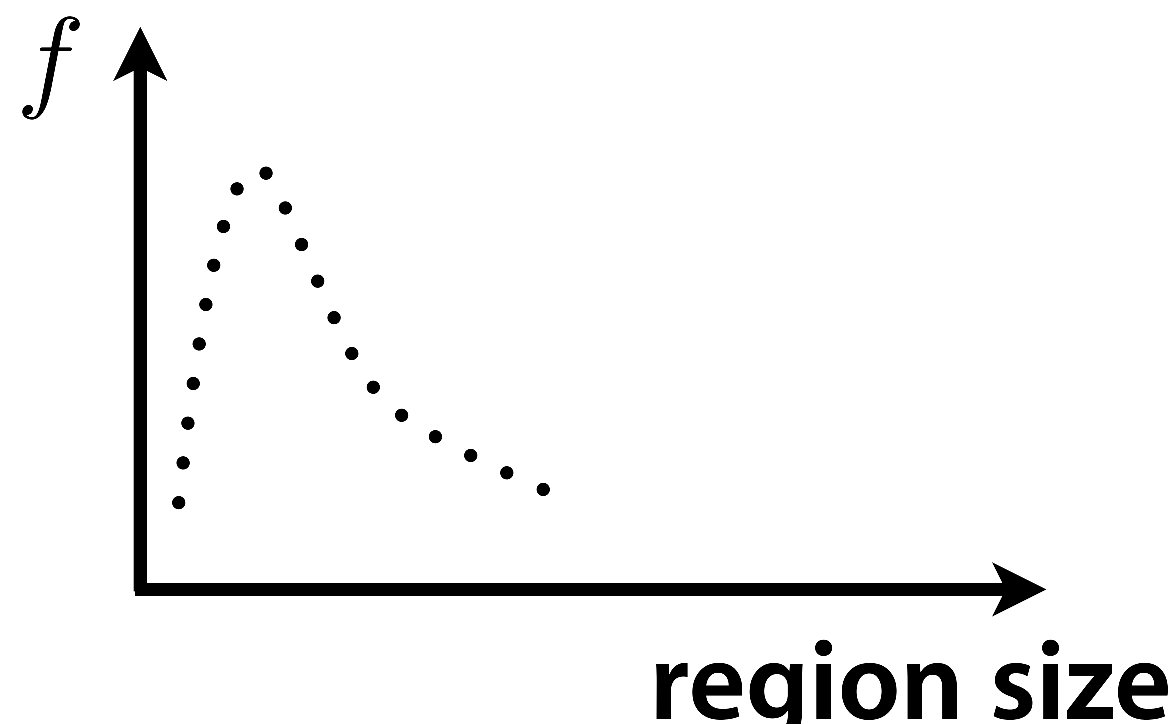
Characteristic Scale

peak response feature detector




f
region size


f
region size


f
region size

Scale Invariant Description

image 1

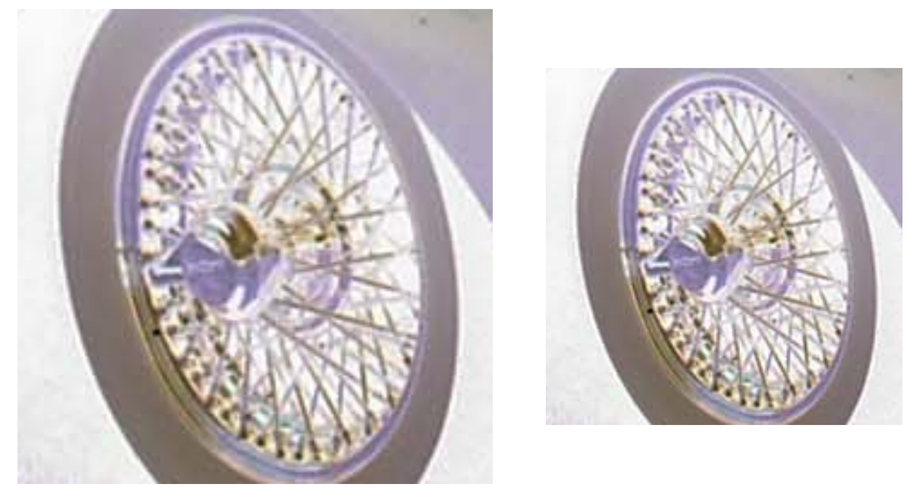


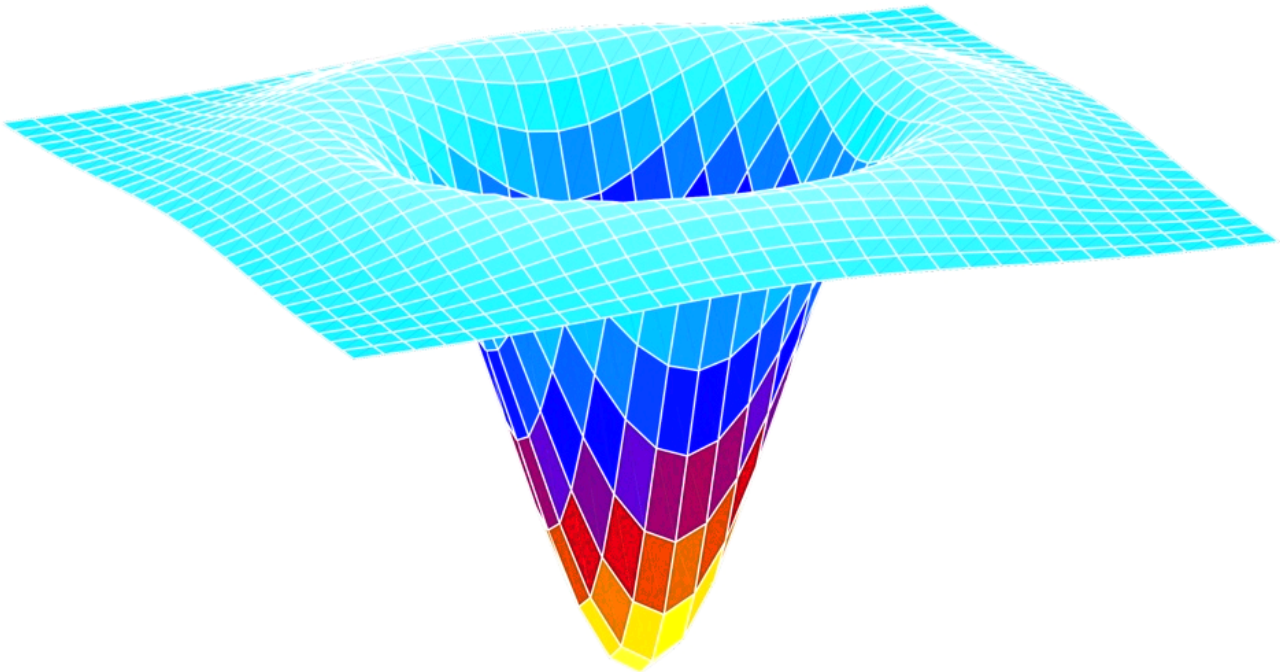
image 2

resample image about keypoint to a canonical size

Which **feature detector?**

$$f = \mathrm{Kernel} * \mathrm{Image}$$

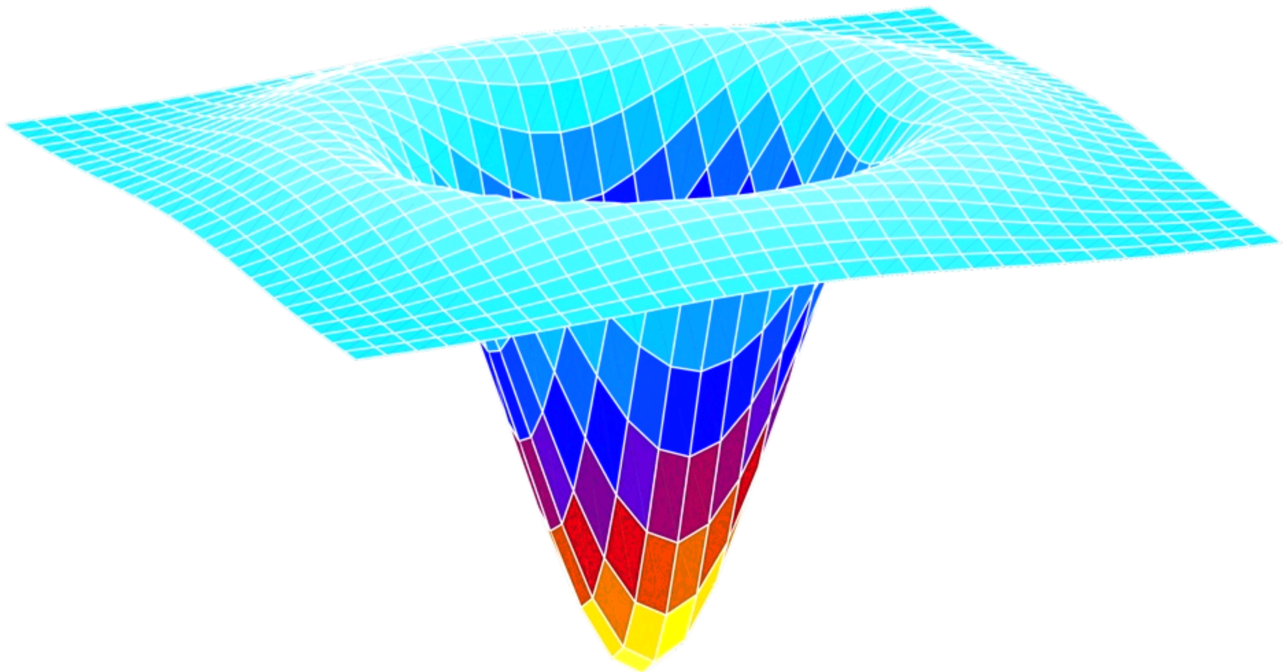
Mexican hat operator



Laplacian of Gaussian (LoG)

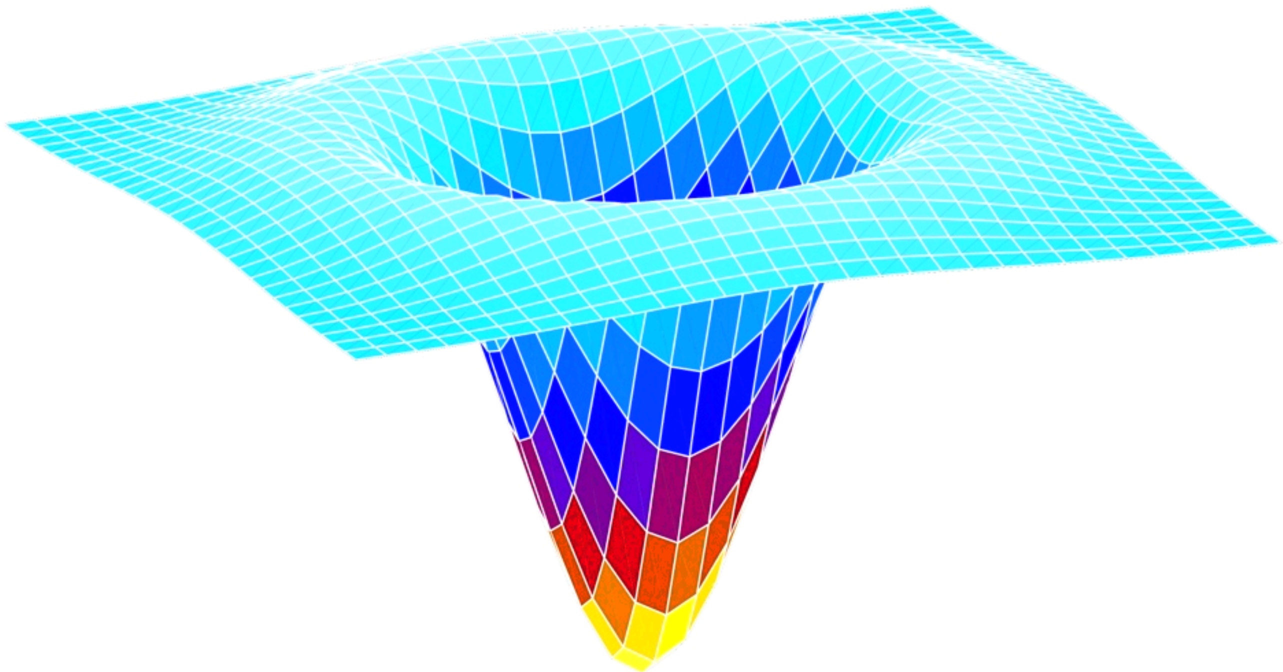
What pattern does this filter detect?

blob



Laplacian of Gaussian (LoG)

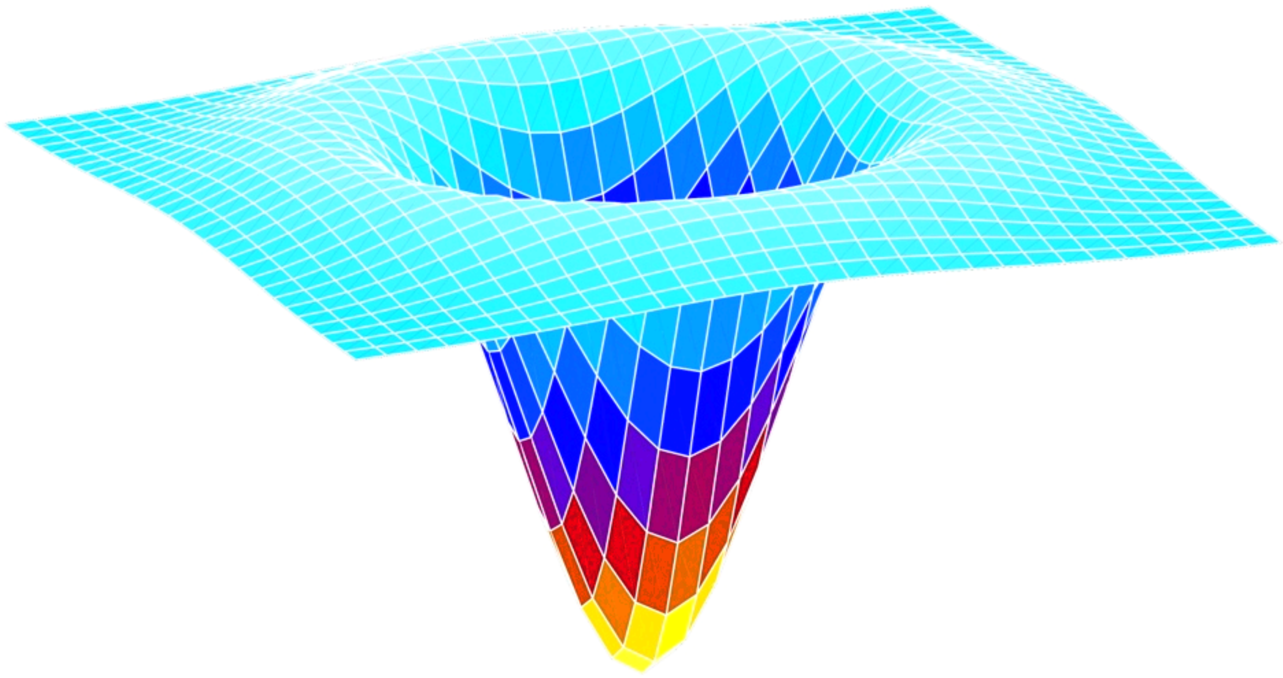
$$L(x, y; \sigma) = \sigma^2 (G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma))$$



Laplacian of Gaussian (LoG)

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

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

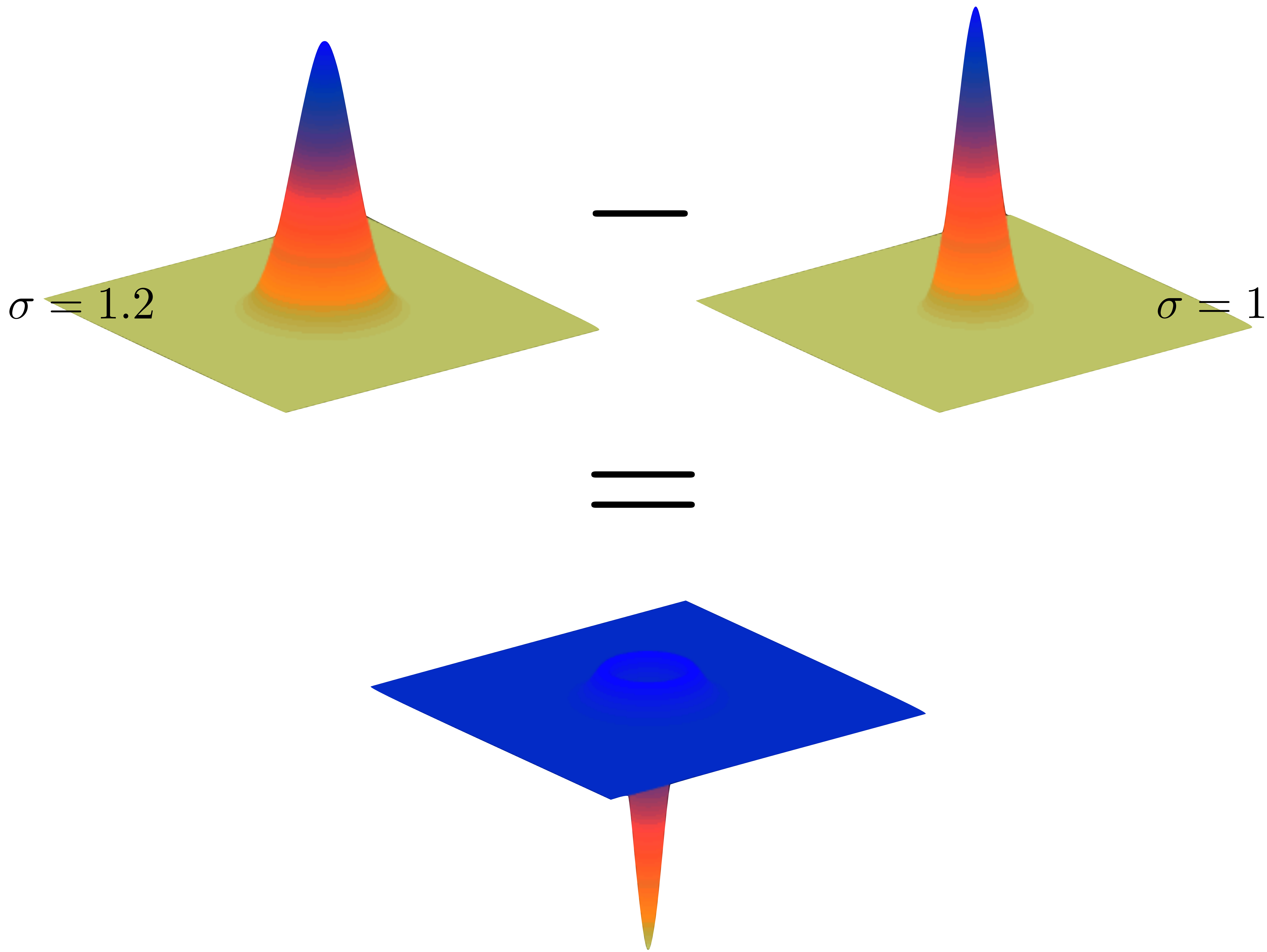


Laplacian of Gaussian (LoG)

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

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

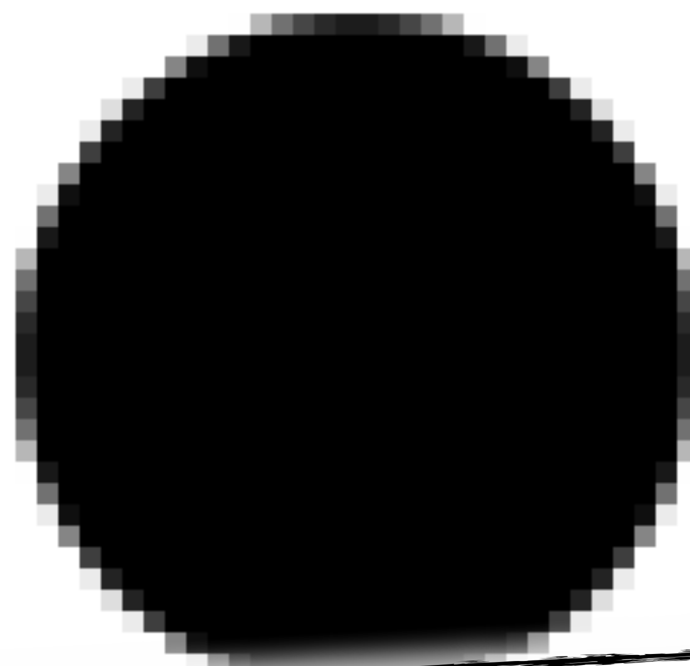
Difference of Gaussians (DoG)



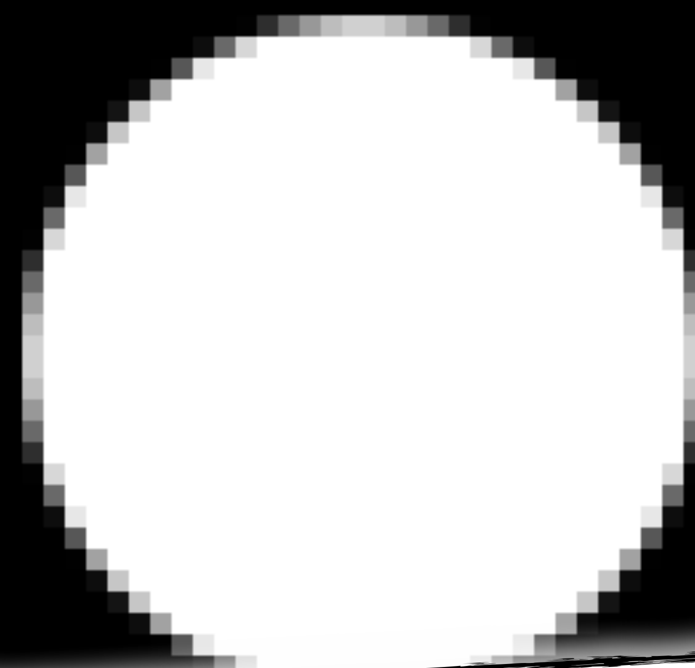
$\sigma = 1.2$
$\sigma = 1$

Blob Detection

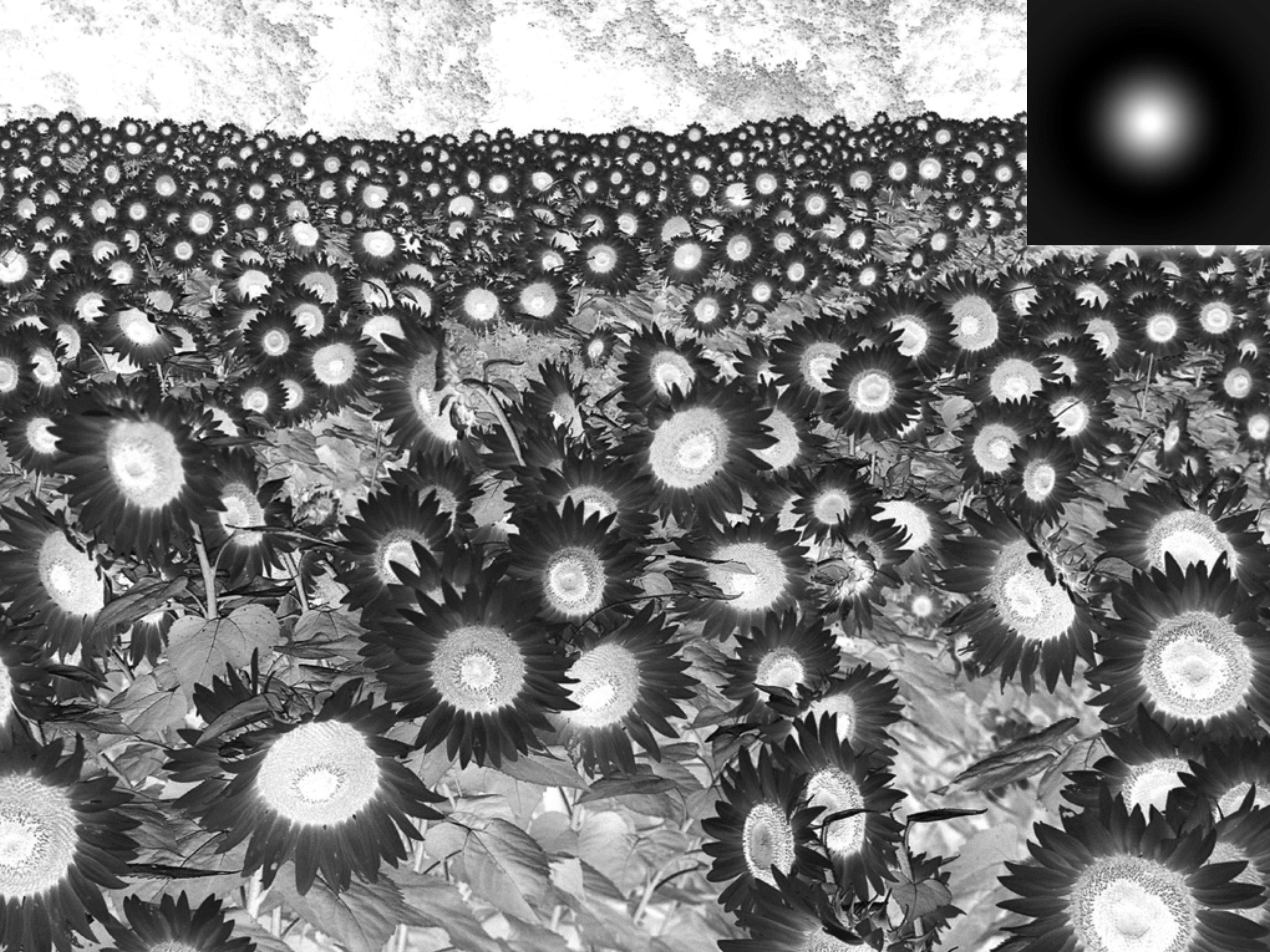
Convolve image with “blob” filter at multiple scales and find extrema

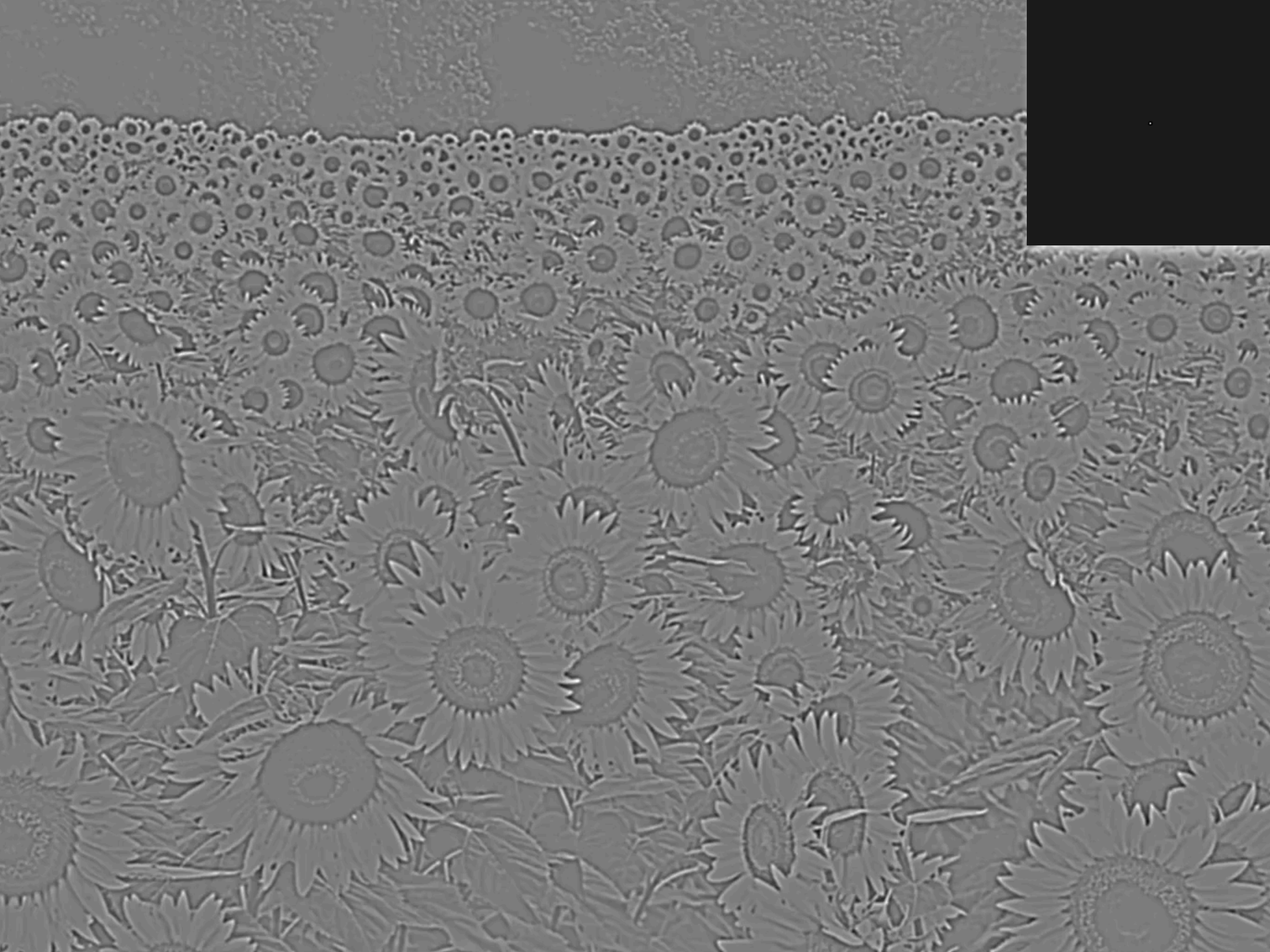


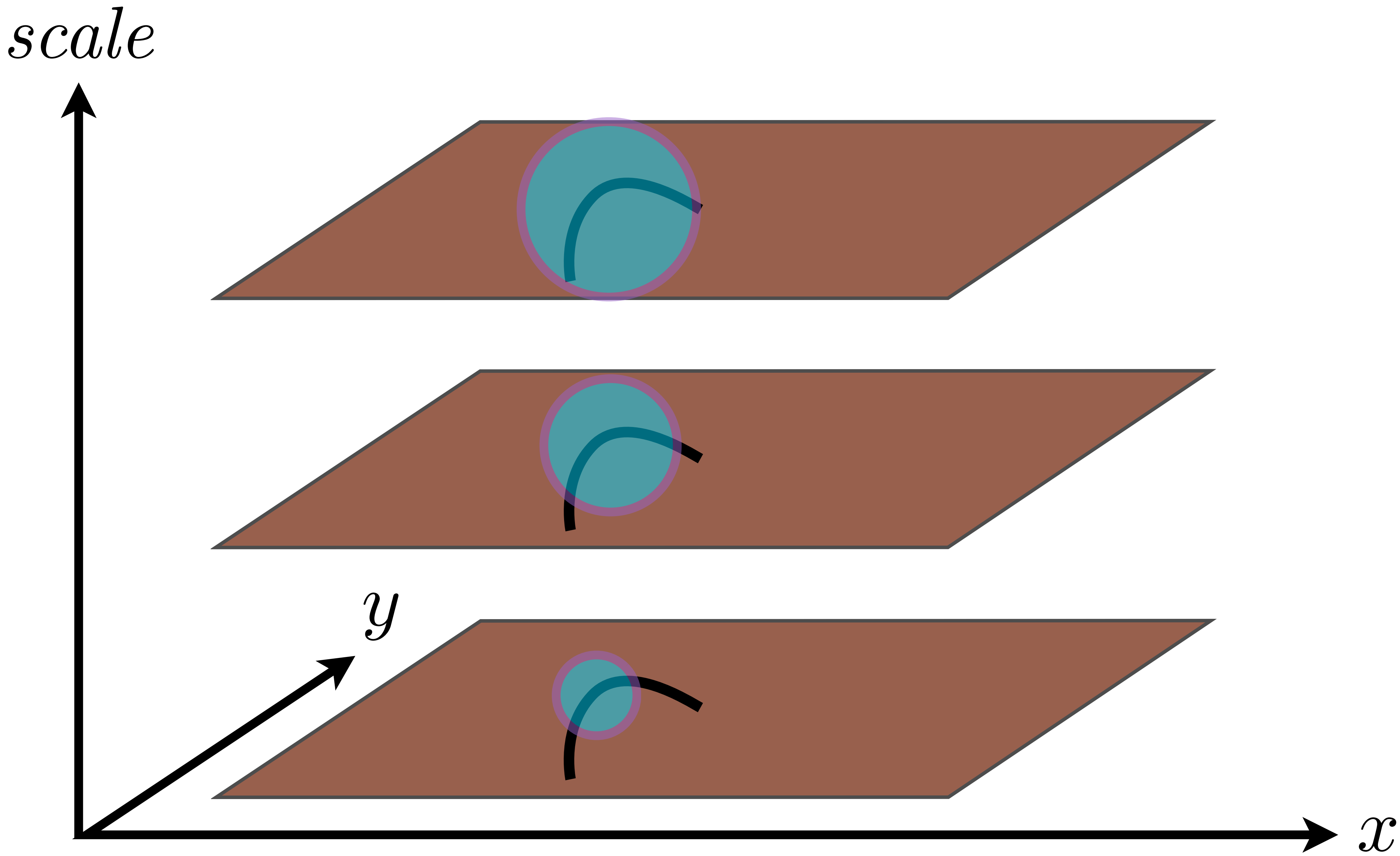
Laplacian extreme response is at $\sigma = \mathrm{radius}/\sqrt{2}$



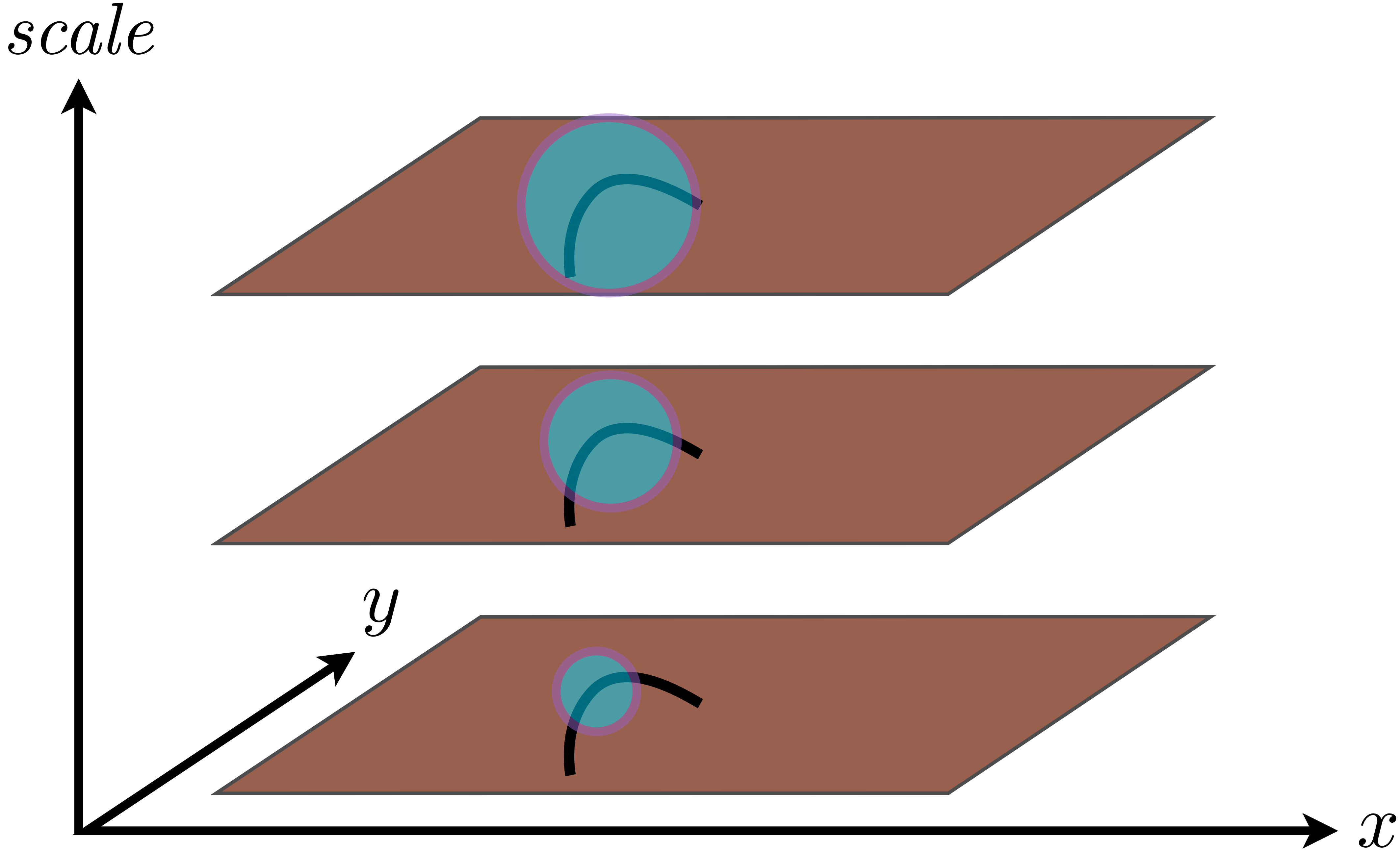
Laplacian extreme response is at $\sigma = \mathrm{radius}/\sqrt{2}$



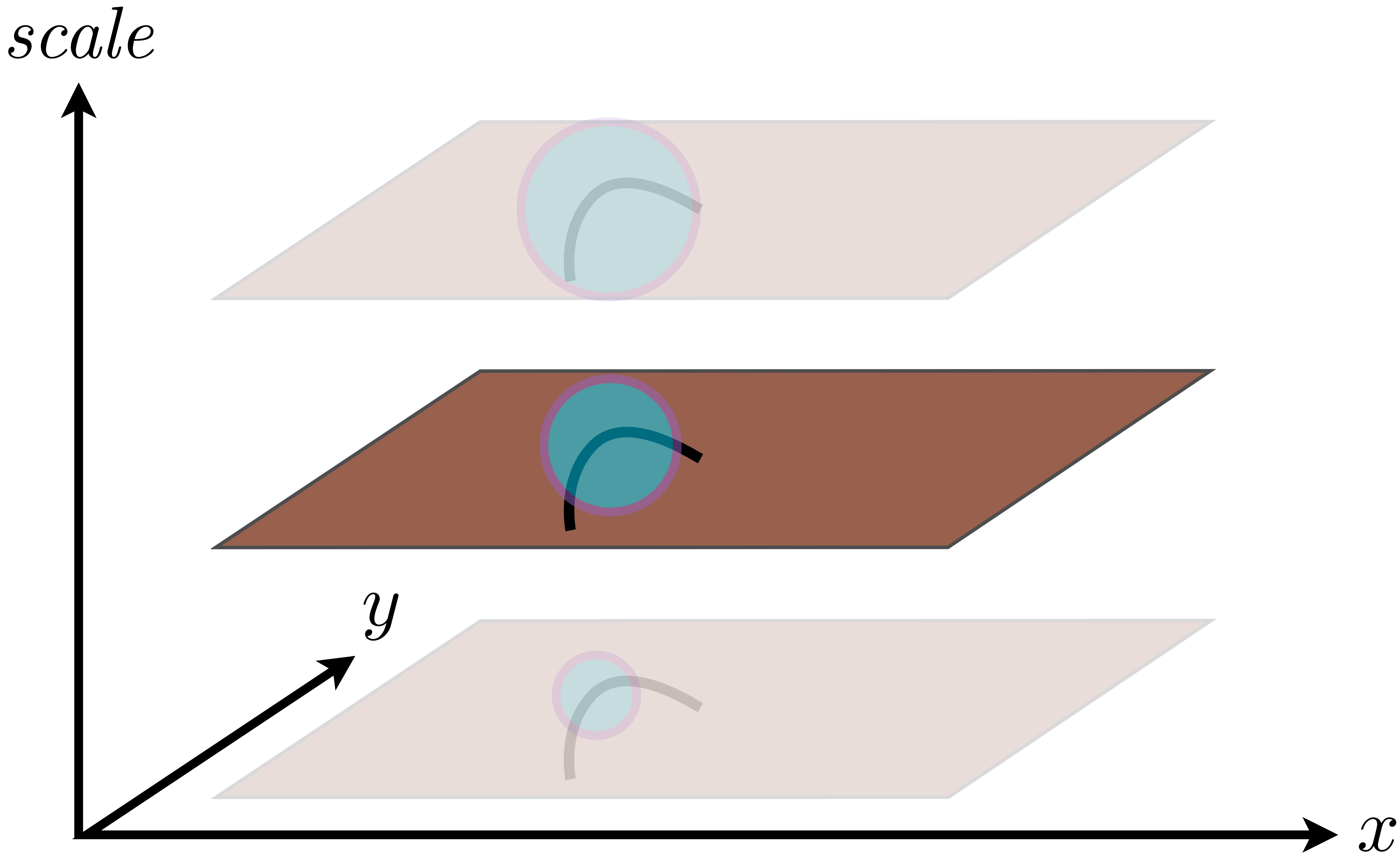




Convolve DoG across space at several scales

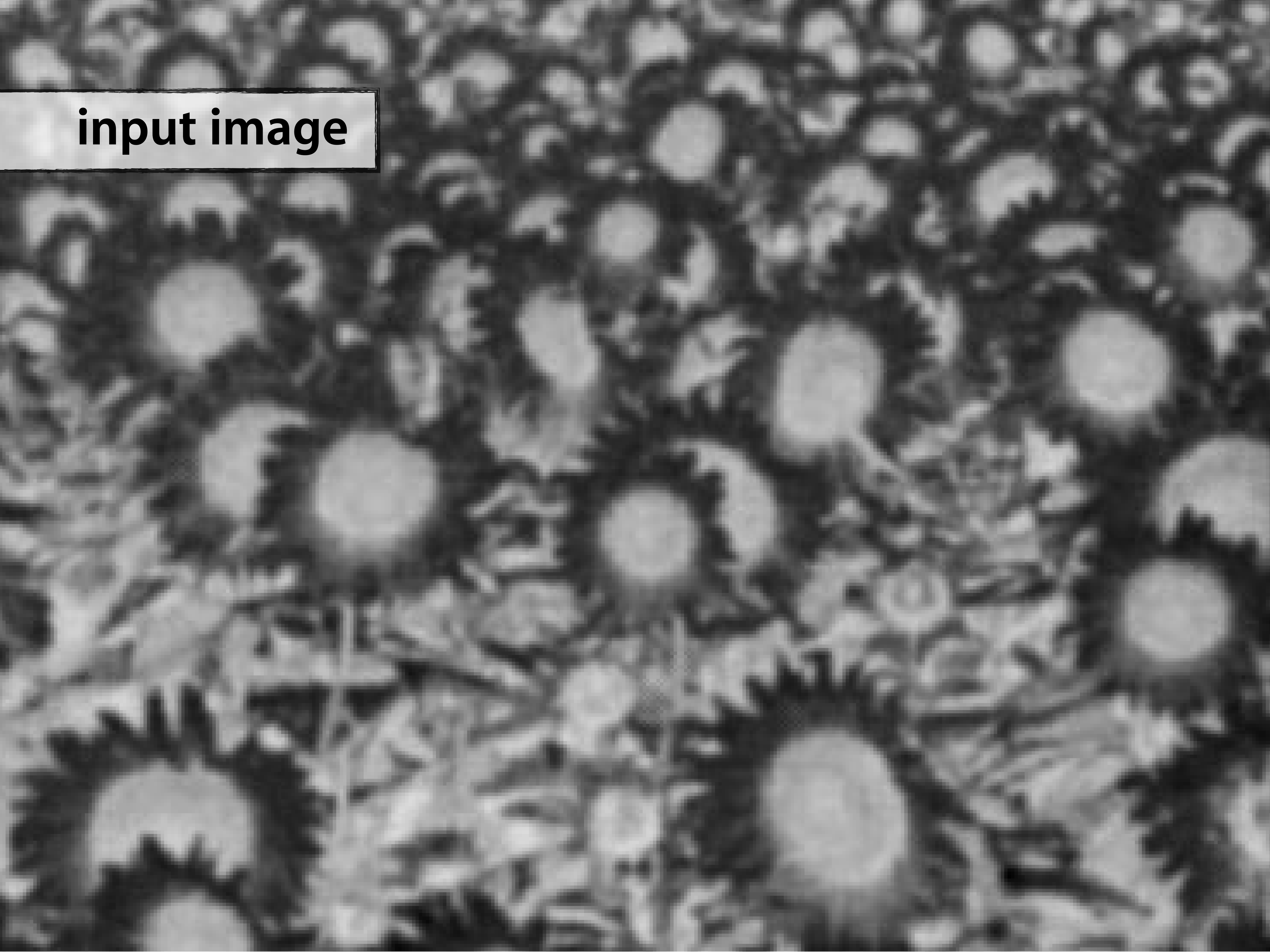


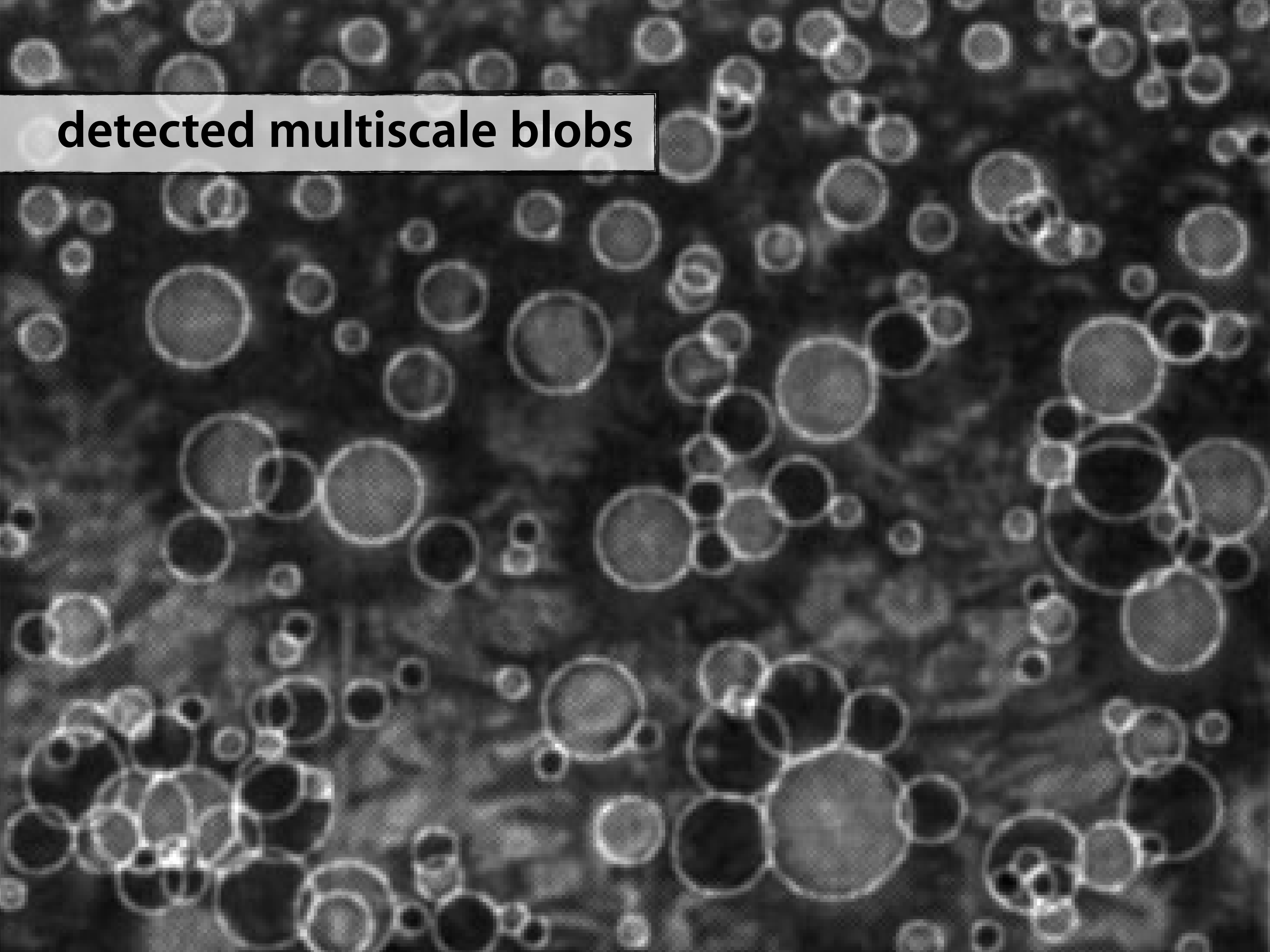
Find local maxima of DoG in space

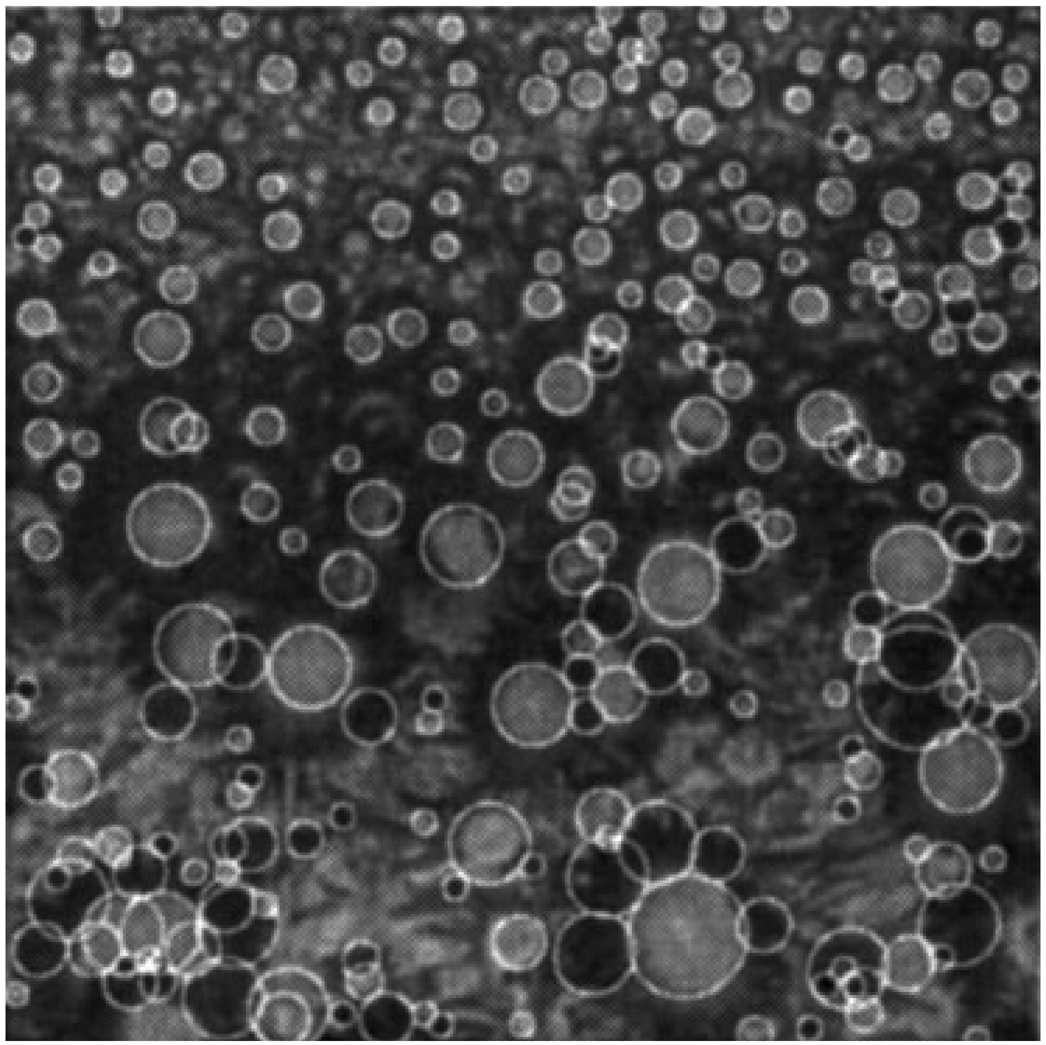


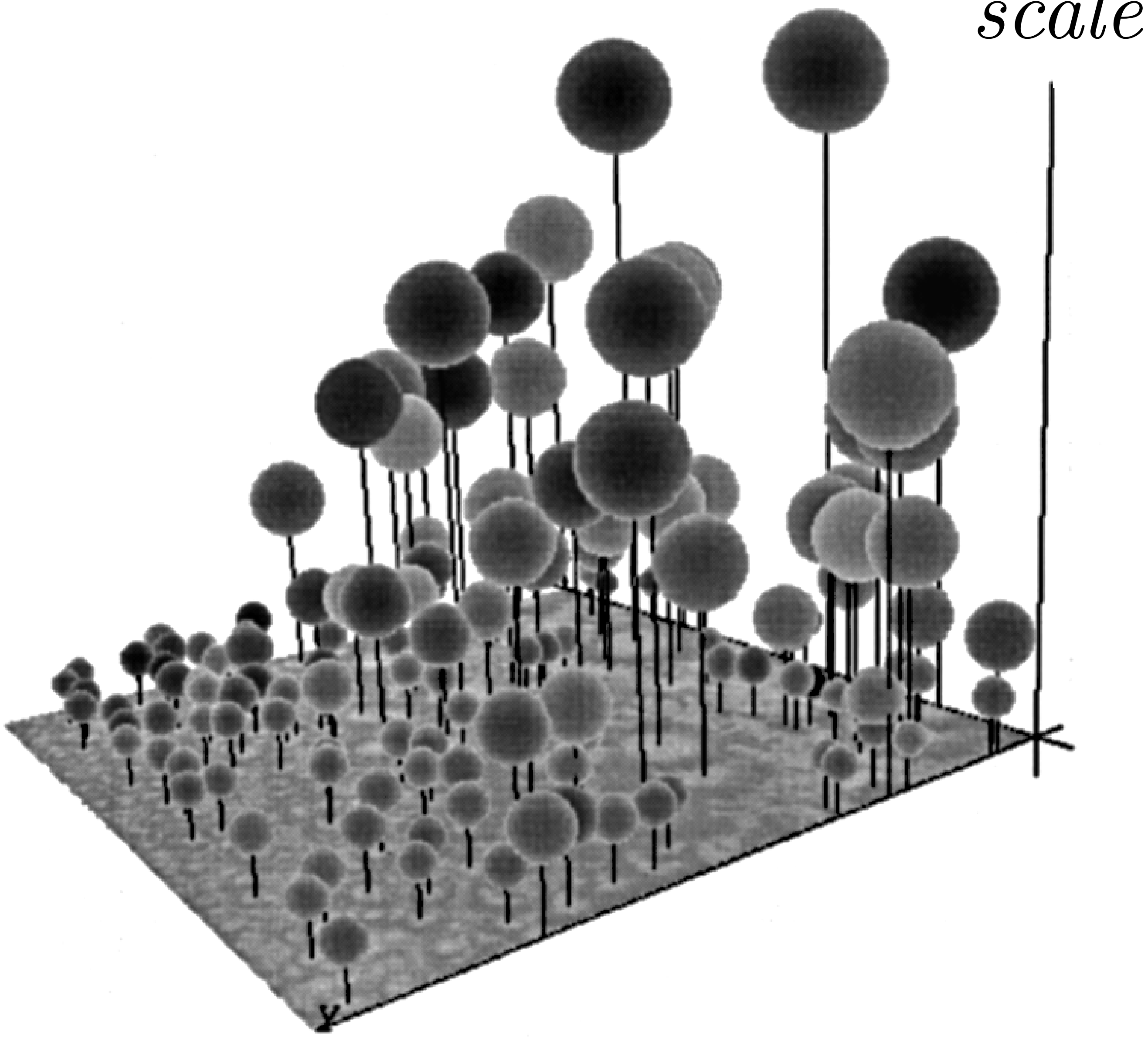
Find local maxima of DoG in space

and scale


input image

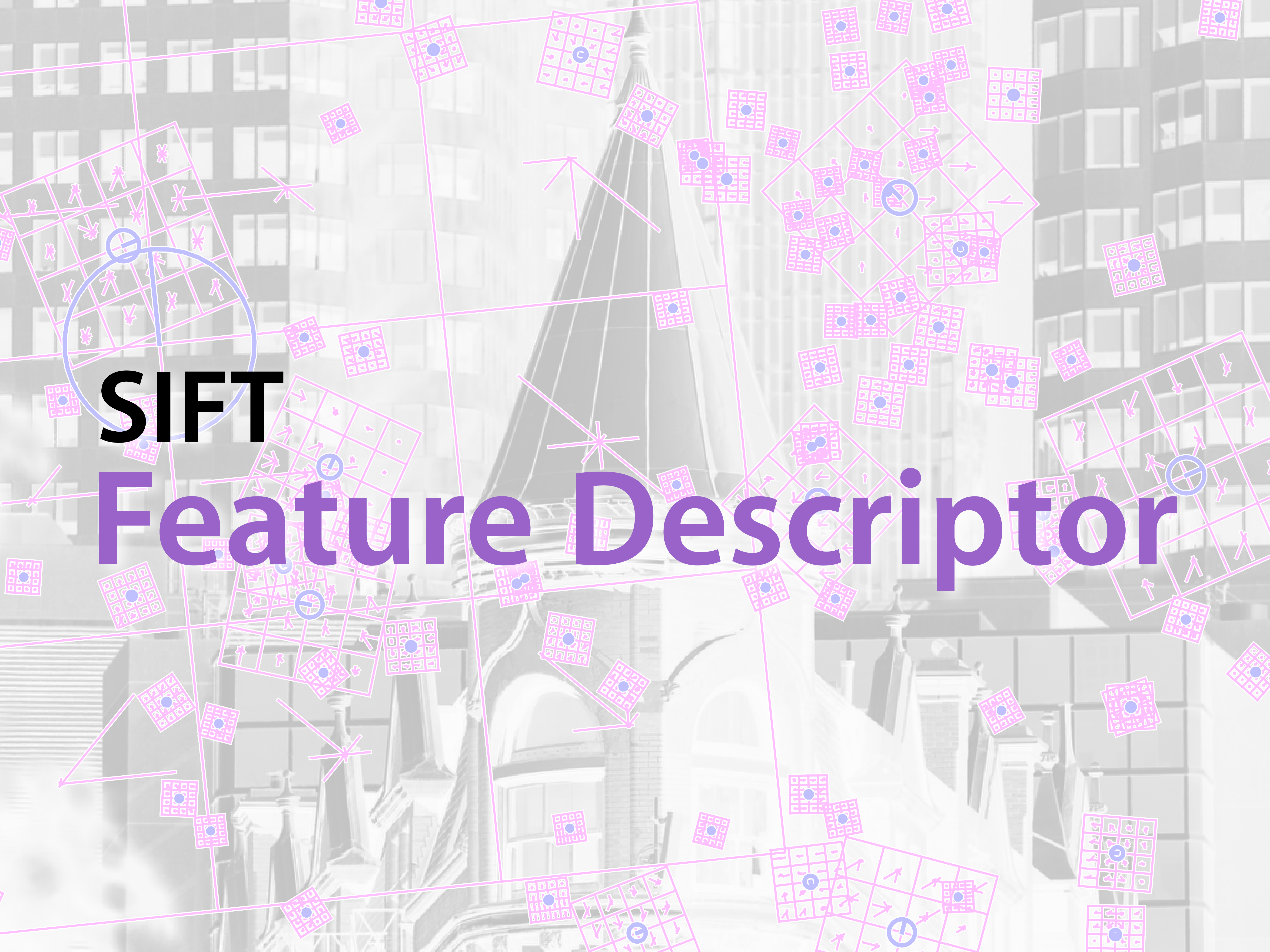

detected multiscale blobs

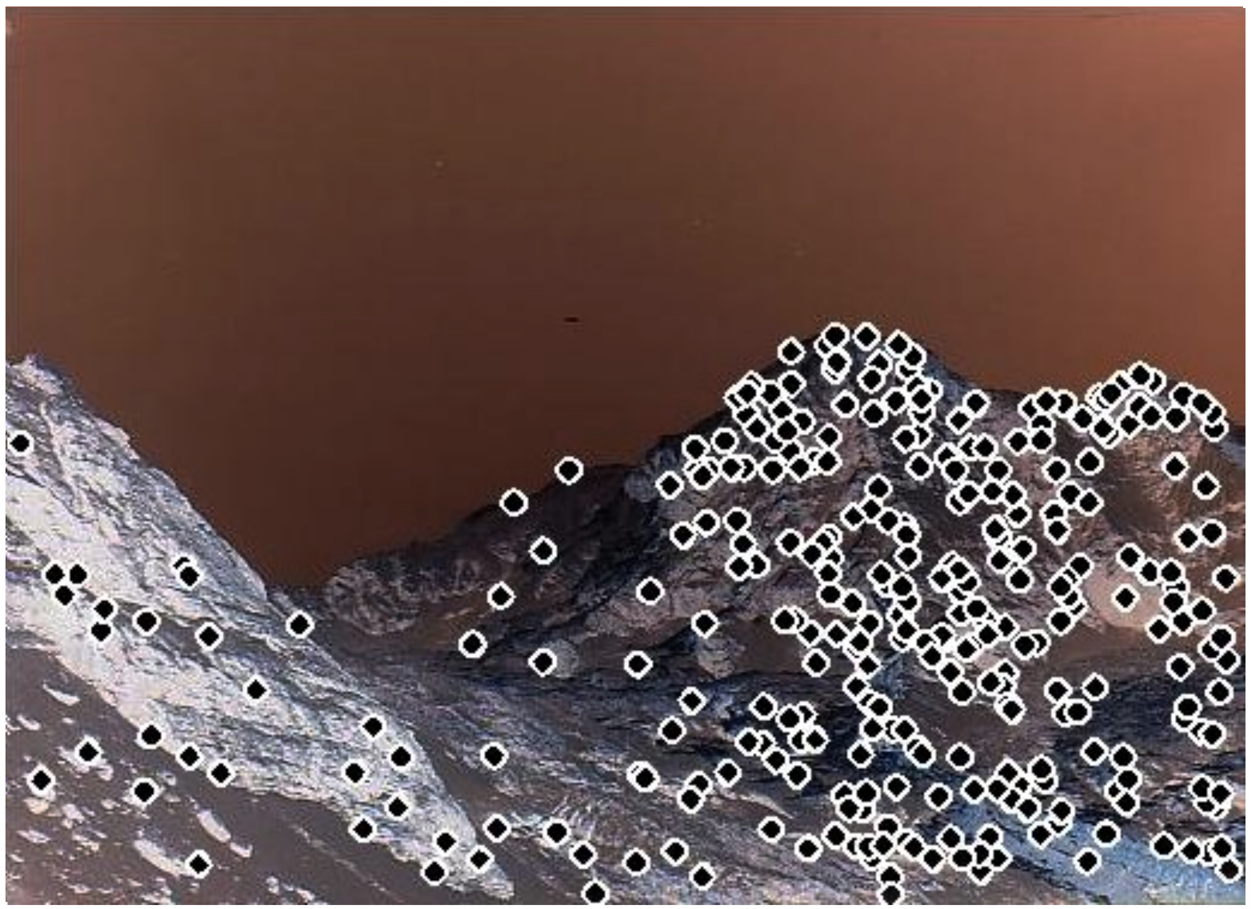


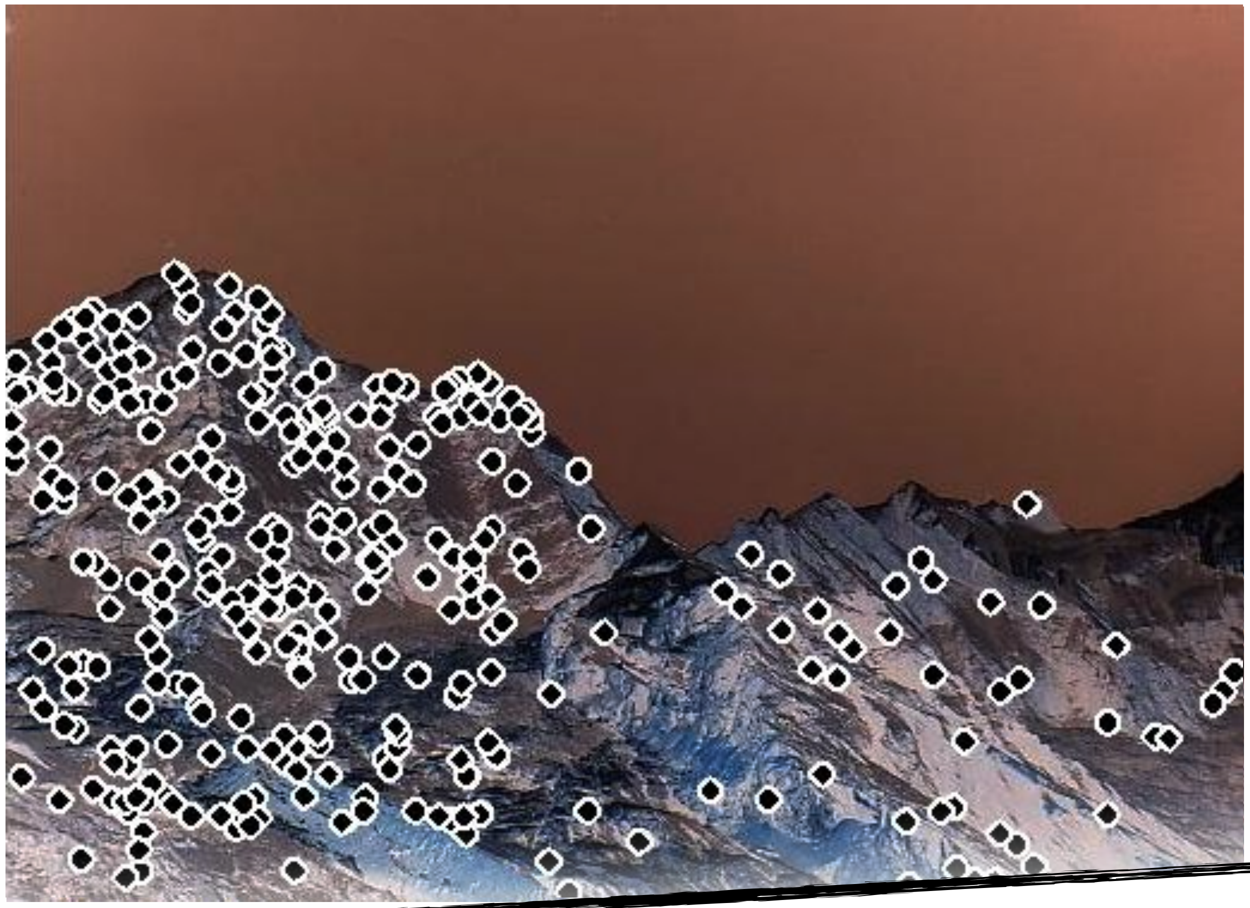


Lindeberg, Feature Detection with Automatic Scale Selection, IJCV, 1998

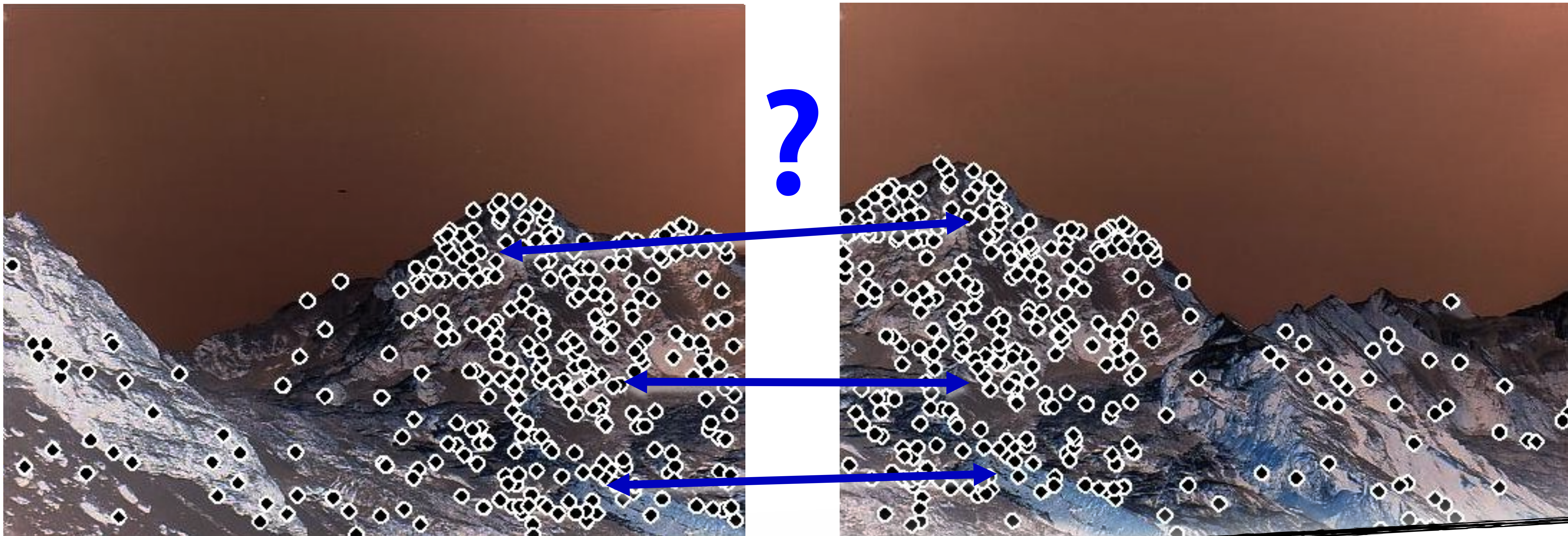
SIFT
Feature Descriptor







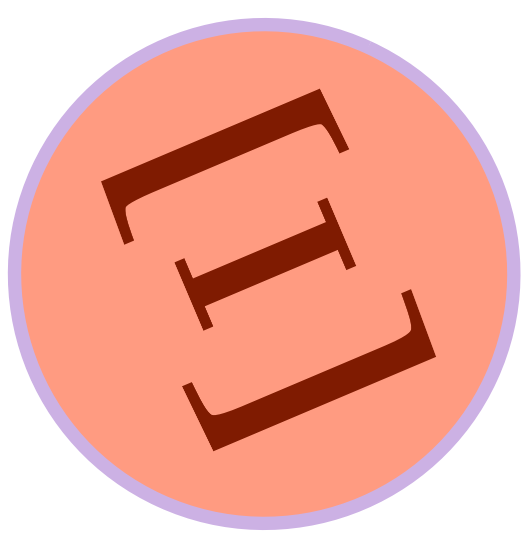
How do we match detected features across images?



How do we match detected features across images?

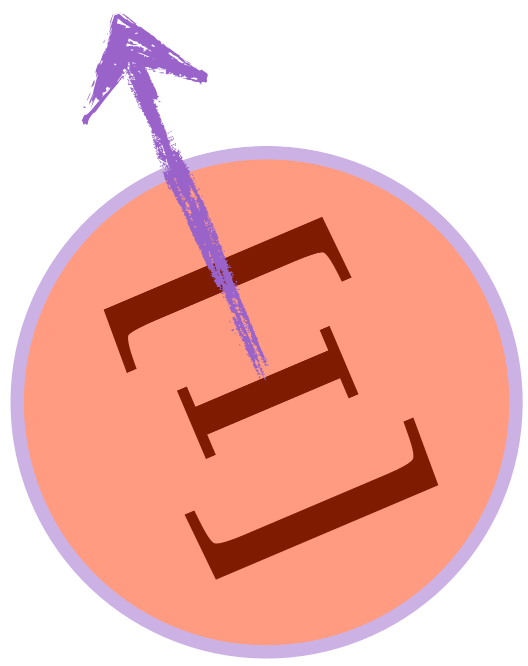
Rotation Invariant Description

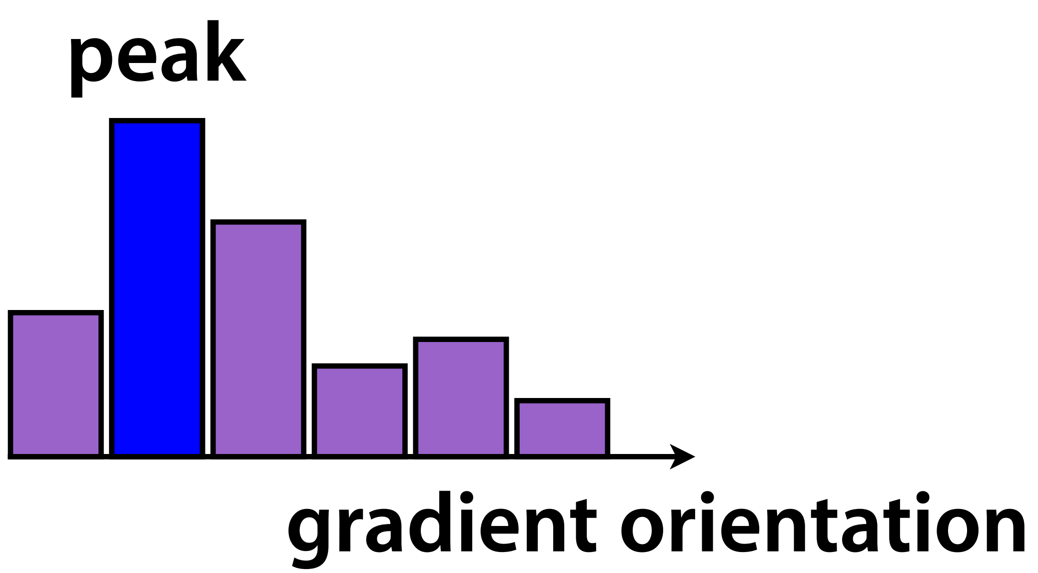
Step 1: Find dominant orientation of the patch



Rotation Invariant Description

Step 1: Find dominant orientation of the patch

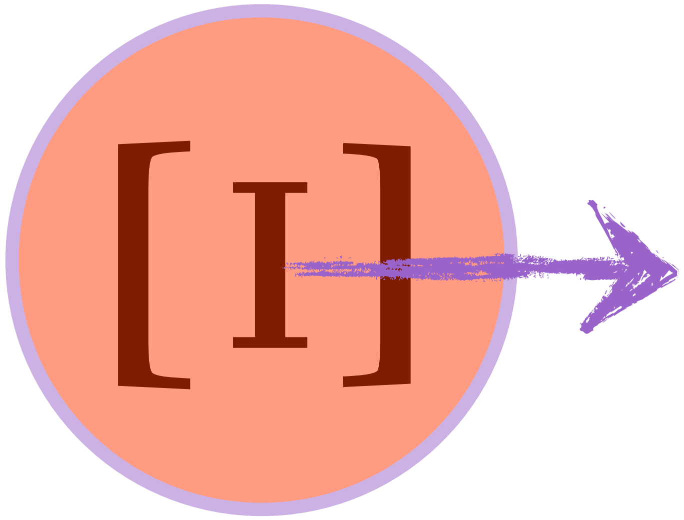


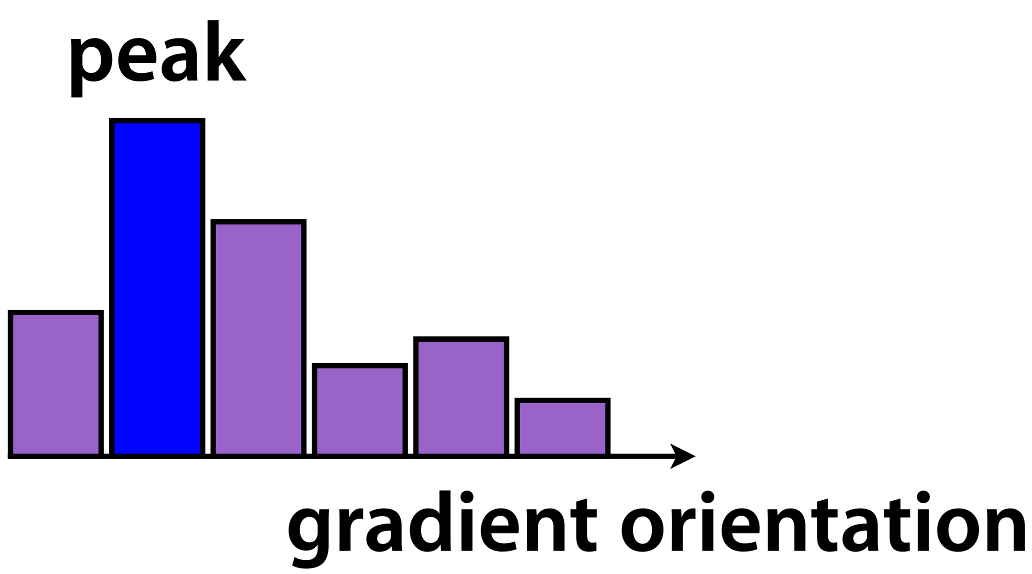


Rotation Invariant Description

Step 1: Find dominant orientation of the patch

Step 2: Rotate the patch to point along x-axis





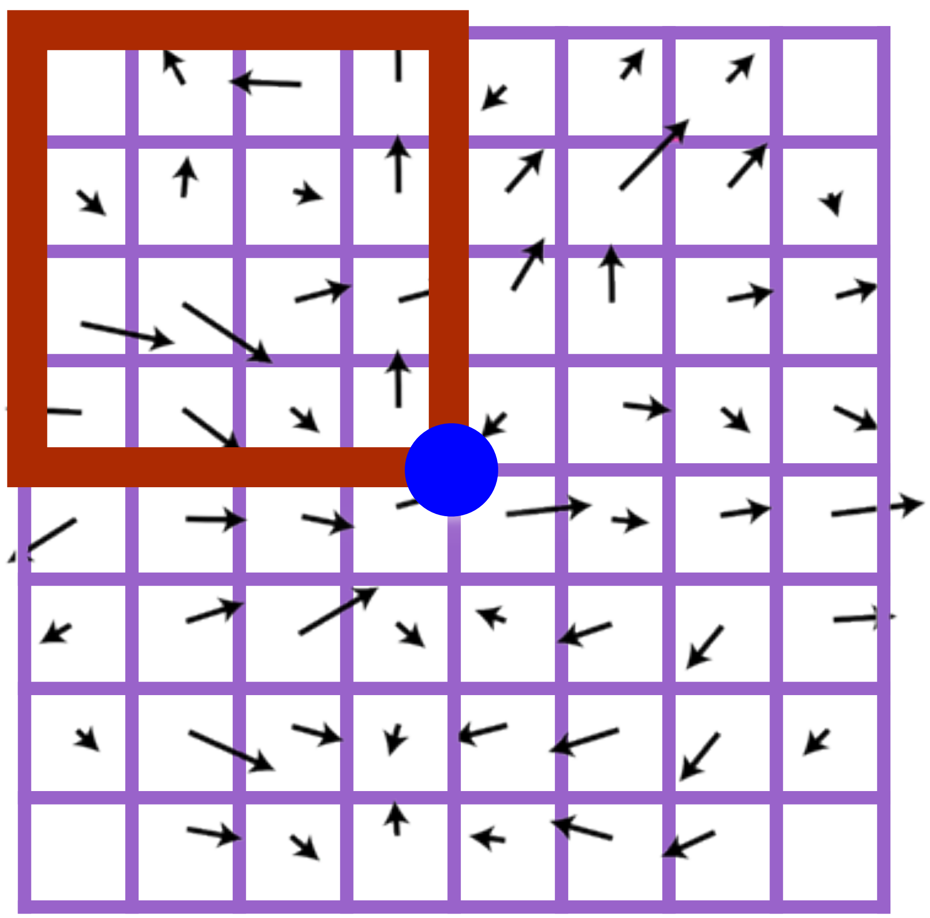


image gradients

SIFT Descriptor

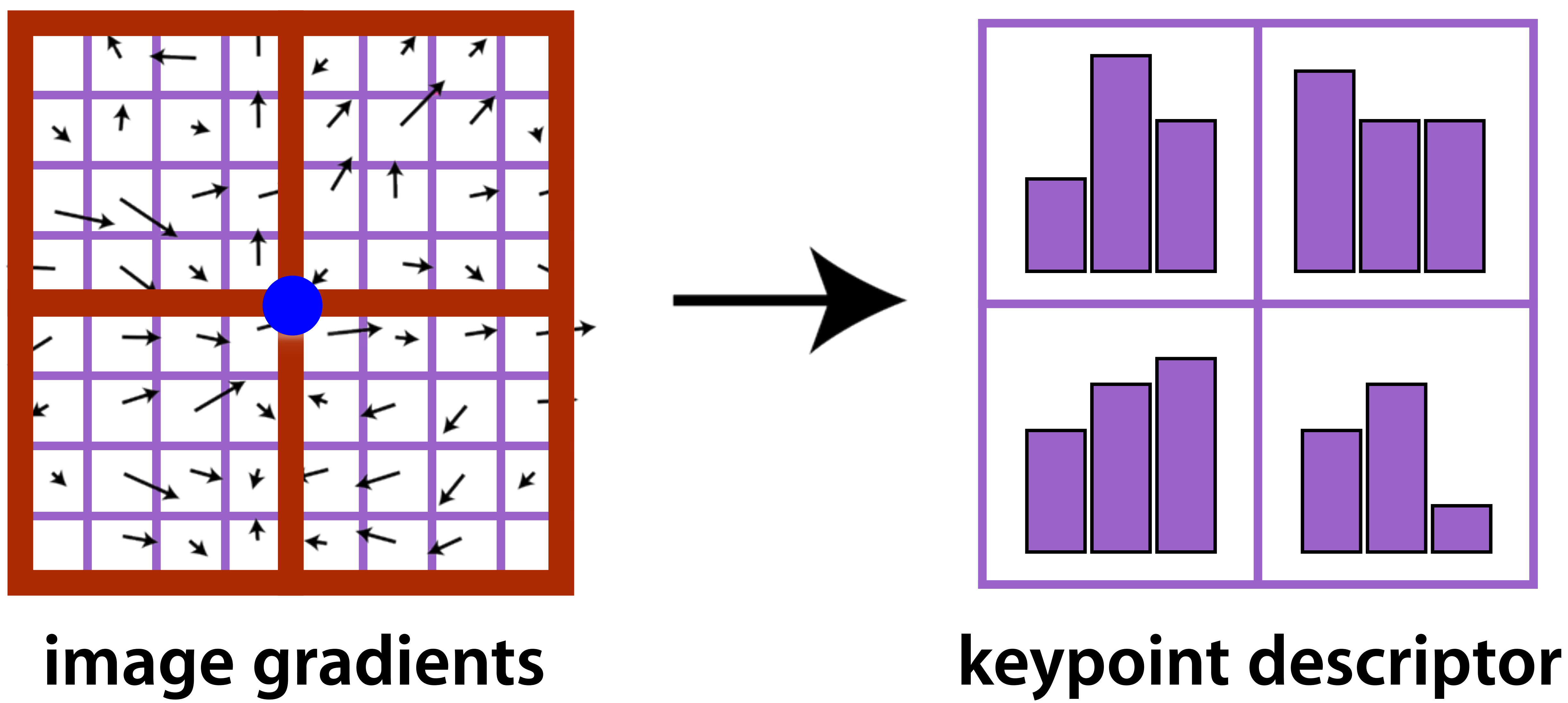
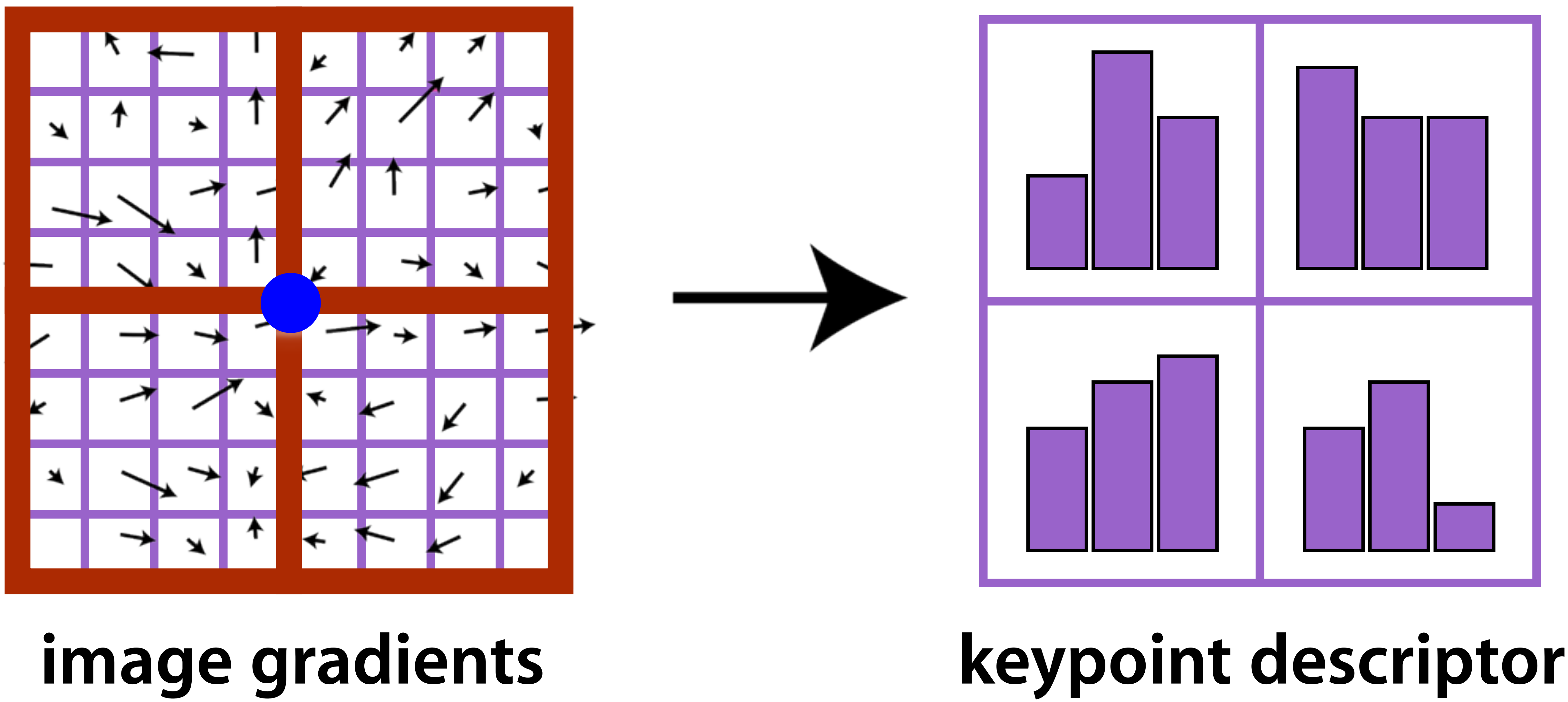


image gradients
keypoint descriptor



16 cells x 8 bins = 128 dimensional descriptor


4098
506 MAIN STREET STN

SIFT Properties

Robust to image rotation, scale and intensity change

SIFT Properties

Robust to image rotation, scale and intensity change

Robust to moderate out of plane rotation

SIFT Properties

Robust to image rotation, scale and intensity change

Robust to moderate out of plane rotation

Fast and efficient

1. **Keypoint detection**

   Search across image locations and scales for feature response extrema

2. **Keypoint orientation assignment**

   Determine best orientation(s) for each keypoint

3. **Keypoint description**

   Describe keypoint region at selected scale and rotation with image gradients