Intro to

Computer Vision

Edge detection

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. PAMI-8, NO. 6, NOVEMBER 1986 679

A Computational Approach to Edge Detection

JOHN CANNY, MEMBER, IEEE

***Abstract*—This paper describes a computational approach to edge detection. The success of the approach depends on the definition of a *comprehensive* set of goals for the computation of edge points. These goals must be precise enough to delimit the desired behavior of the detector while making minimal assumptions about the form of the solution. We define detection and localization criteria for a class of edges, and present mathematical forms for these criteria as functionals on the**

detector as input to a program which could isolate simple geometric solids. More recently the model-based vision system ACRONYM [3] used an edge detector as the front end to a sophisticated recognition program. Shape from motion [29], [13] can be used to infer the structure of three-dimensional objects from the motion of edge con-

36 Thousand

citations

4

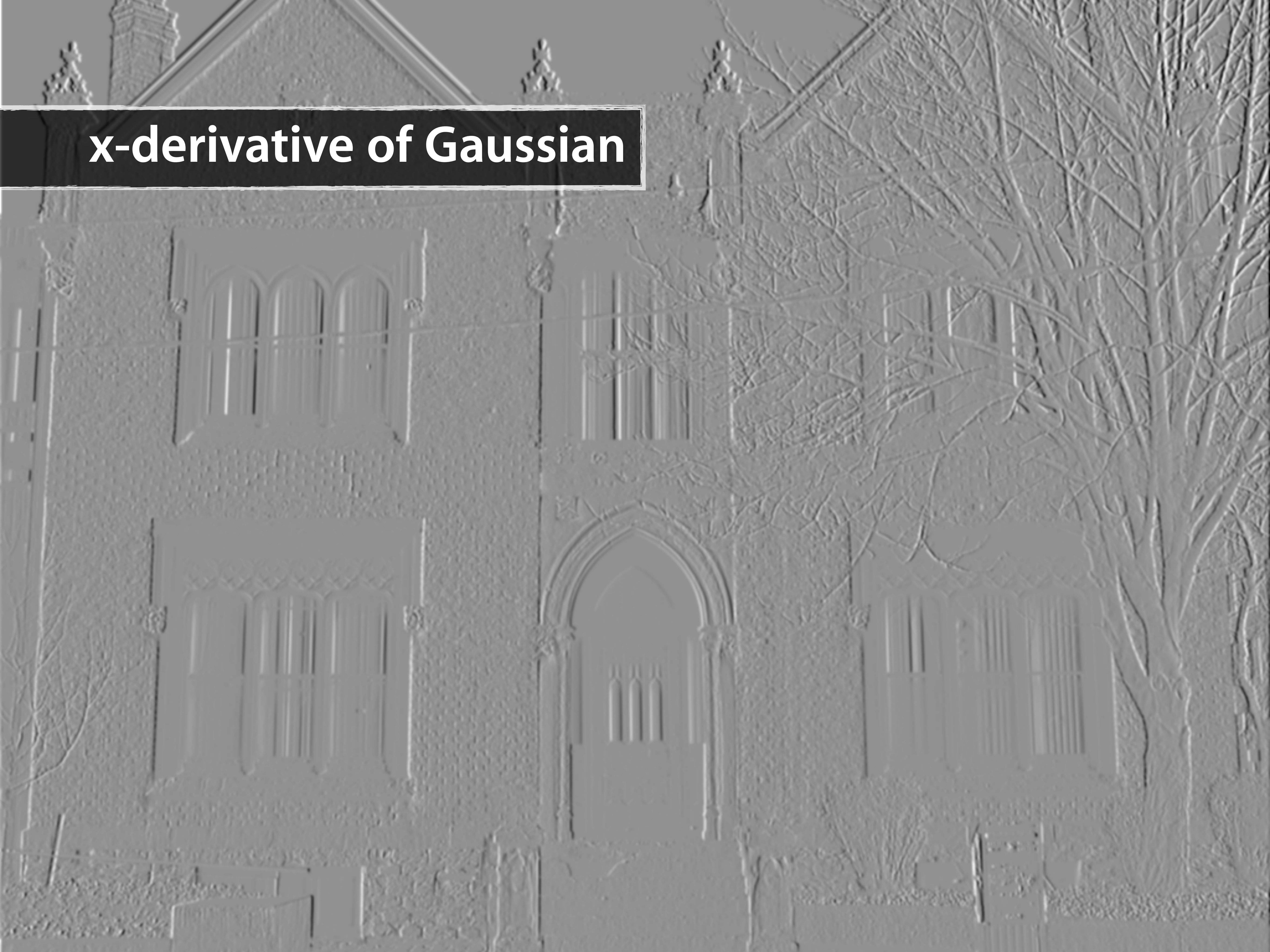
major steps

Step 1

Filter image with x and y derivative of Gaussian


input image

x-derivative of Gaussian



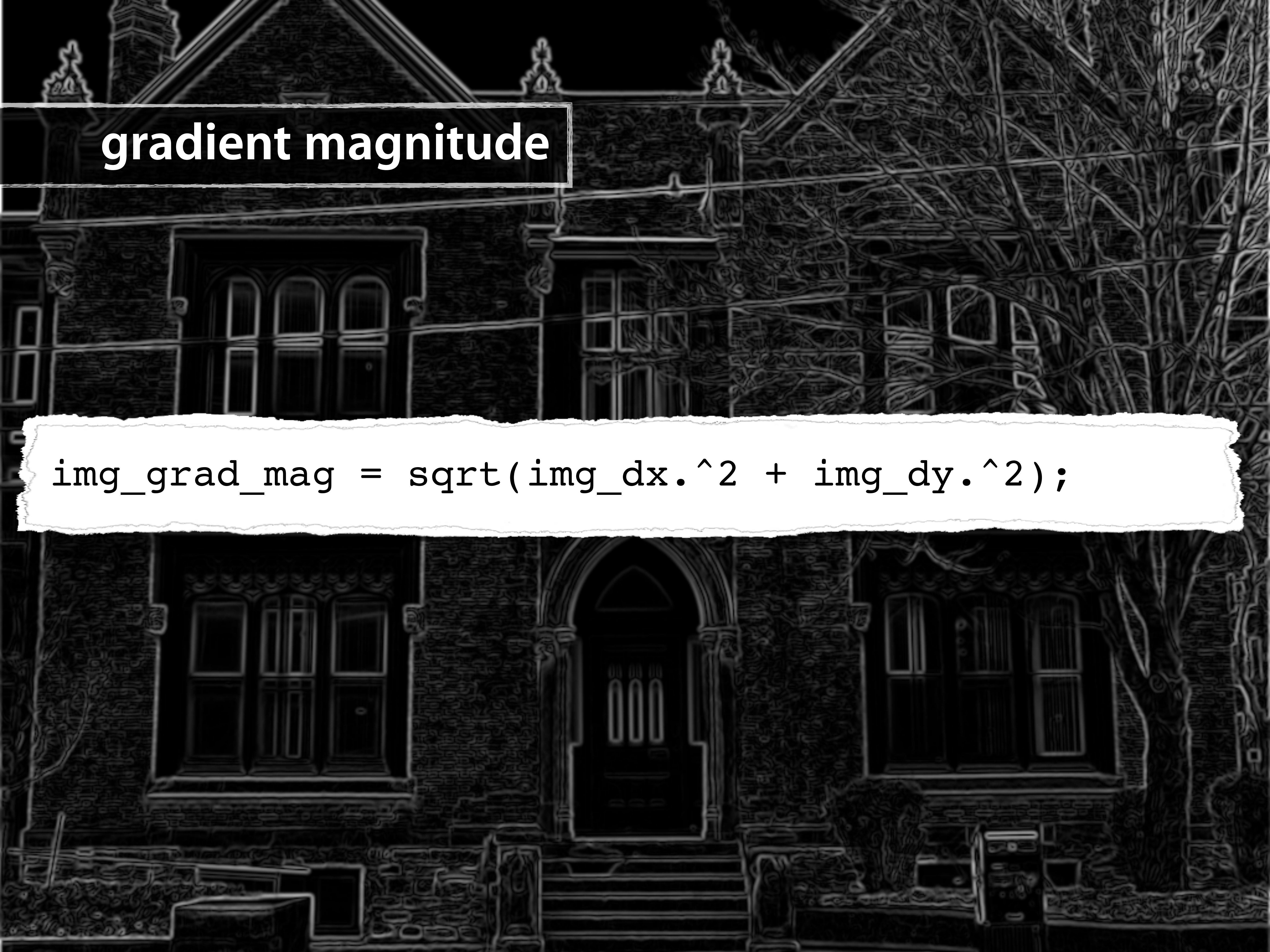
y-derivative of Gaussian

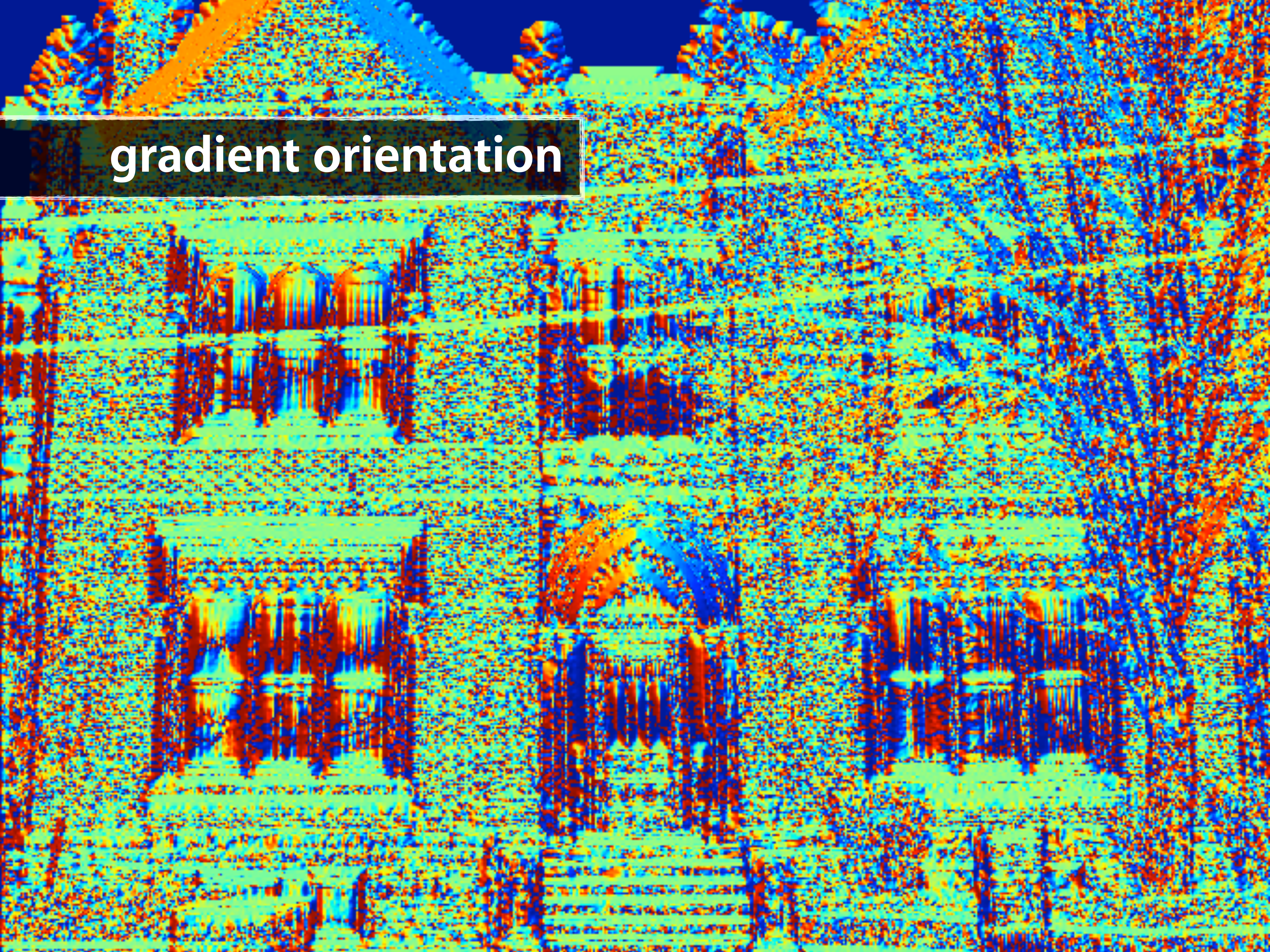
Step 2

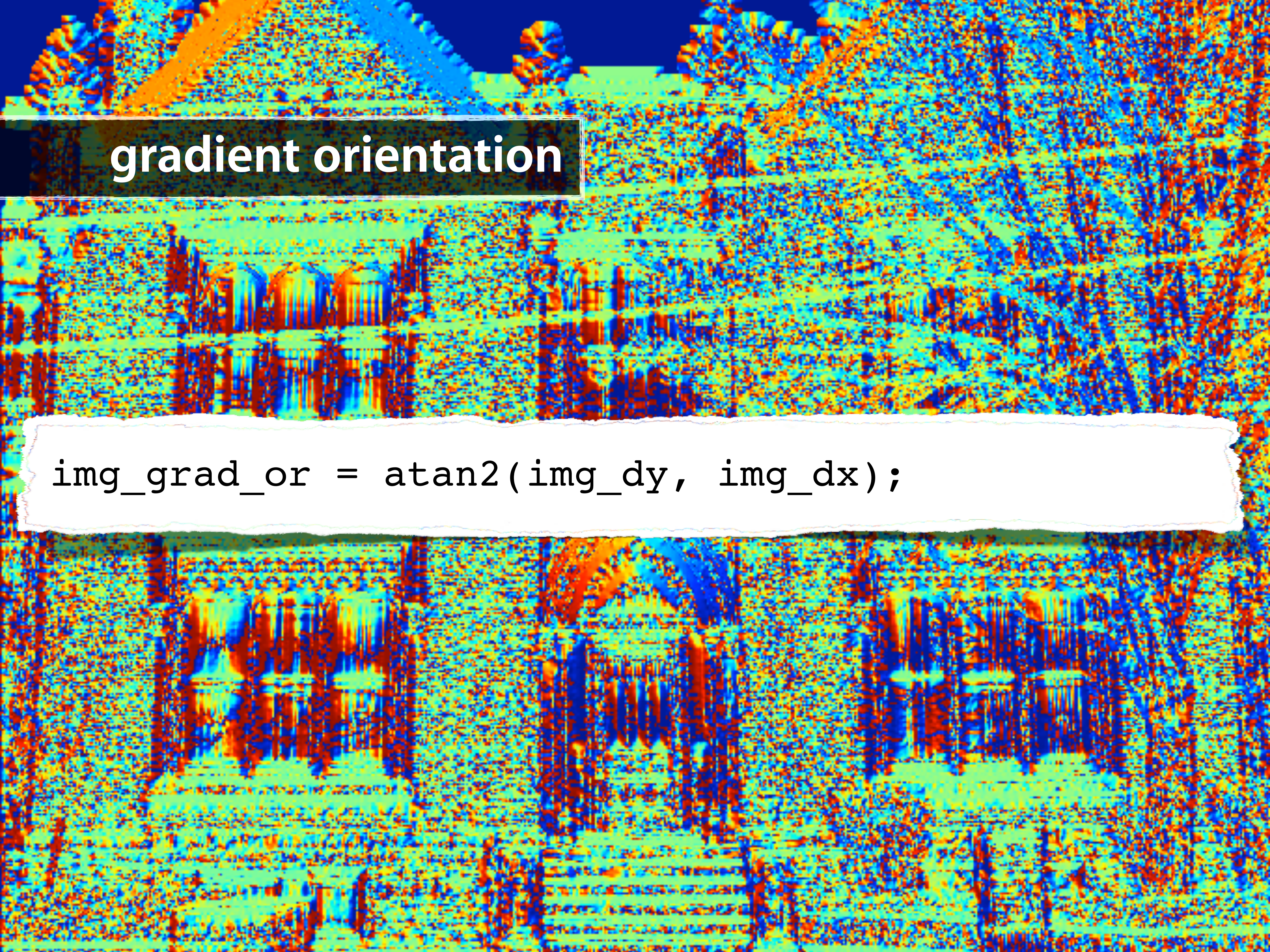
Find magnitude and direction of gradient


input image

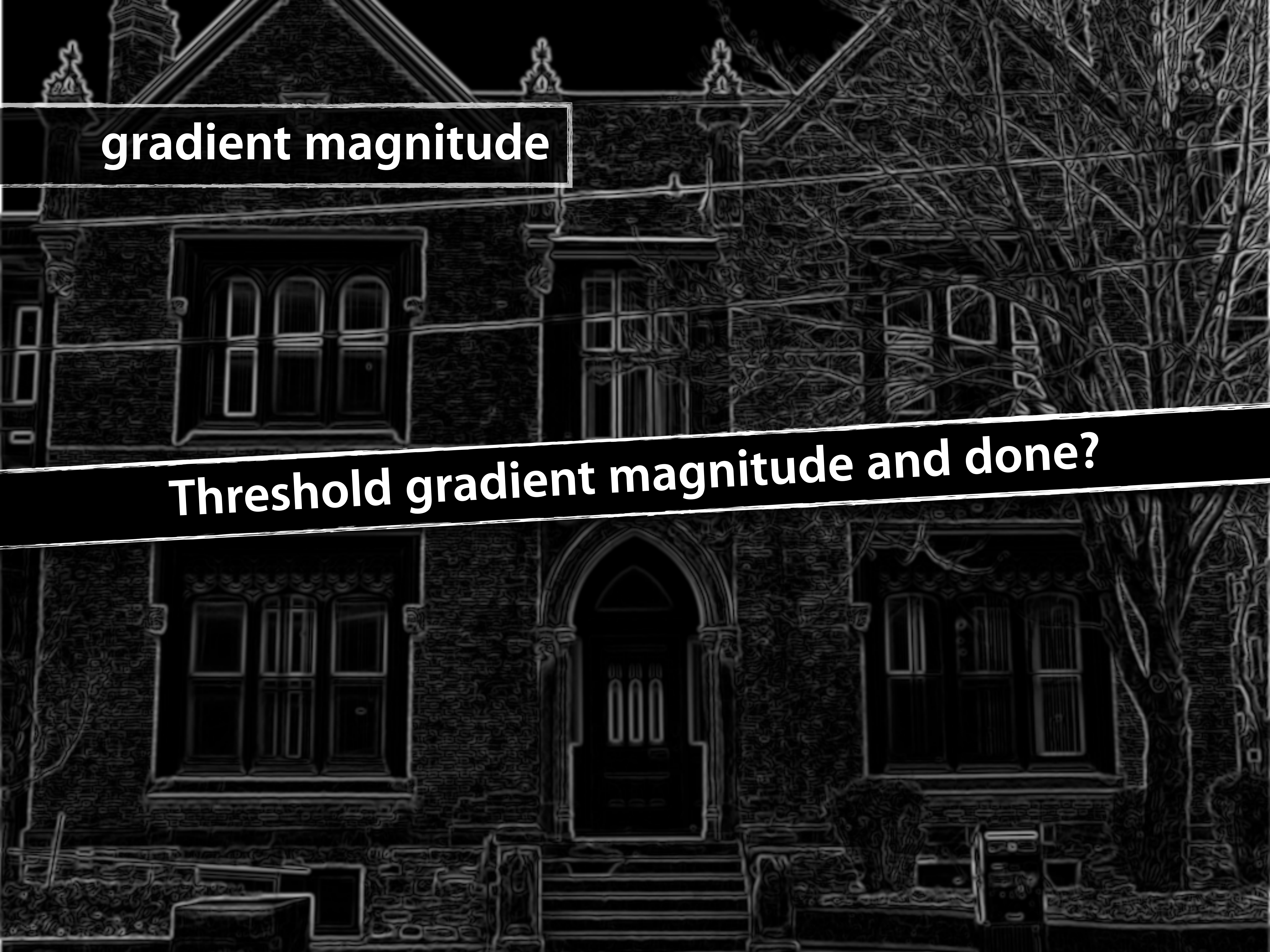

gradient magnitude

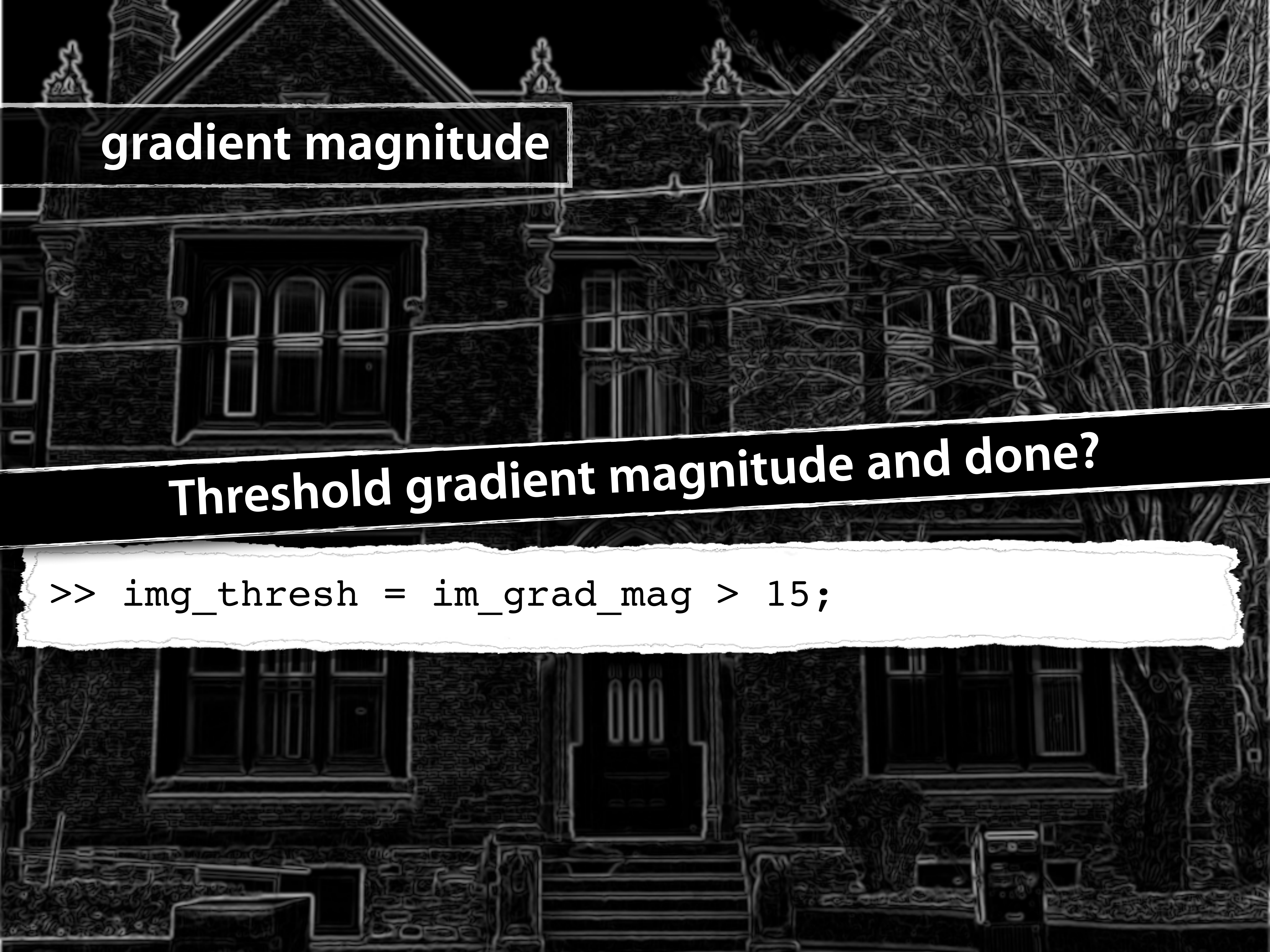

gradient magnitude
img_grad_mag = sqrt(img_dx.^2 + img_dy.^2);


gradient orientation


gradient orientation
img_grad_or = atan2(img_dy, img_dx);


gradient magnitude


gradient magnitude
Threshold gradient magnitude and done?



gradient magnitude

Threshold gradient magnitude and done?

```
>> img_thresh = im_grad_mag > 15;
```

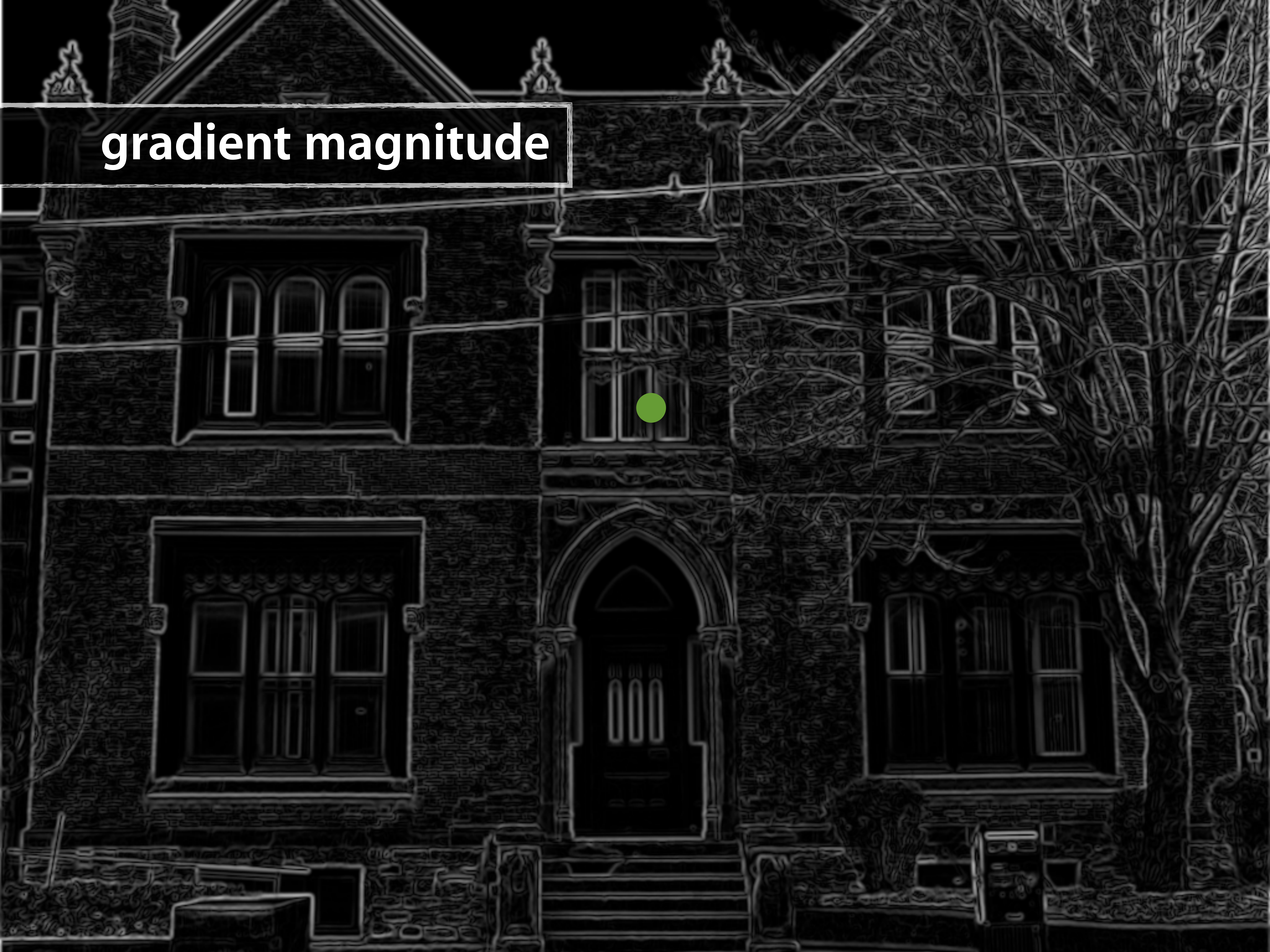

gradient magnitude threshold

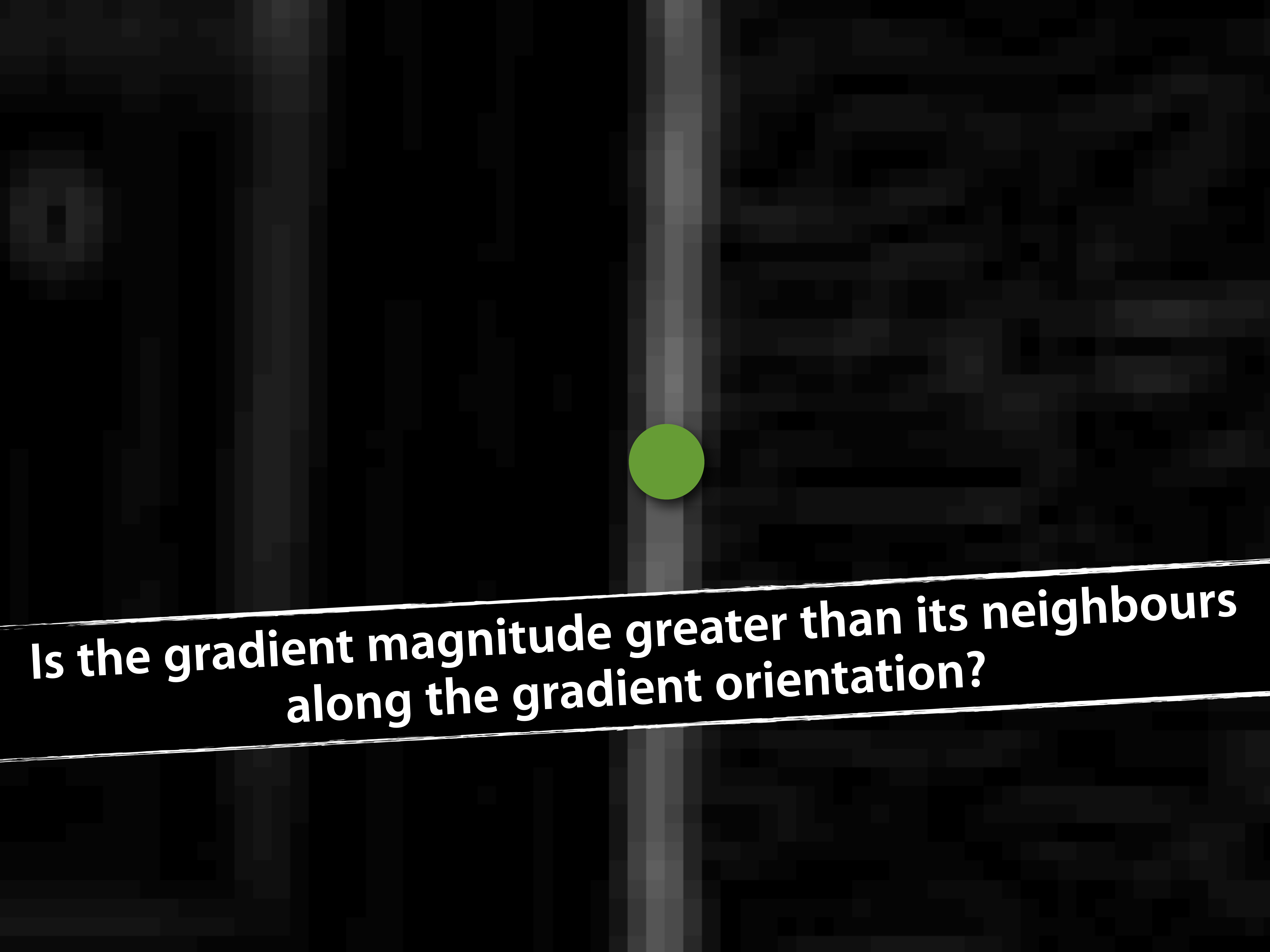

gradient magnitude threshold
PROBLEM: End up with thick edges

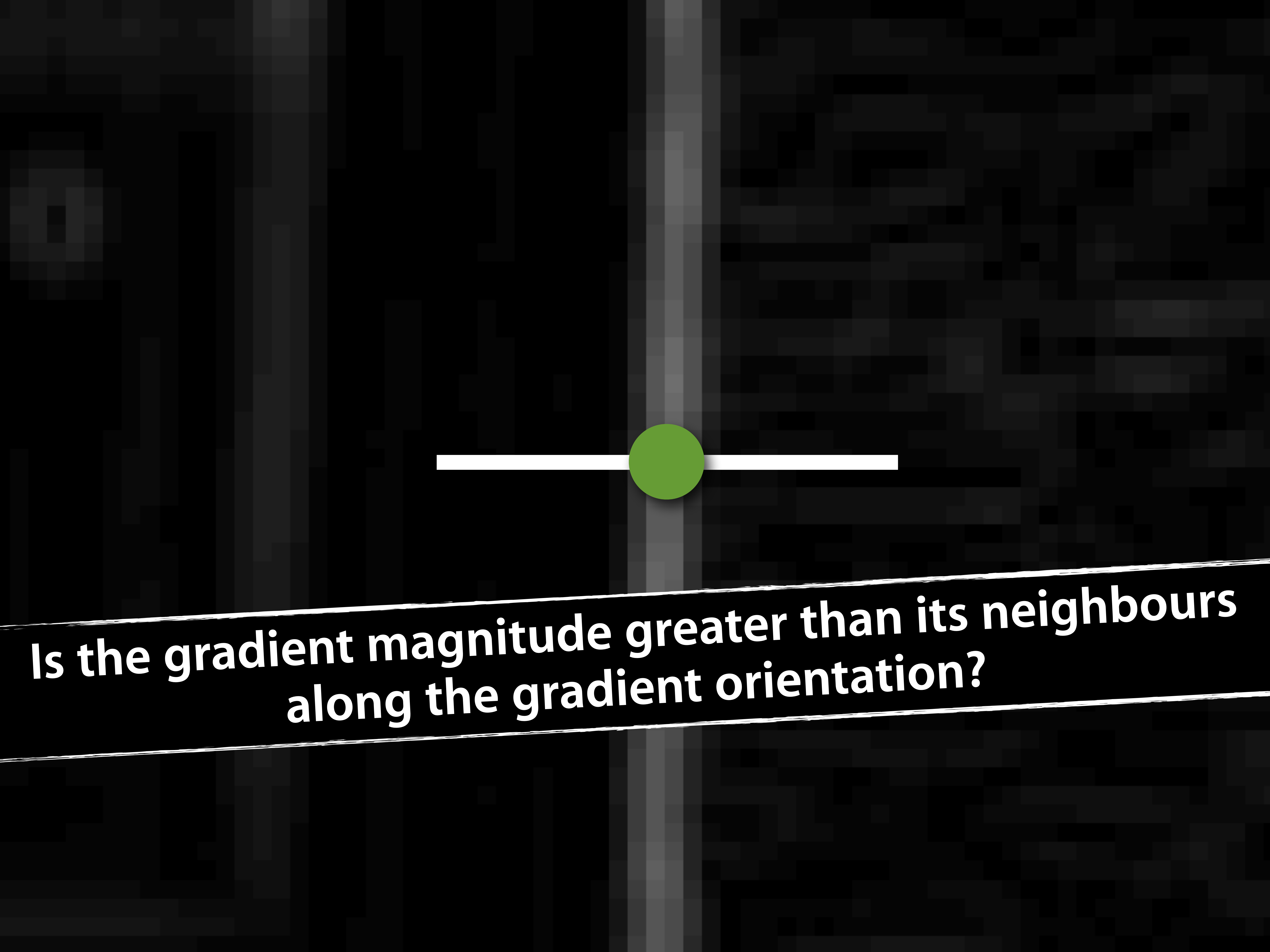
Step 3

Perform non-maximum suppression

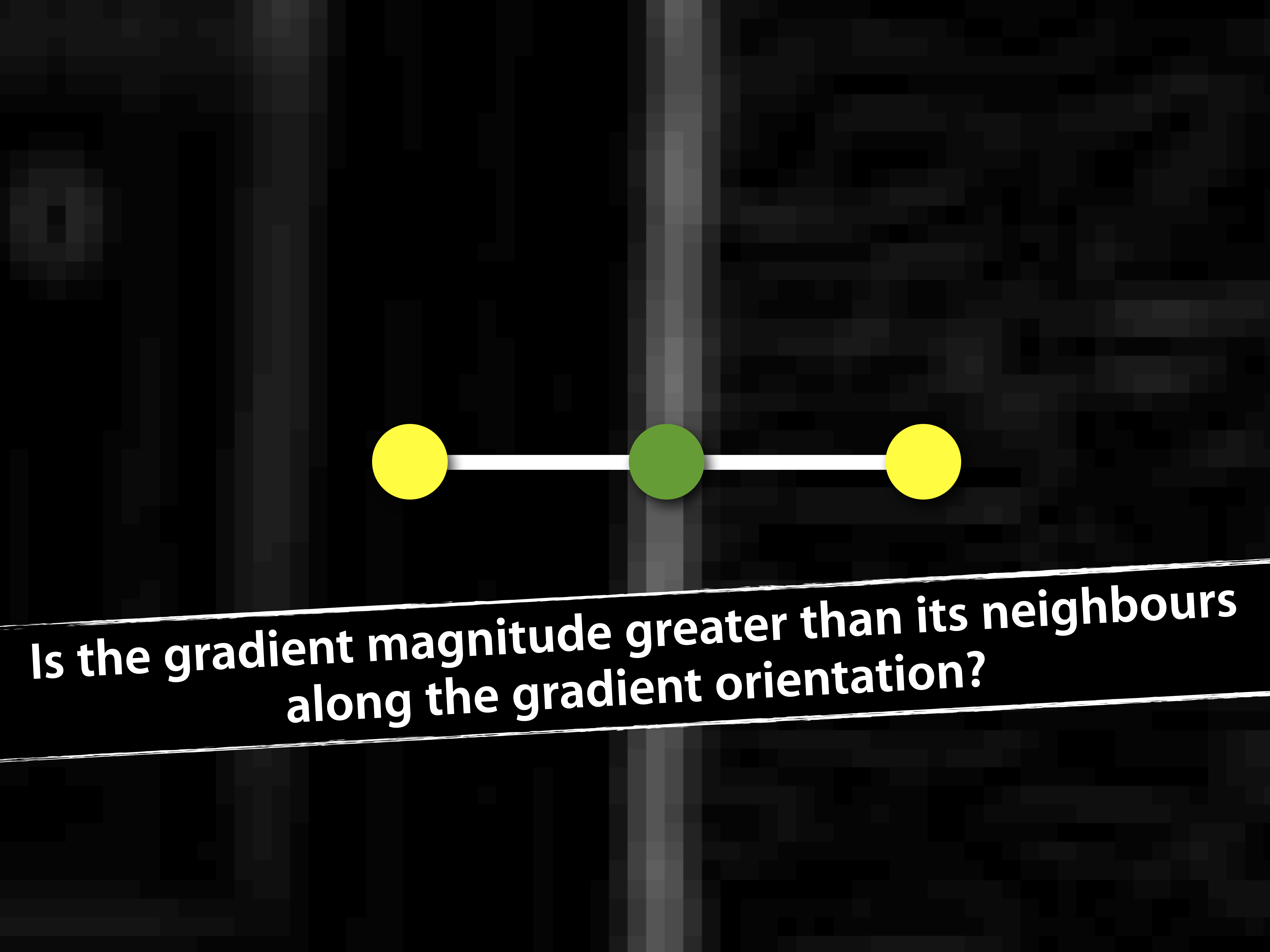

gradient magnitude

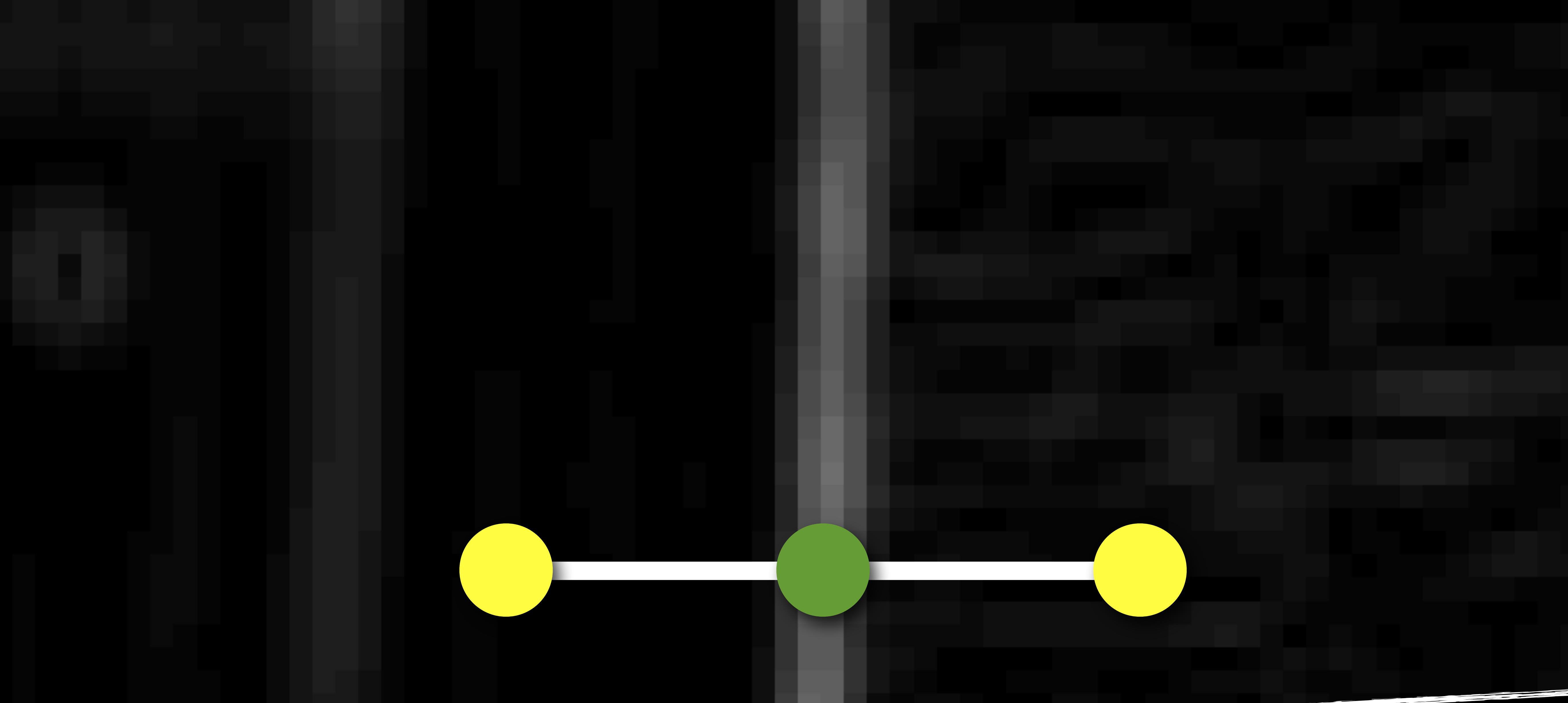

gradient magnitude


Is the gradient magnitude greater than its neighbours
along the gradient orientation?



Is the gradient magnitude greater than its neighbours along the gradient orientation?


Is the gradient magnitude greater than its neighbours
along the gradient orientation?



Is the gradient magnitude greater than its neighbours along the gradient orientation?

If yes, keep, otherwise suppress.


gradient magnitude

non-max suppression

Step 4

Threshold and link

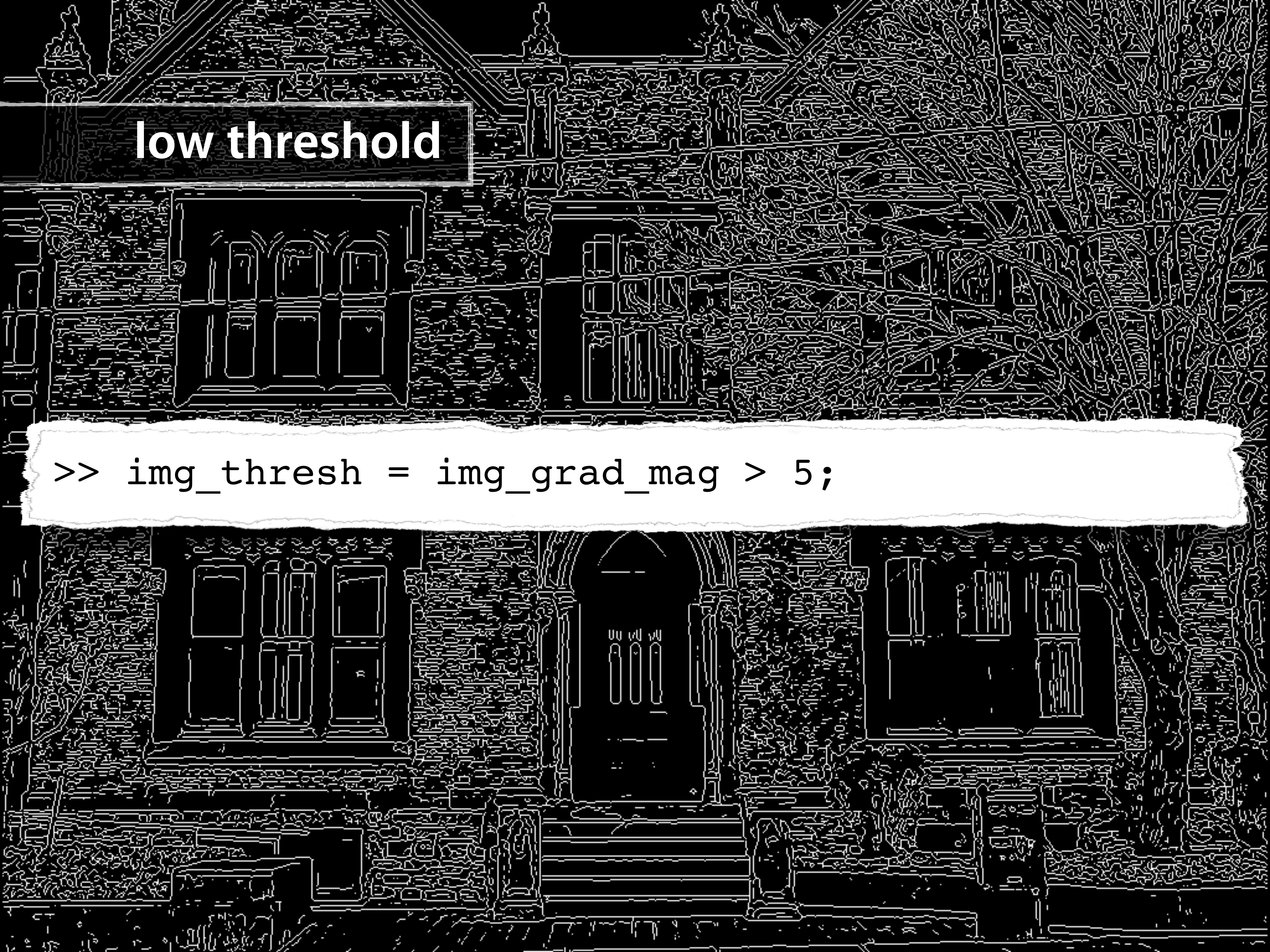
Step 4

Threshold and link

Hysteresis thresholding


non-max suppression

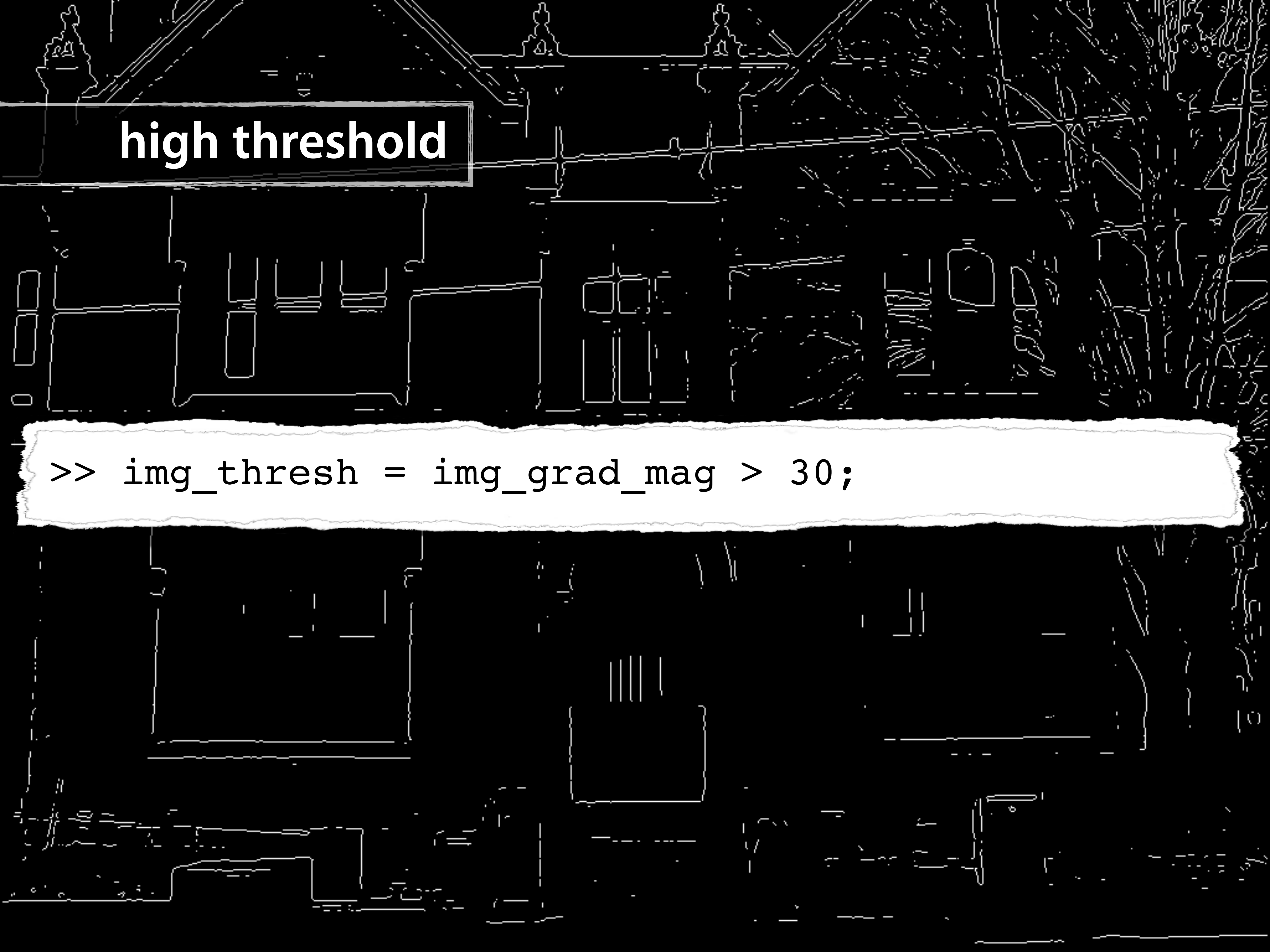

low threshold



low threshold

```
>> img_thresh = img_grad_mag > 5;
```


high threshold


high threshold
>> img_thresh = img_grad_mag > 30;

high threshold

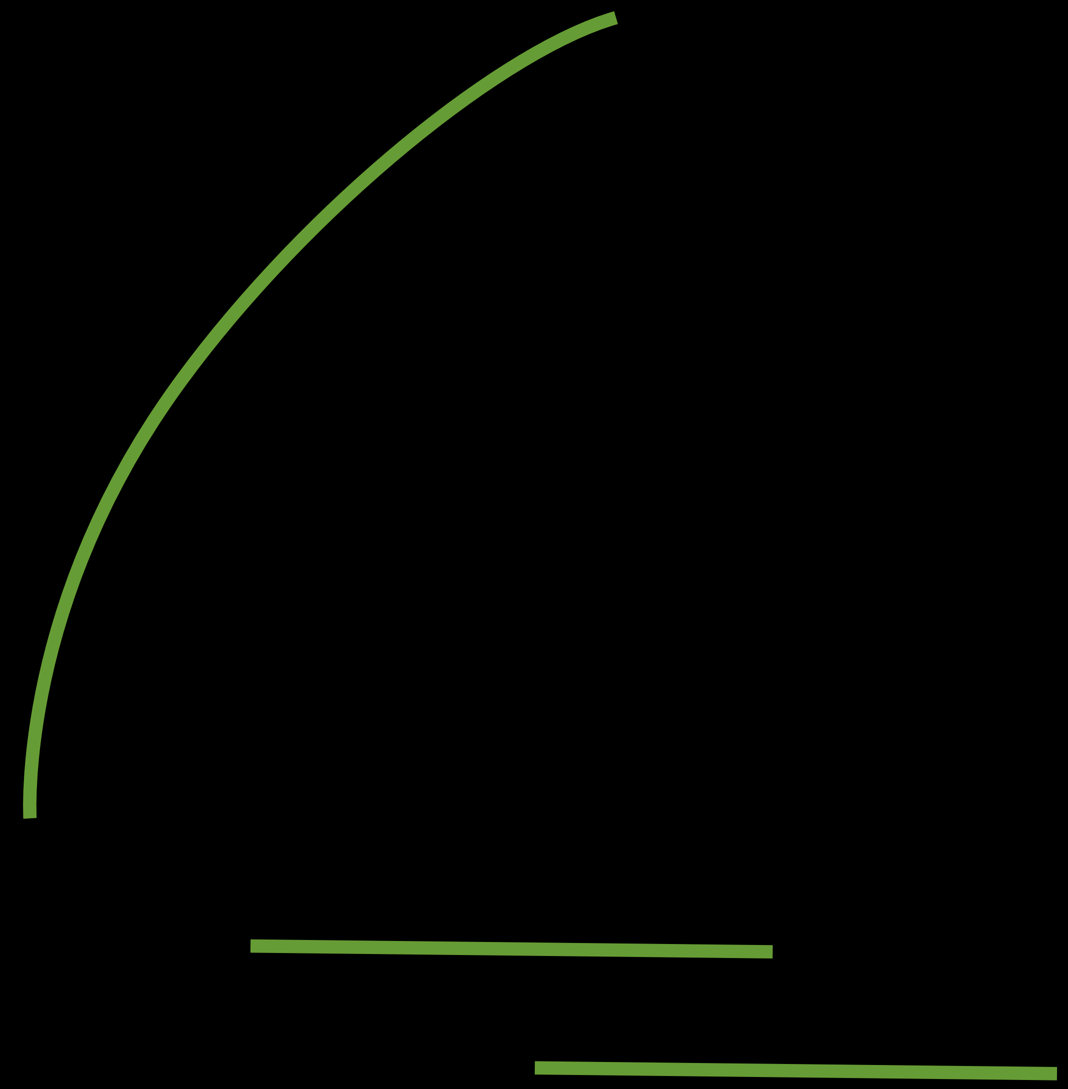
low threshold


non-max suppression

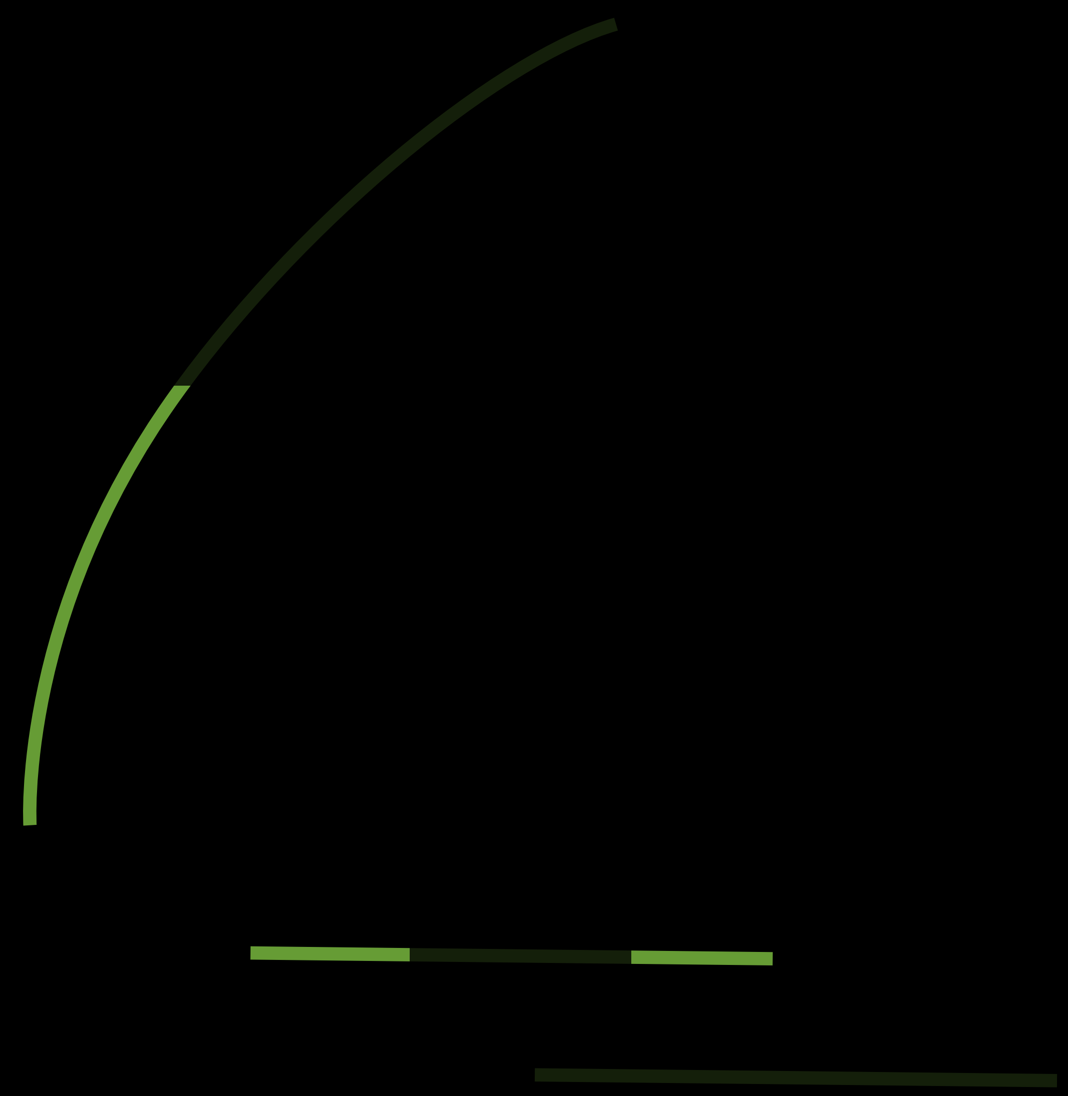

non-max suppression

non-max suppression

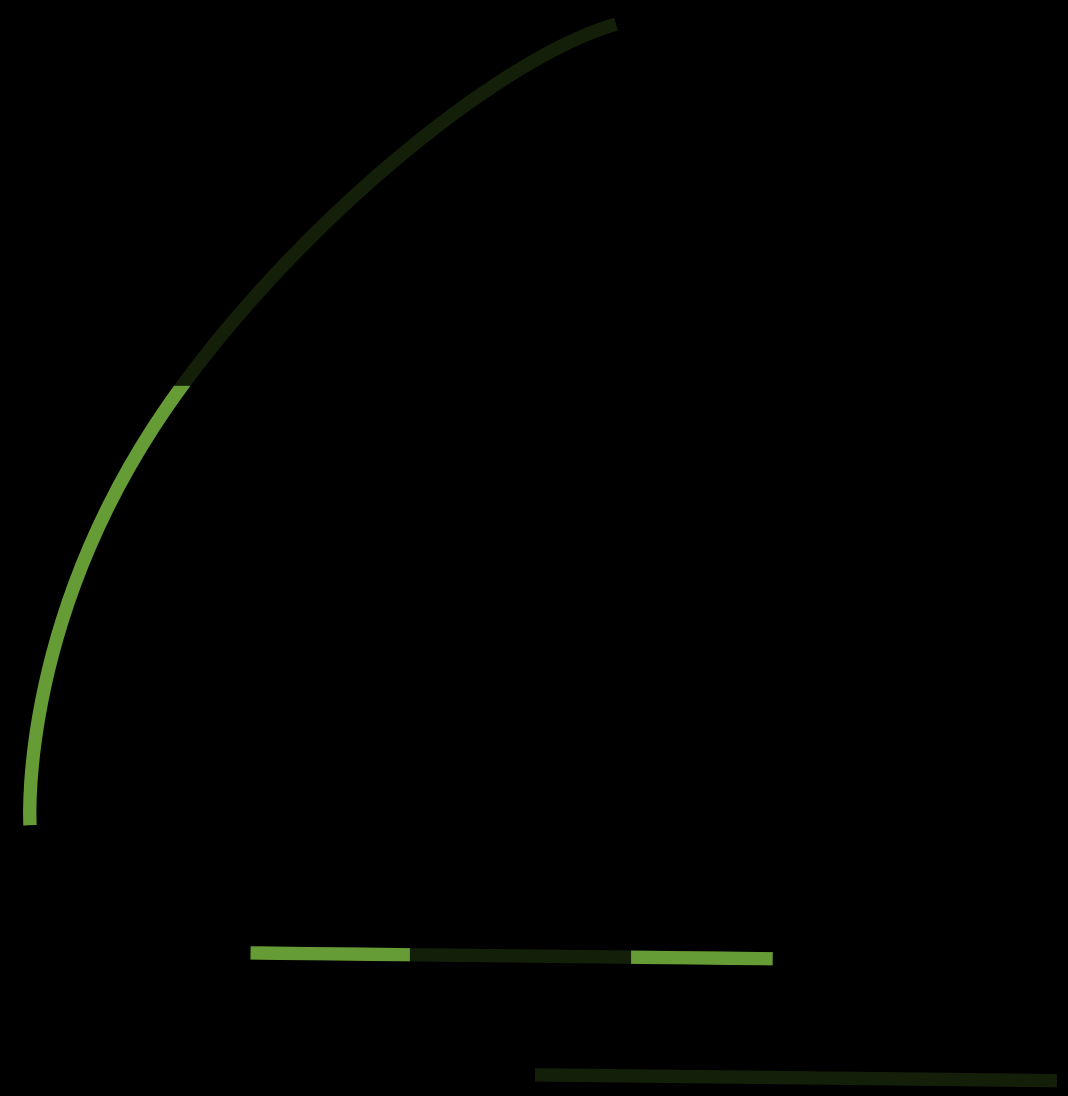

non-max suppression



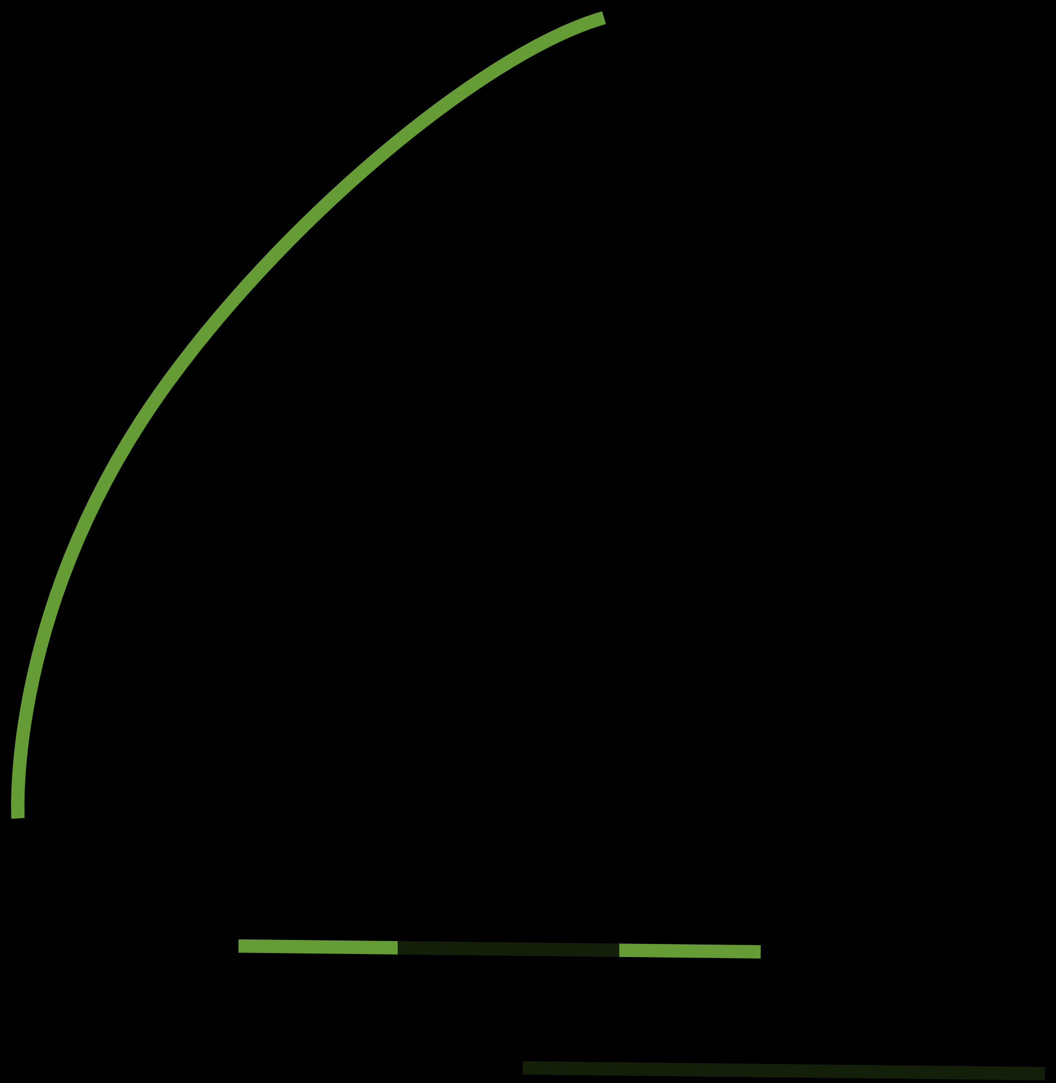
Define two thresholds, low and high



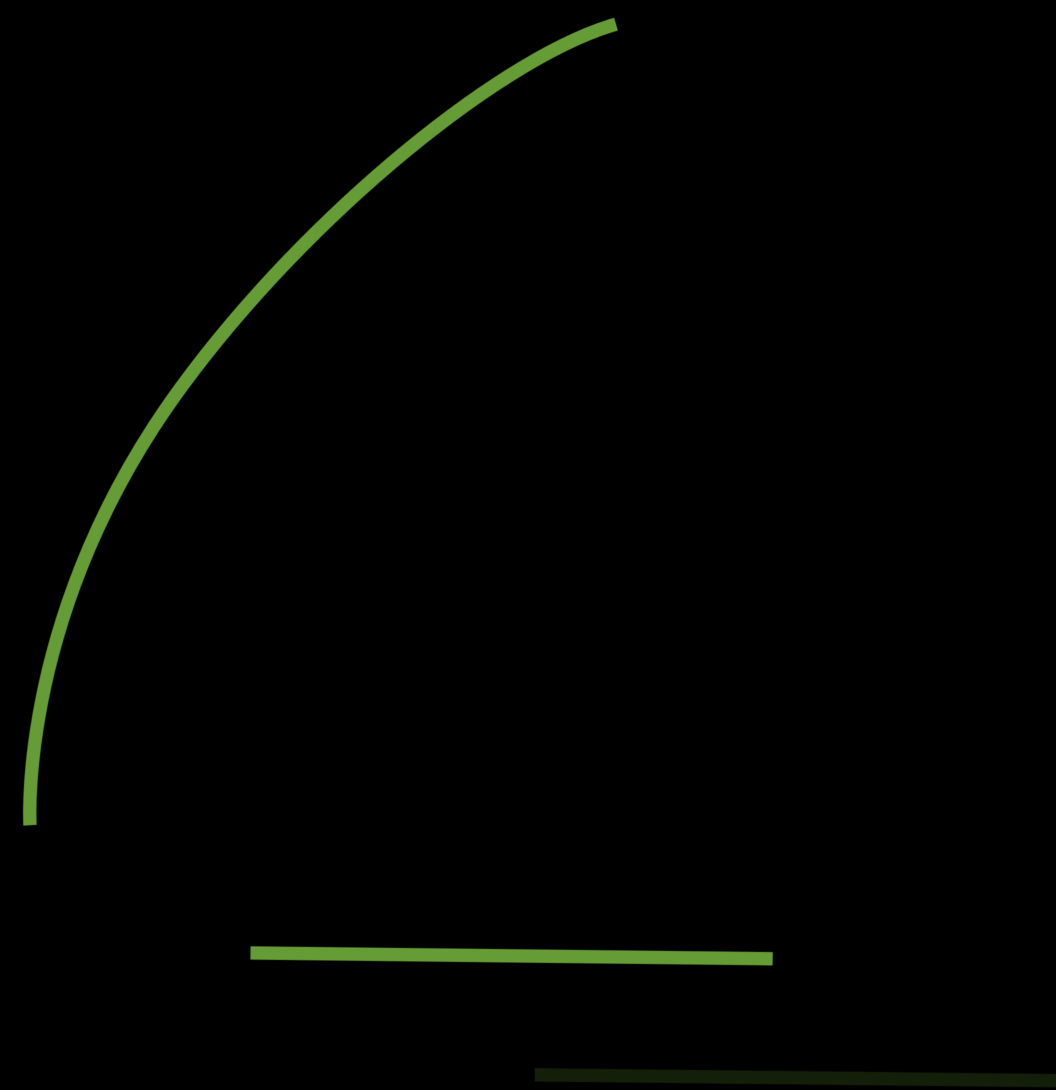
Define two thresholds, low and high



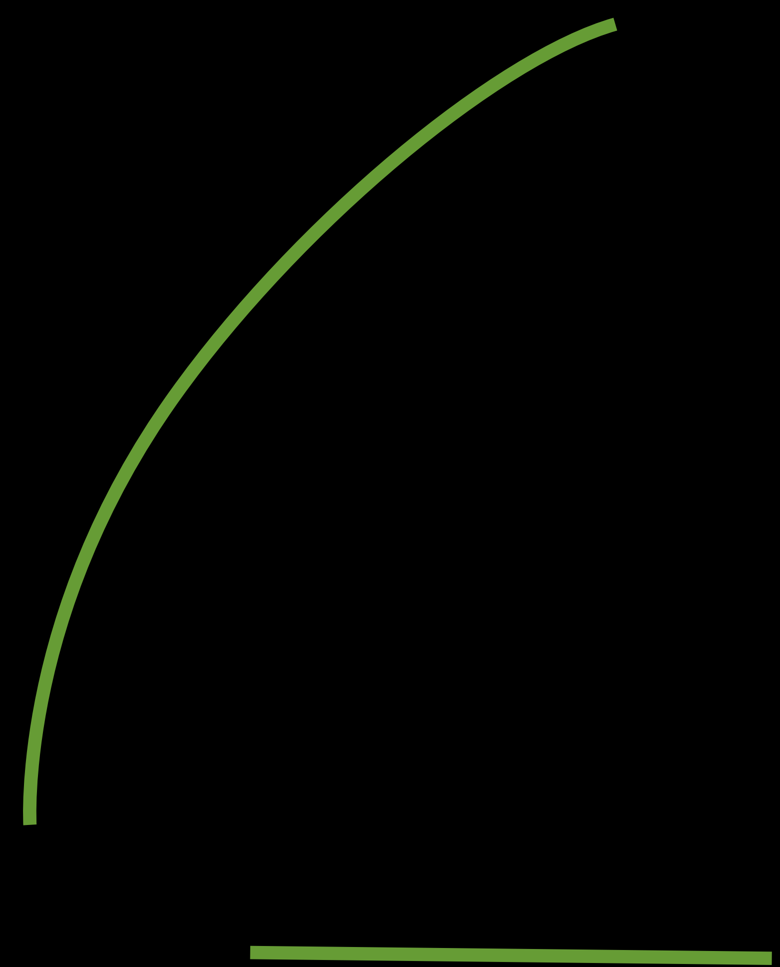
Starting from a pixel passing a high threshold, keep only pixels on the path that pass the low threshold



Starting from a pixel passing a high threshold, keep only pixels on the path that pass the low threshold



Starting from a pixel passing a high threshold, keep only pixels on the path that pass the low threshold



Starting from a pixel passing a high threshold, keep only pixels on the path that pass the low threshold


non-max suppression

hysteresis thresholding

too long; didn’t listen

tl;dl

Canny Detector
tl;dl

Canny Detector
tl;dl

1. **Filter image with x and y derivative of Gaussian**

1. Filter image with x and y derivative of Gaussian
2. Find magnitude and direction of gradient

Canny Detector
tl;dl

1. Filter image with x and y derivative of Gaussian
2. Find magnitude and direction of gradient
3. Perform non-maximum suppression

Canny Detector
tl;dl

1. Filter image with x and y derivative of Gaussian

2. Find magnitude and direction of gradient

3. Perform non-maximum suppression

4. Threshold and link


MATLAB break


Scale?


input
LOVE


Canny
$\sigma = 1$
LOVE

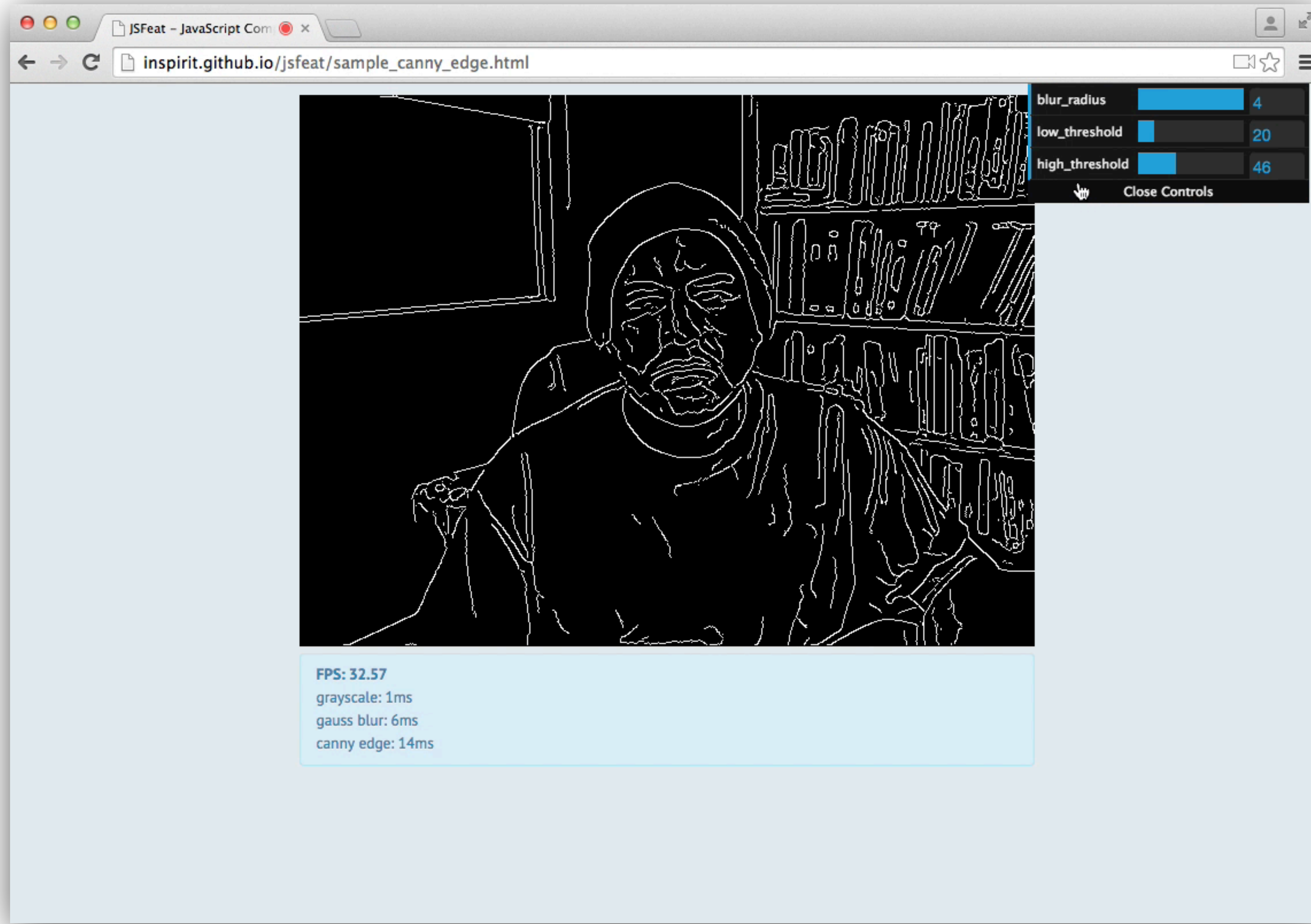

Canny
$\sigma = 5$

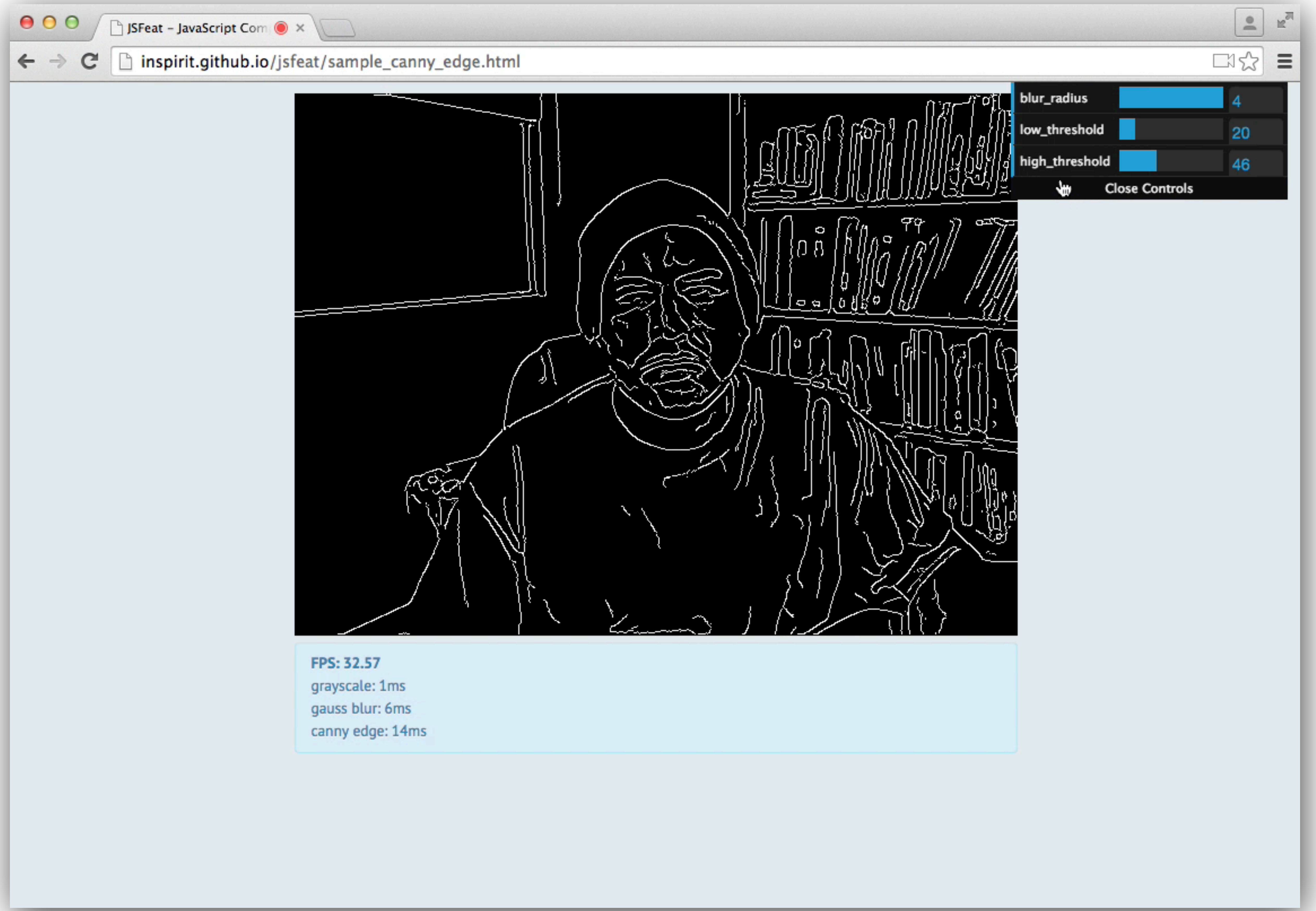

Canny
$\sigma = 5$


input
LOVE


Canny
$\sigma = 1$


Canny
$\sigma = 5$


JSFeat – JavaScript Com
inspirit.github.io/jsfeat/sample_canny_edge.html
blur_radius
4
low_threshold
20
high_threshold
46
Close Controls
FPS: 32.57
grayscale: 1ms
gauss blur: 6ms
canny edge: 14ms


JSFeat – JavaScript Com
inspirit.github.io/jsfeat/sample_canny_edge.html
blur_radius 4
low_threshold 20
high_threshold 46
Close Controls
FPS: 32.57
grayscale: 1ms
gauss blur: 6ms
canny edge: 14ms

One more thing...


Detect changes in colour?


Detect changes in texture?

How can we combine cues to detect object contours?

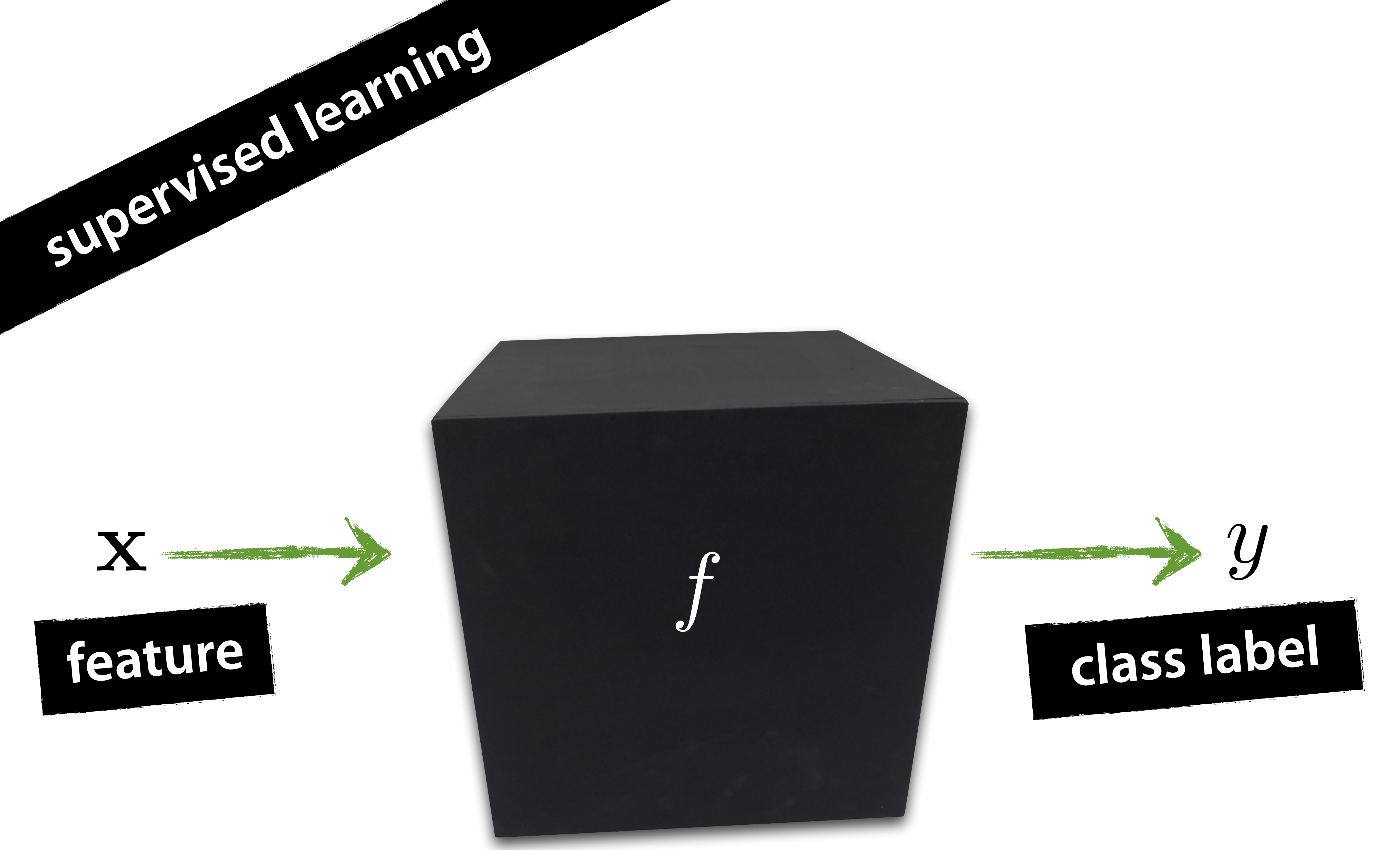
Machine learning

supervised learning

supervised learning

$\mathbf{x}$

feature



supervised learning
$\mathbf{x}$
feature
f
y
class label

Given training data, estimate the function f

Given training data, estimate the function f

$$\{(\mathbf{x}_i, y_i)\}$$

$\mathbf{x}$ →  → y

feature

class label

Given training data, estimate the function f

$$\{(\mathbf{x}_i, y_i)\}$$

Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues

David R. Martin, *Member*, *IEEE*, Charless C. Fowlkes, and Jitendra Malik, *Member*, *IEEE*

Abstract—The goal of this work is to accurately detect and localize boundaries in natural scenes using local image measurements.
We formulate features that respond to characteristic changes in brightness, color, and texture associated with natural boundaries. In

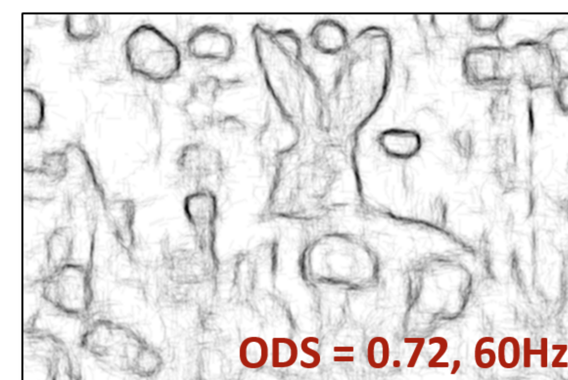
Structured Forests for Fast Edge Detection

Piotr Dollár
Microsoft Research
pdollar@microsoft.com

C. Lawrence Zitnick
Microsoft Research
larryz@microsoft.com

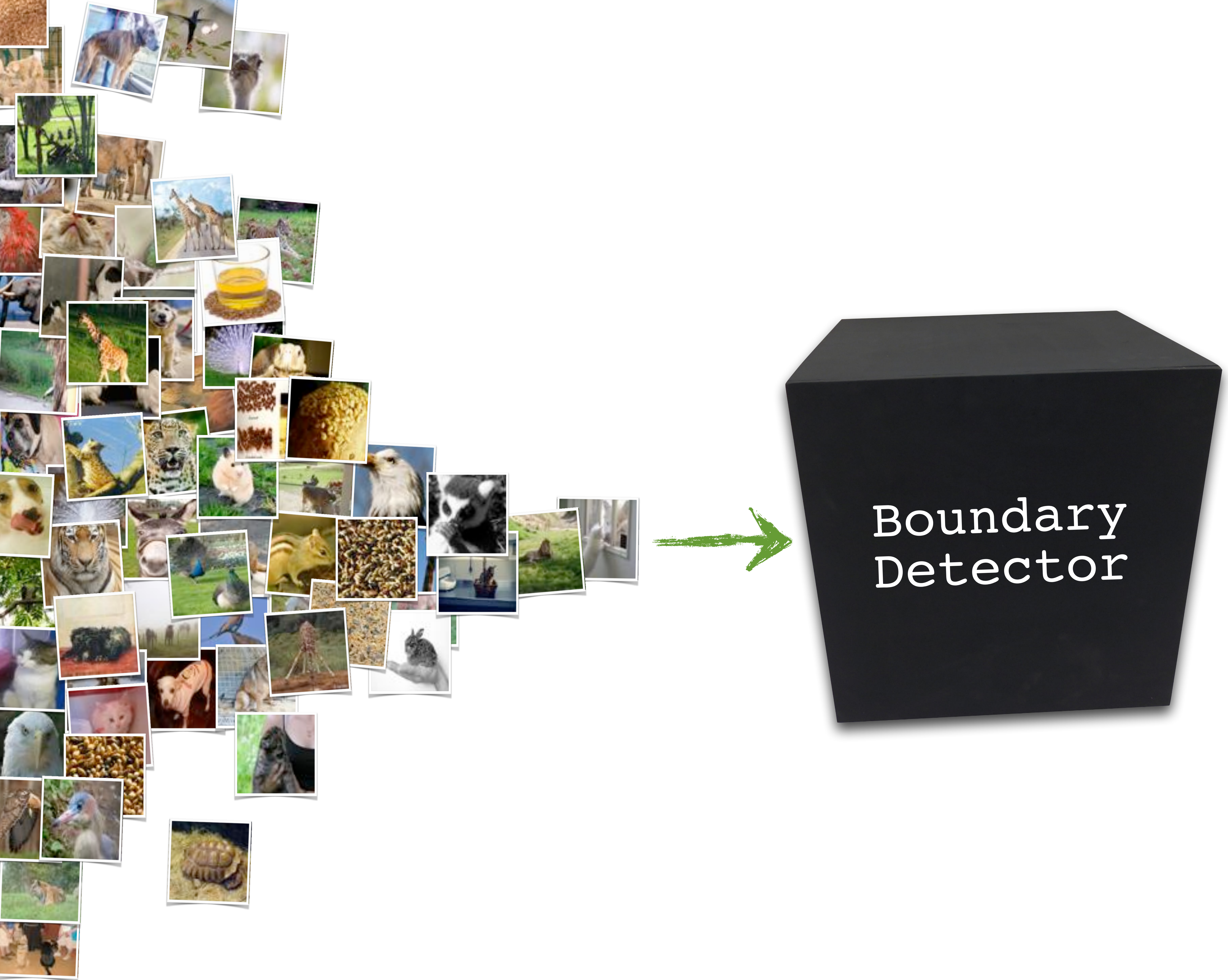
Abstract

Edge detection is a critical component of many vision systems, including object detectors and image segmentation algorithms. Patches of edges exhibit well-known forms of local structure, such as straight lines or T-junctions. In this




handcrafted features

DEEP
learning


Boundary
Detector

PUSHING THE BOUNDARIES OF BOUNDARY DETECTION USING DEEP LEARNING

Iasonas Kokkinos
Center for Visual Computing
CentraleSupélec and INRIA
Chatenay-Malabry, 92095, France
{iasonas.kokkinos}@ecp.fr

ABSTRACT

In this work we show that adapting Deep Convolutional Neural Network training to the task of boundary detection can result in substantial improvements over the current state-of-the-art in boundary detection

Pushing the Boundaries of Boundary Detection using Deep Learning

Iasonas Kokkinos
Center for Visual Computing
CentraleSupélec and INRIA
Chatenay-Malabry, 92095, France
{iasonas.kokkinos}@ecp.fr

Abstract

In this work we show that adapting Deep Convolutional Neural Network training
... can result in substantial improvements over the

Surpasses human accuracy on challenging benchmark!

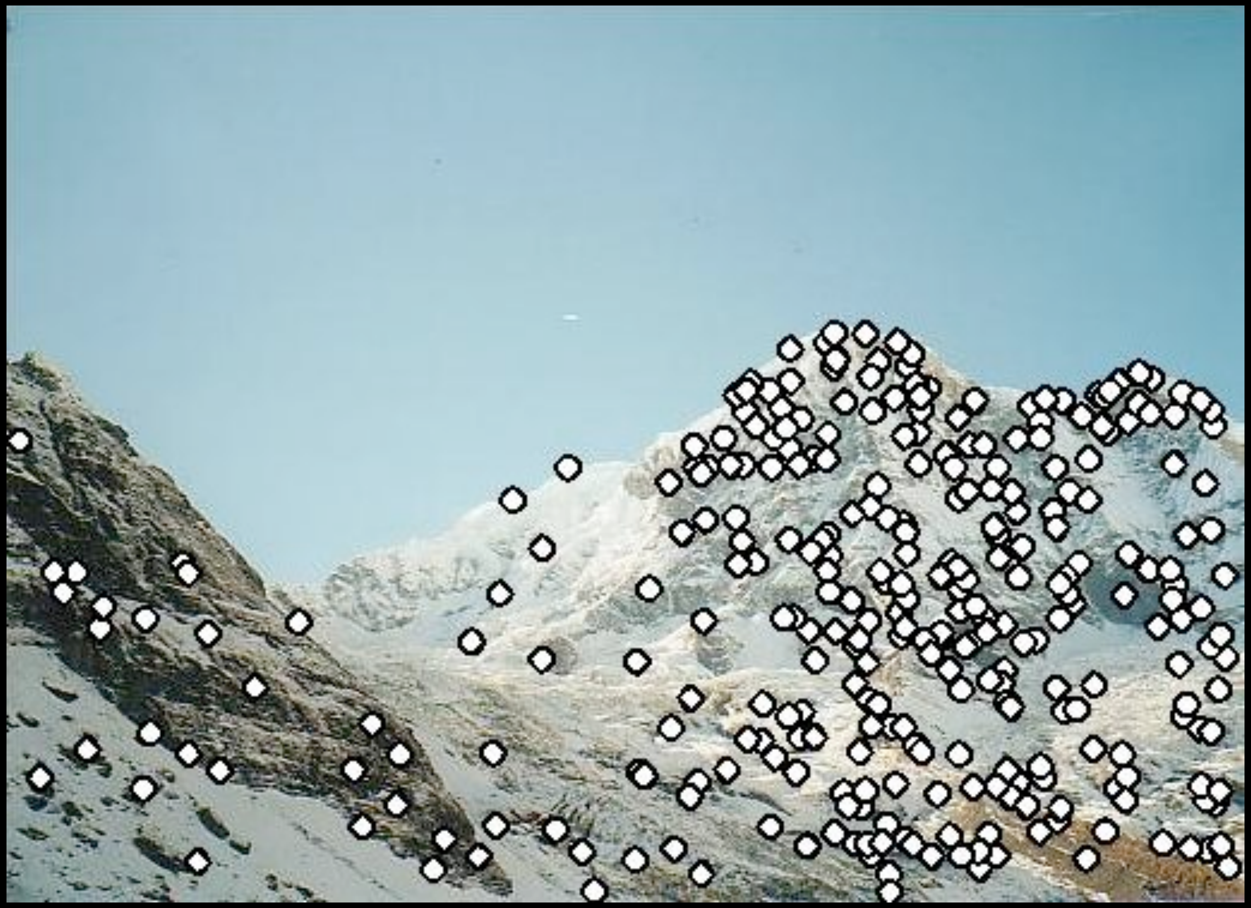
Intro to

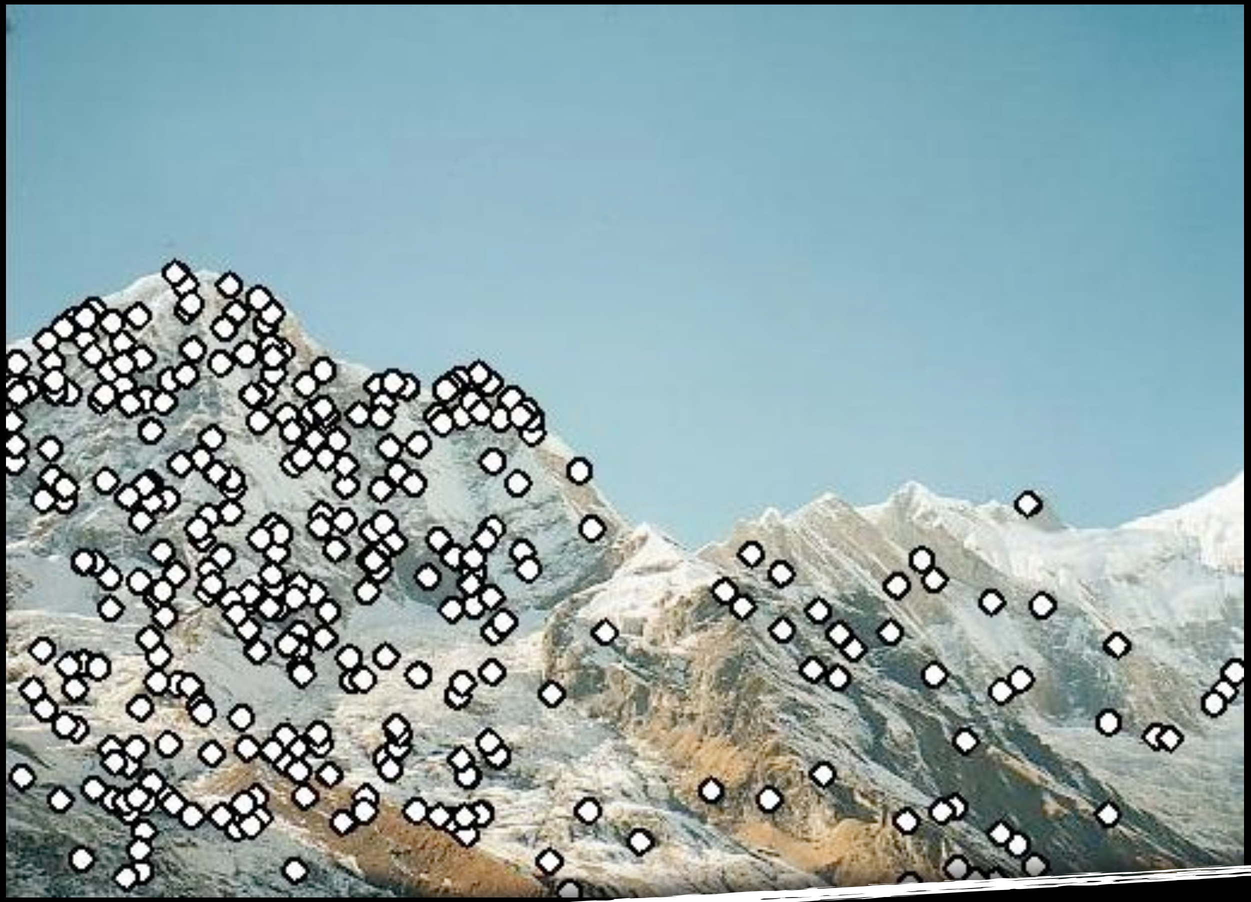
Computer Vision

Feature detection

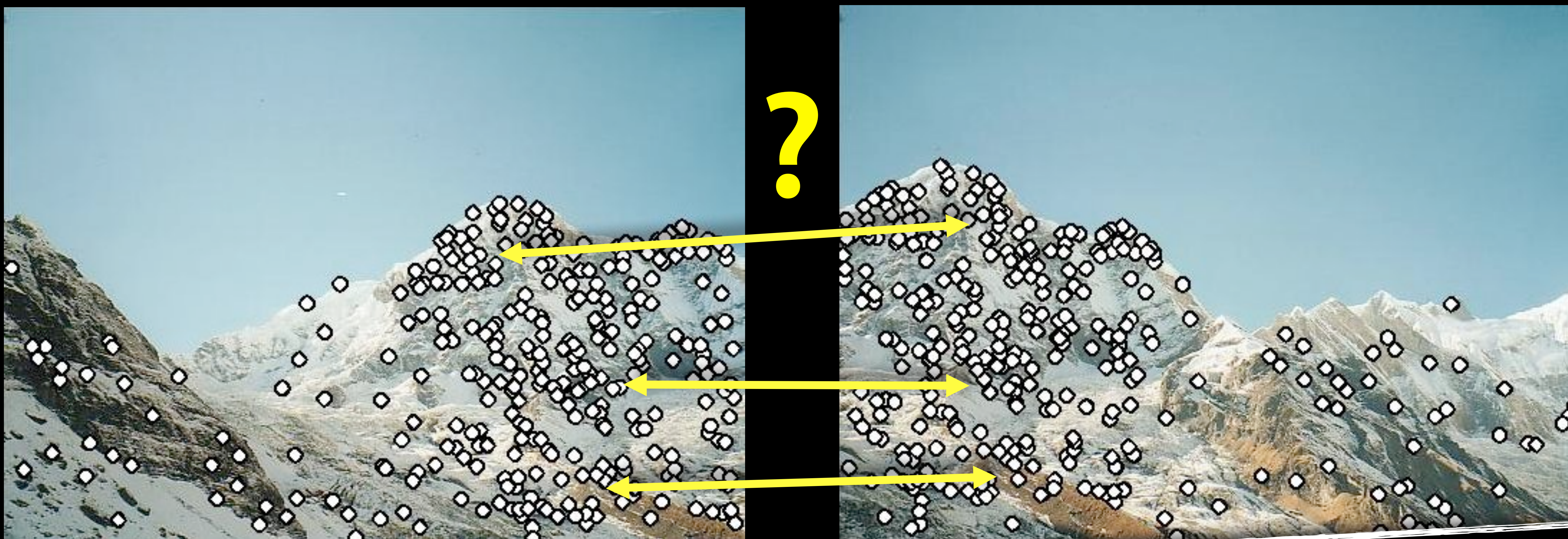


What stuff in an image matches with stuff in another?

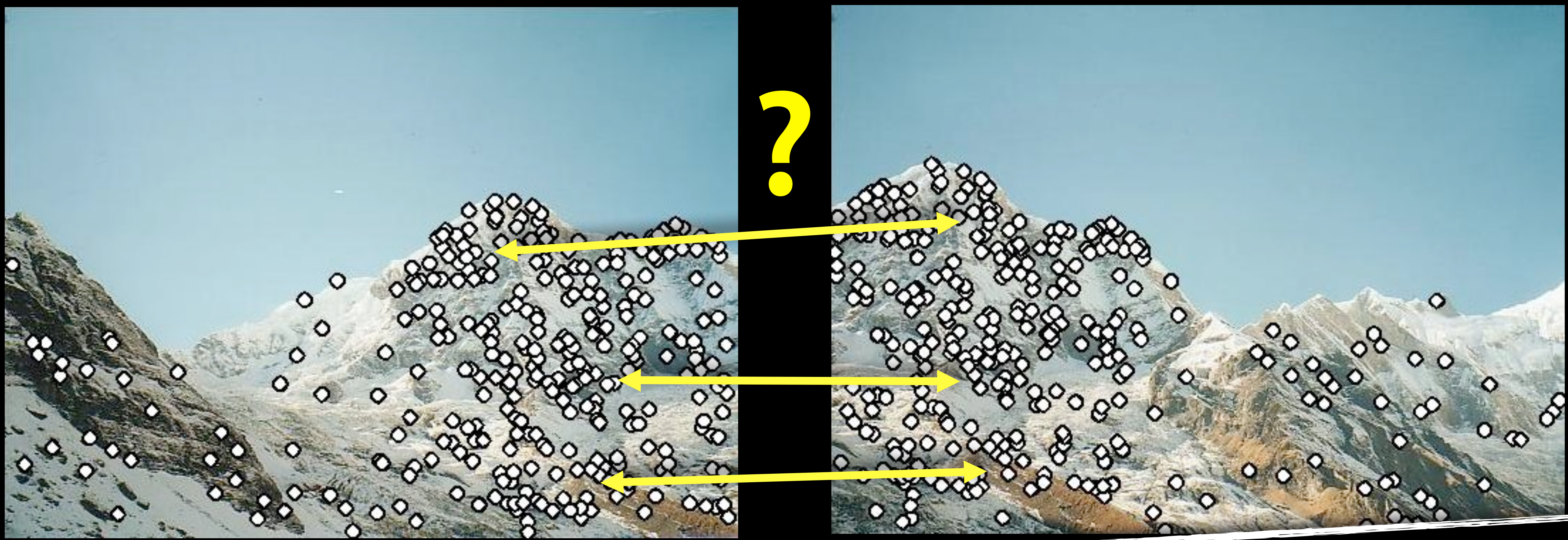




What stuff in an image matches with stuff in another?

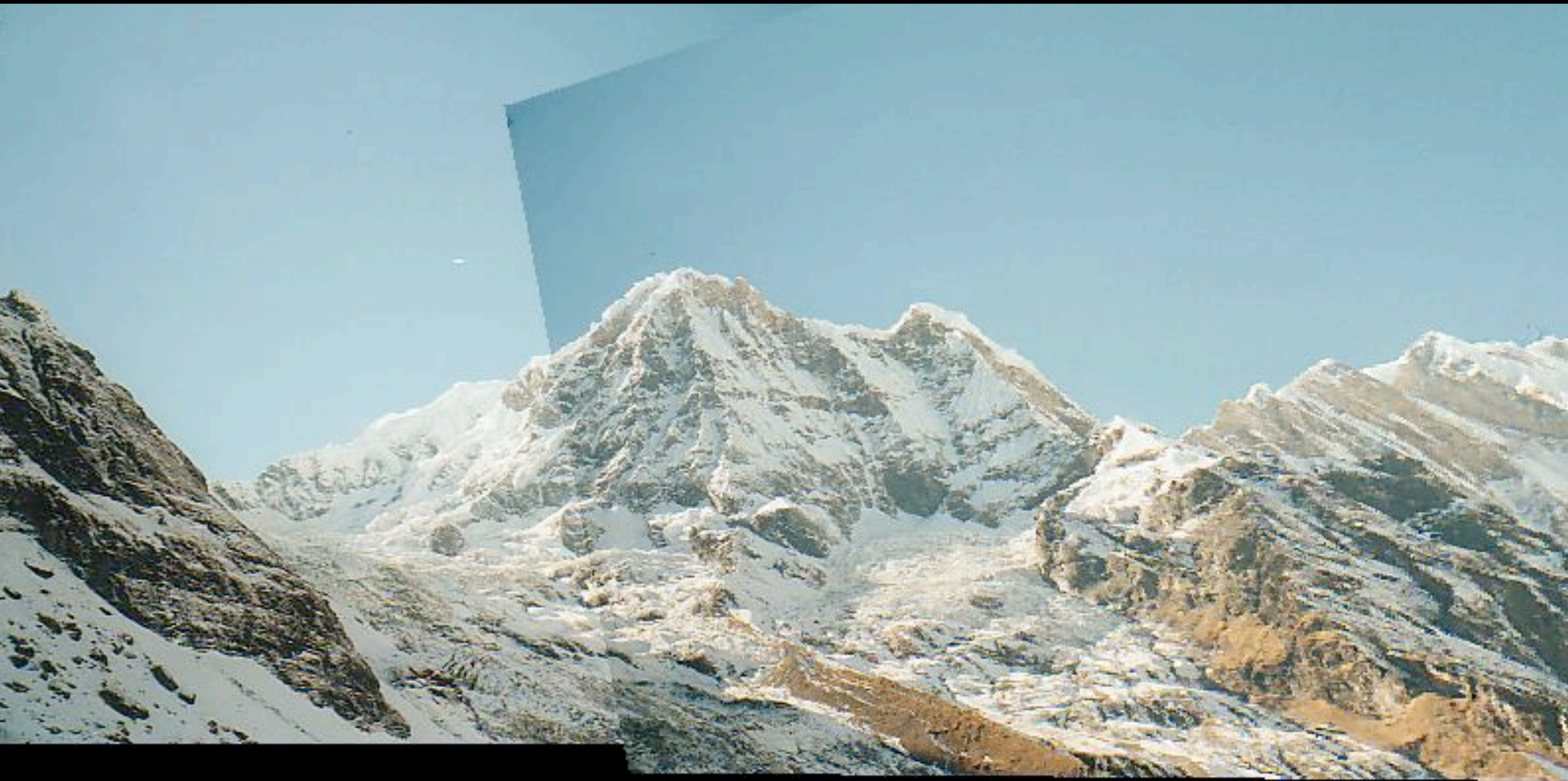


What stuff in an image matches with stuff in another?



What stuff in an image matches with stuff in another?

Find matching pairs and align



What makes a “good” feature?

REPEATABILITY

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


input image

photometric transformation

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


input image

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

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

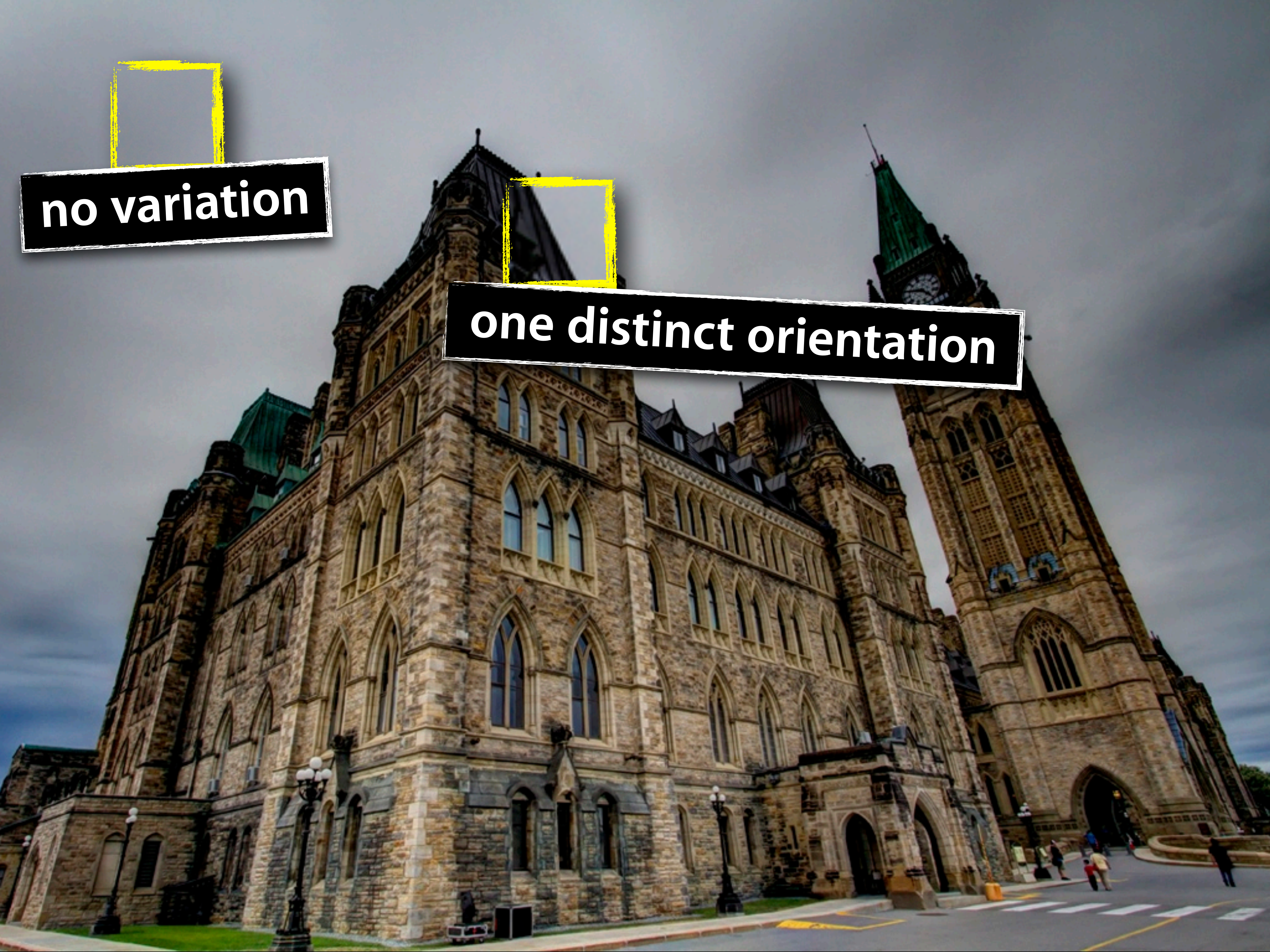
What is a Corner?

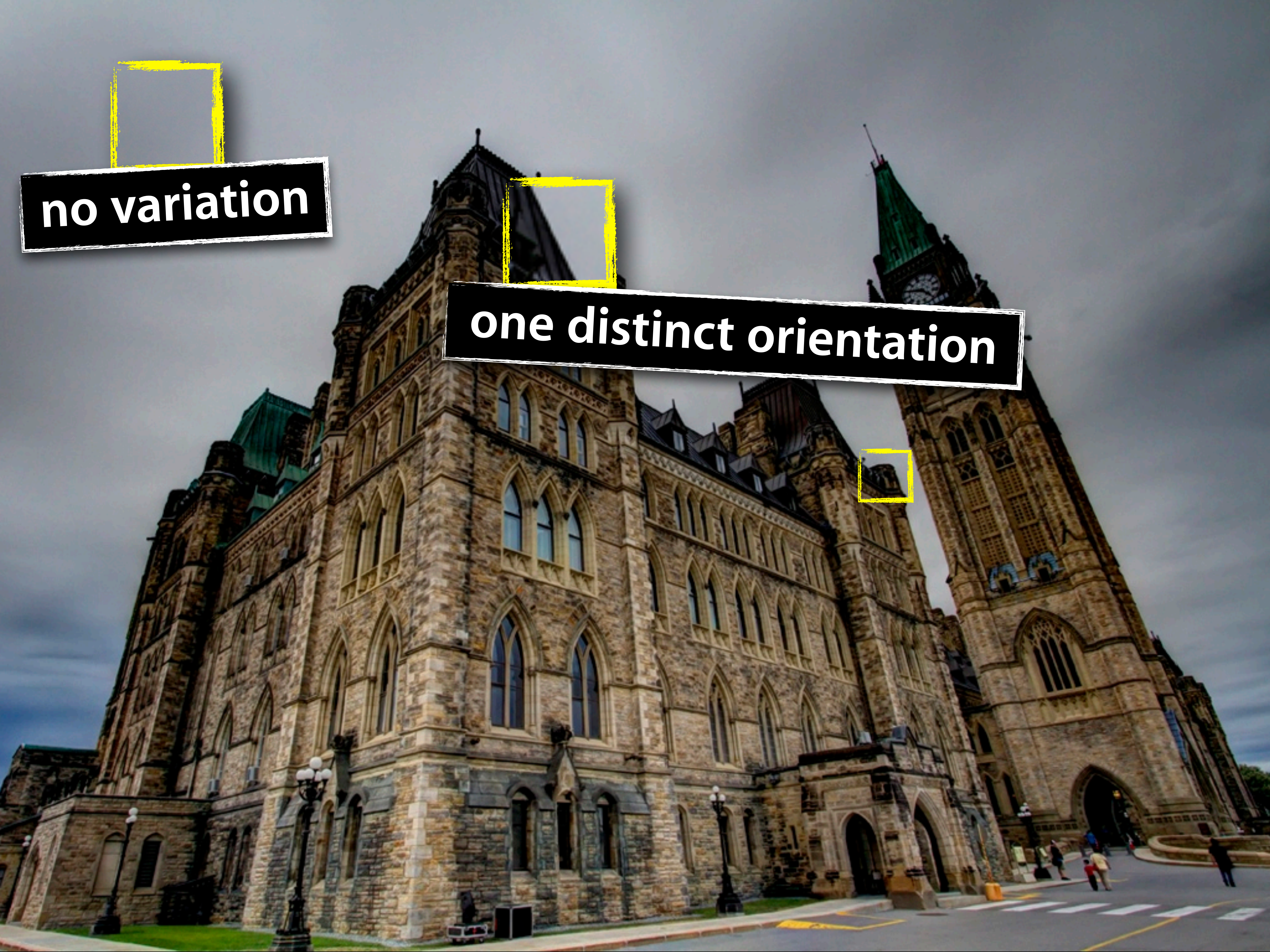
What is a **Corner?**

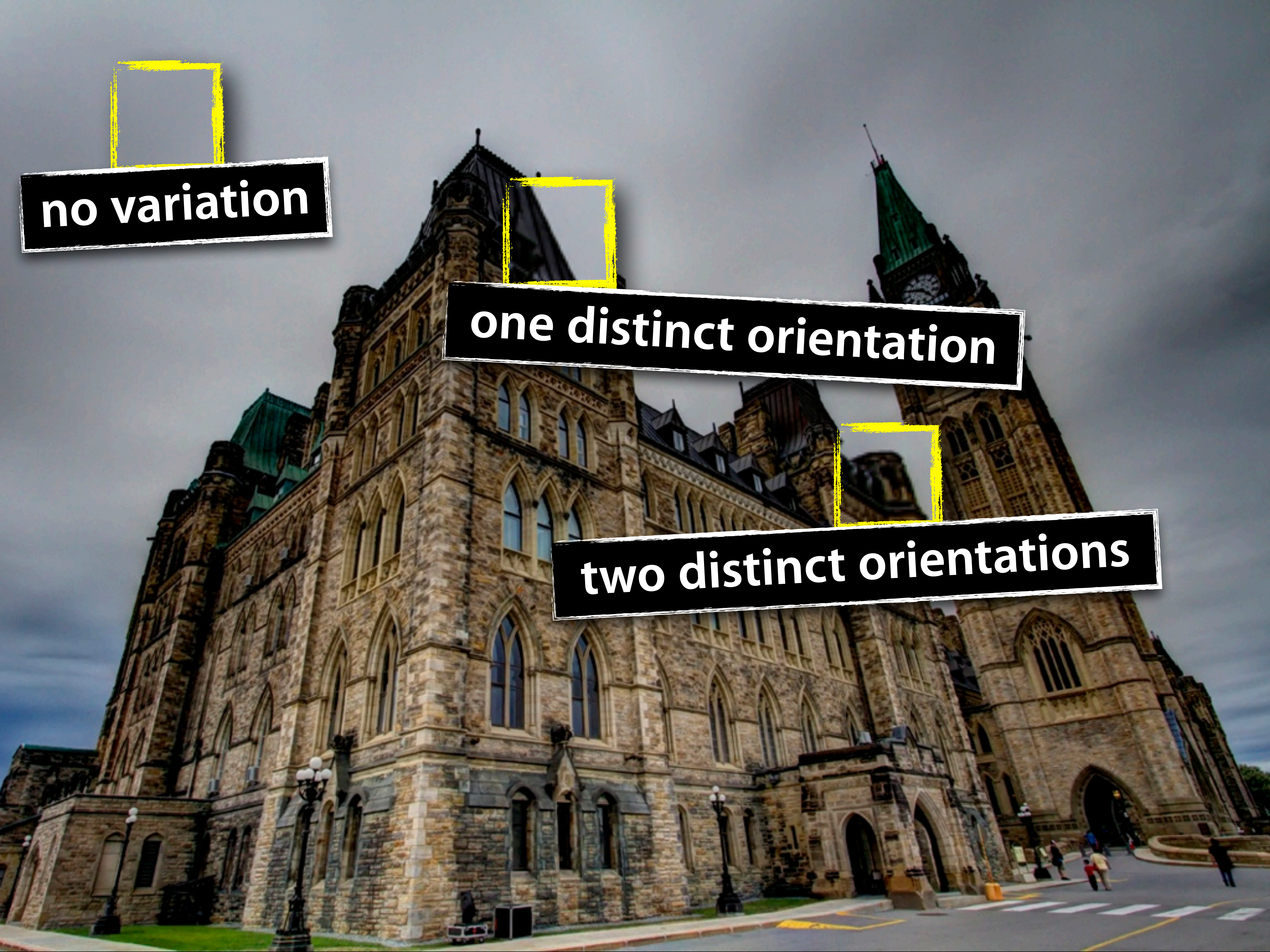
locations where two distinct
oriented structures in the image are present


no variation


no variation


no variation
one distinct orientation

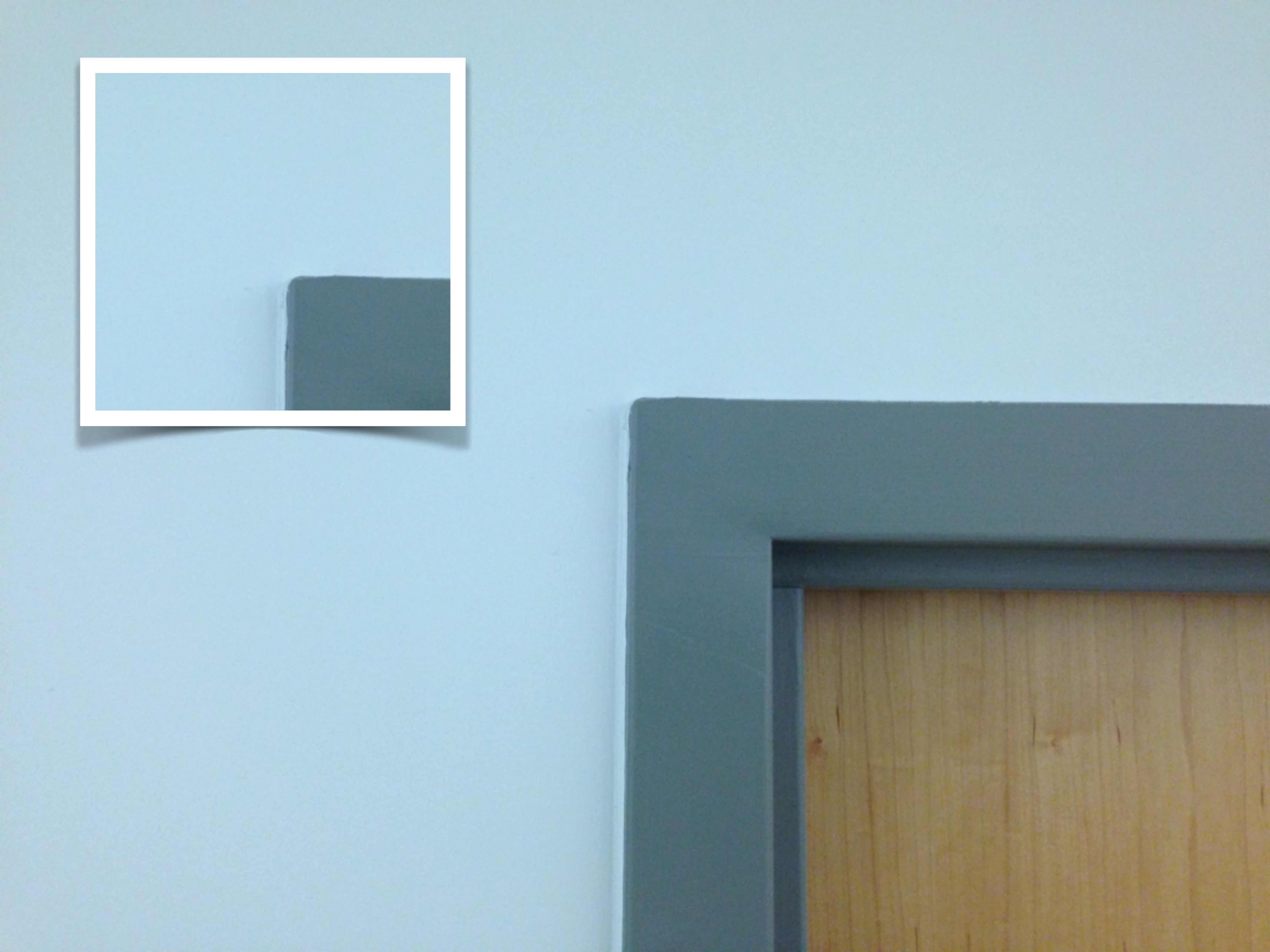

no variation
one distinct orientation


no variation
one distinct orientation
two distinct orientations

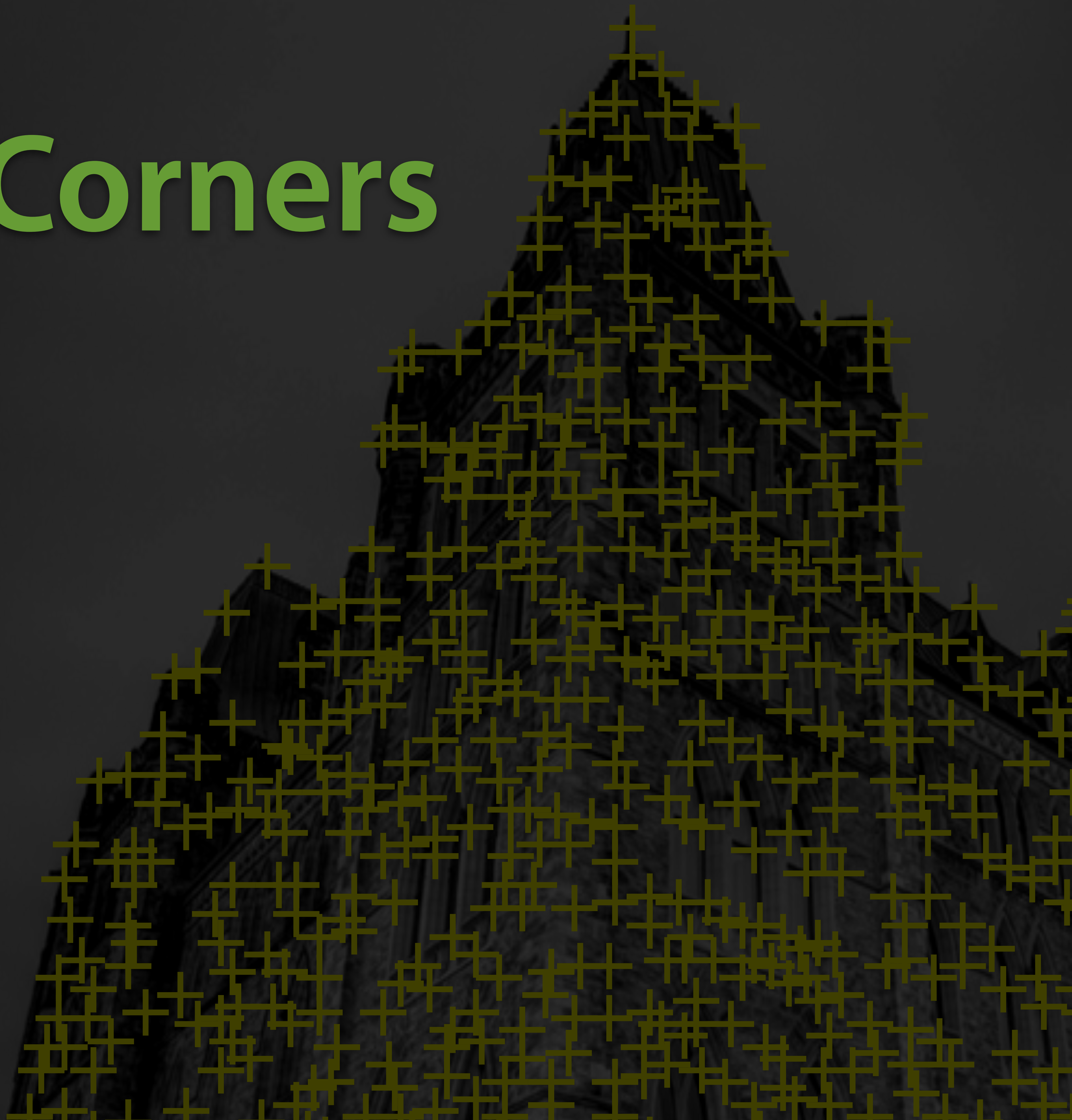
Why bother with corners?

Why bother with corners?

localizes spatial patterns in the image



Harris Corners



A COMBINED CORNER AND EDGE DETECTOR

Chris Harris & Mike Stephens

Plessey Research Roke Manor, United Kingdom
© The Plessey Company plc. 1988

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].

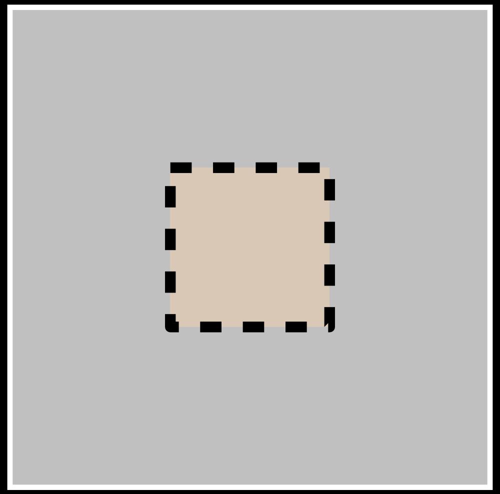
Alvey Vision Conference 1988

Intuition

Analyze the local variation of signal

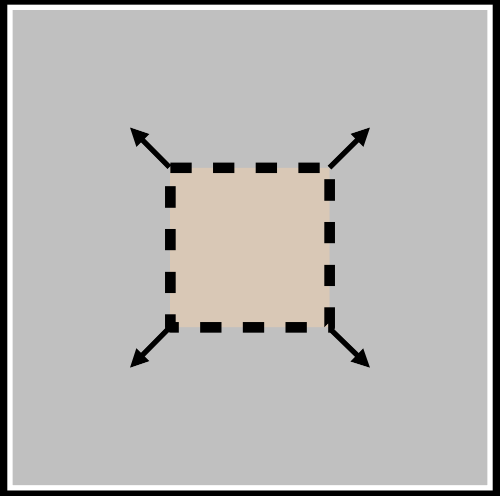
Intuition

Analyze the local variation of signal



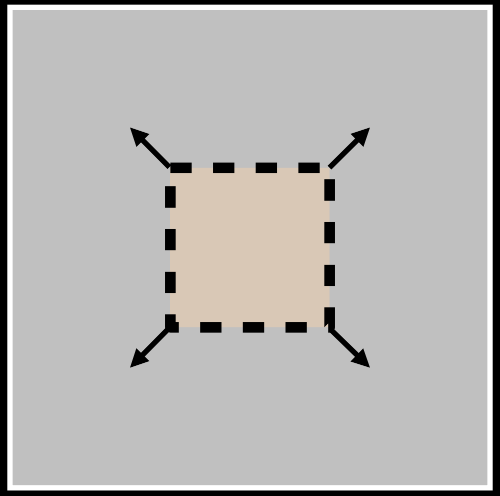
Intuition

Analyze the local variation of signal



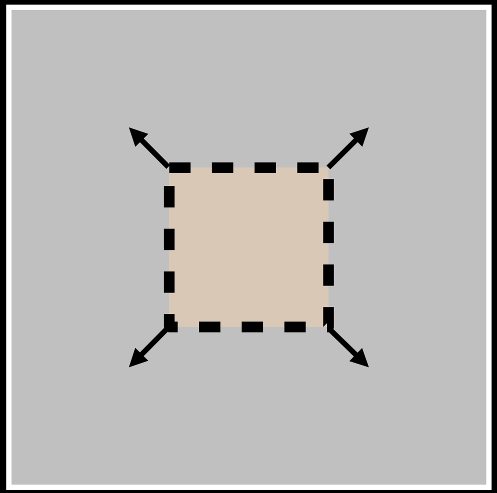
Intuition

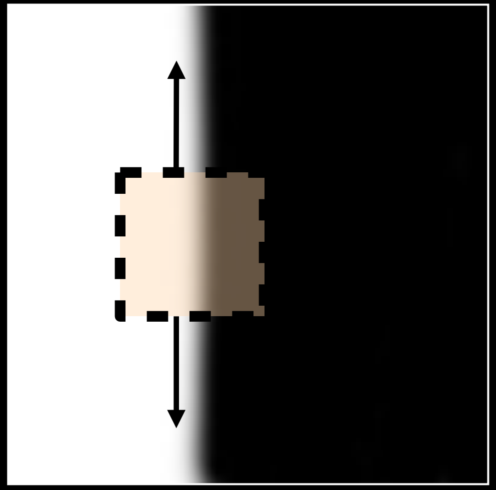
Analyze the local variation of signal



"flat" region
no change in all directions

Analyze the local variation of signal

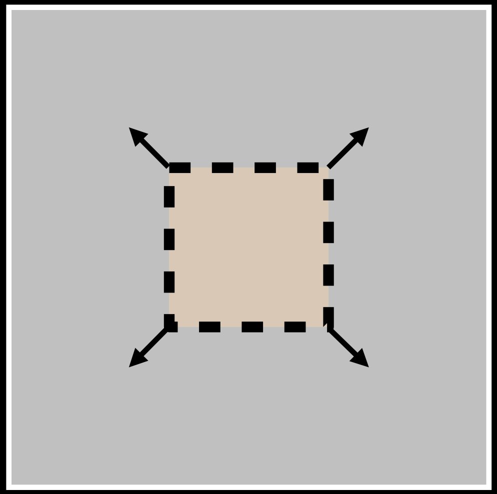




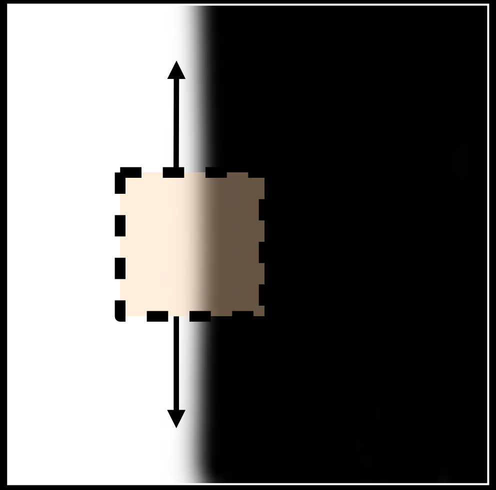
"flat" region
no change in all directions

Intuition

Analyze the local variation of signal

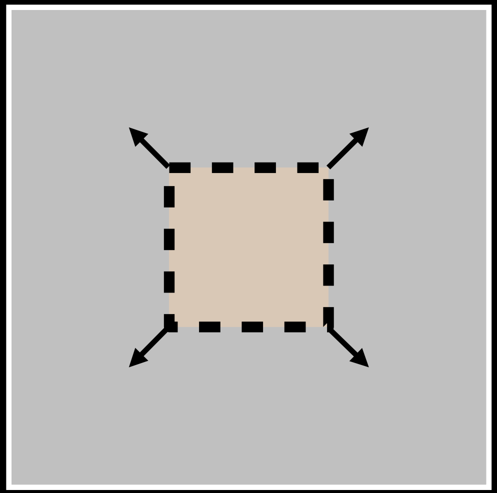


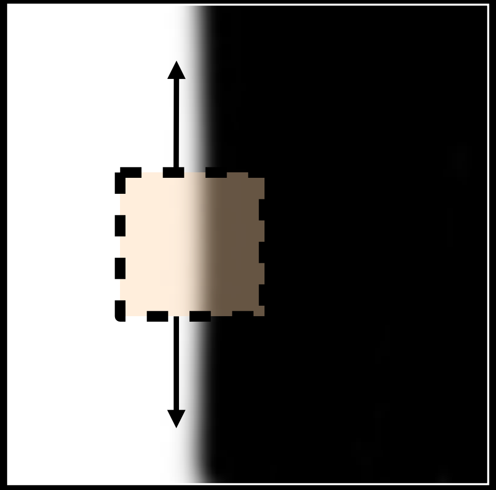
"flat" region
no change in all directions

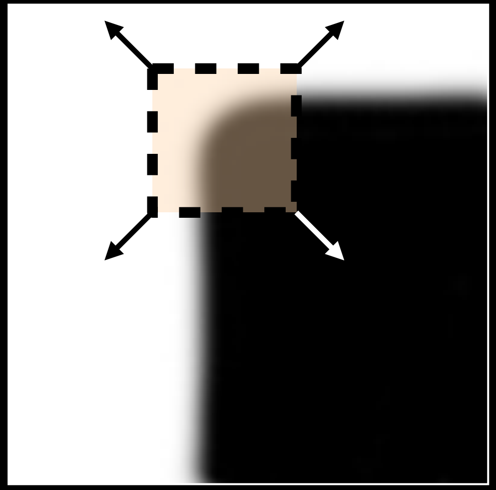


"edge"
no change along the edge direction

Analyze the local variation of signal



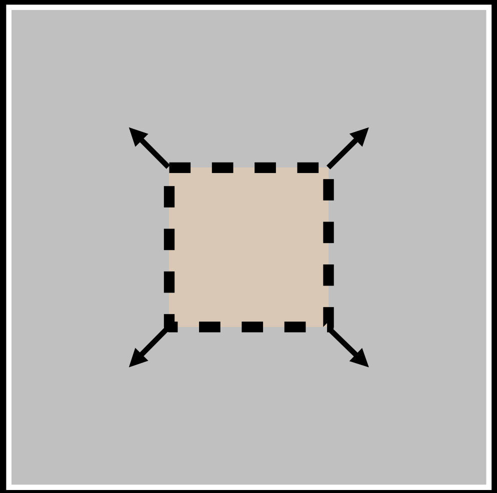




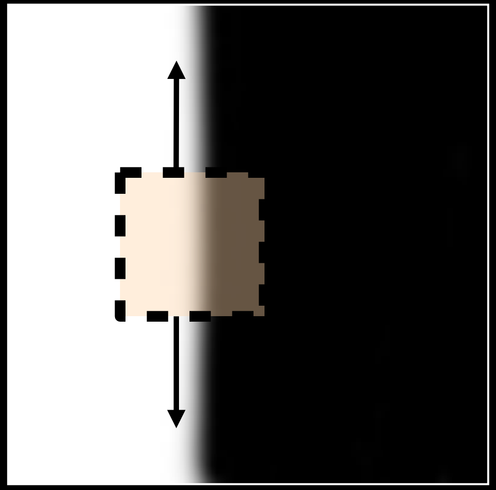
“flat” region
no change in
all directions

“edge”
no change
along the
edge direction

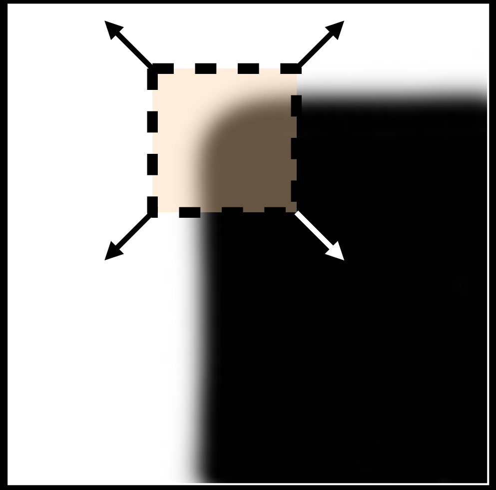
Analyze the local variation of signal



"flat" region
no change in all directions



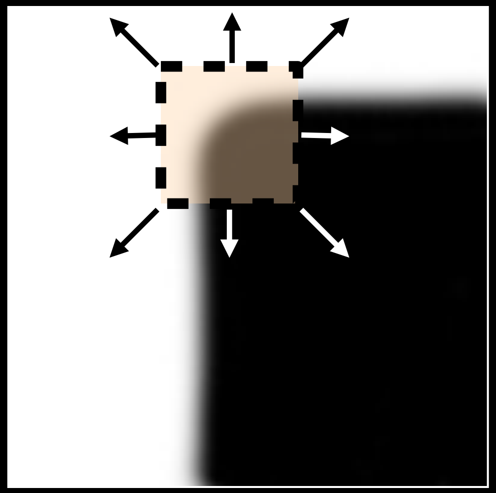
"edge"
no change along the edge direction



"corner"
significant change in all directions

Derivation

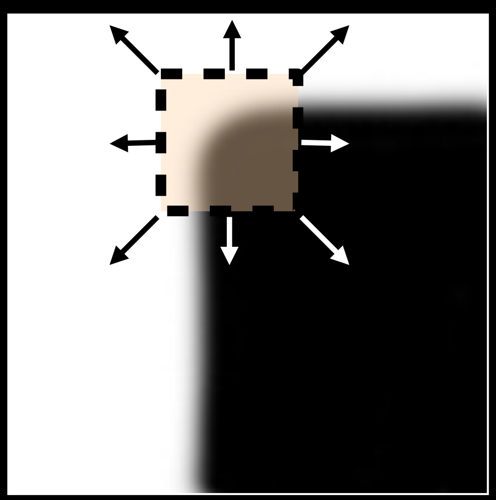
Variation of intensity for the shift $(\Delta x, \Delta y)$



Derivation

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

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

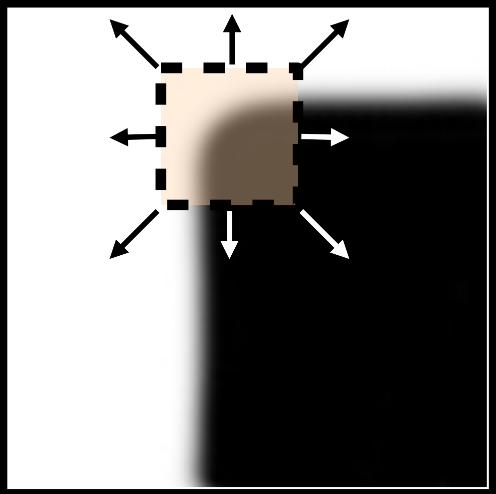


Derivation

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

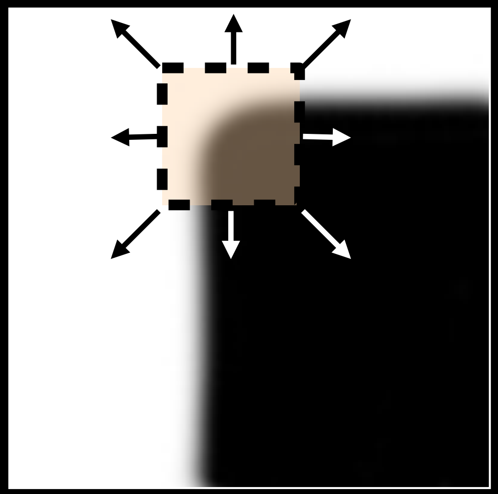


Derivation

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

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

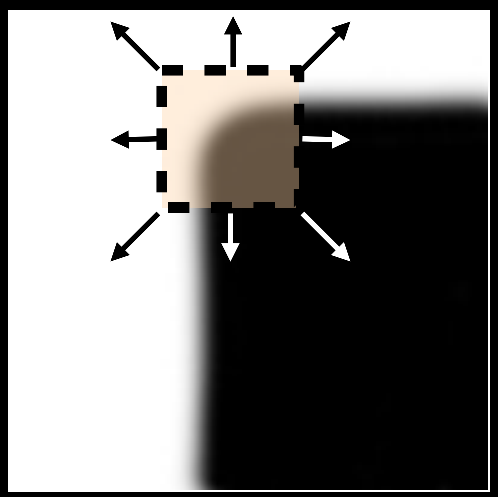
shifted image



Derivation

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

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



Derivation

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

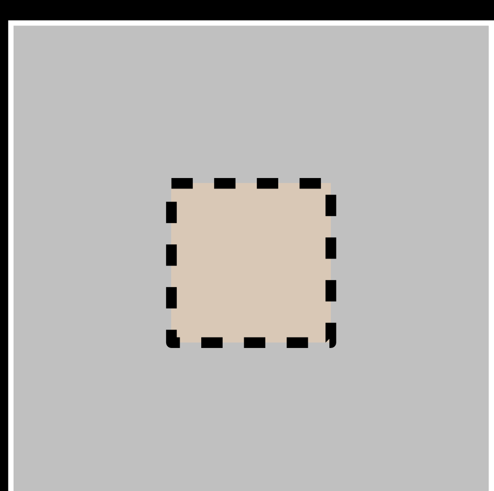
$$E(\Delta x, \Delta y) = \sum_{x,y} \underbrace{w(x,y)}_{\text{window}} [I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

Possible windowing functions

Derivation

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

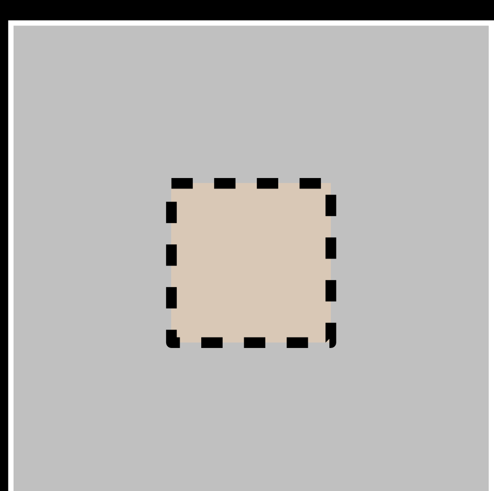
$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$



Derivation

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

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

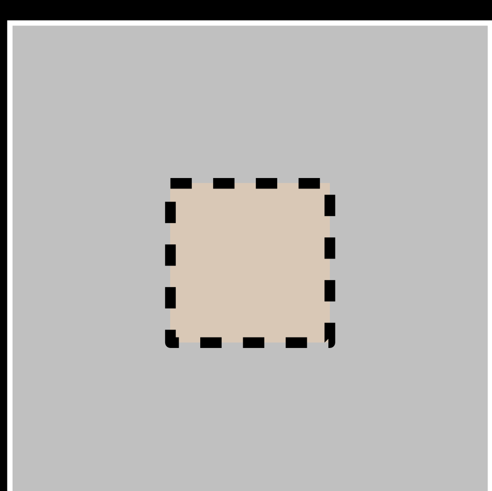


What is the response output?

Derivation

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

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



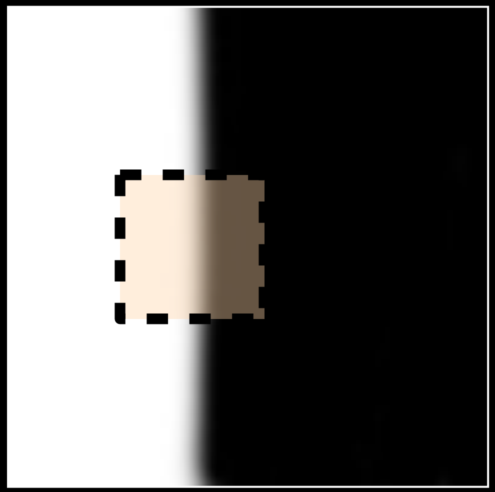
Nearly constant patches

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

Derivation

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

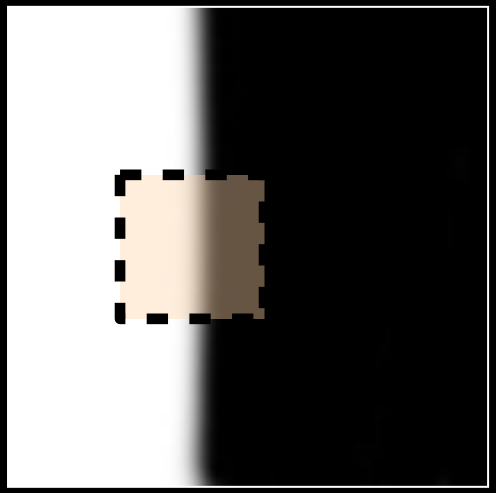
$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$



Derivation

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

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

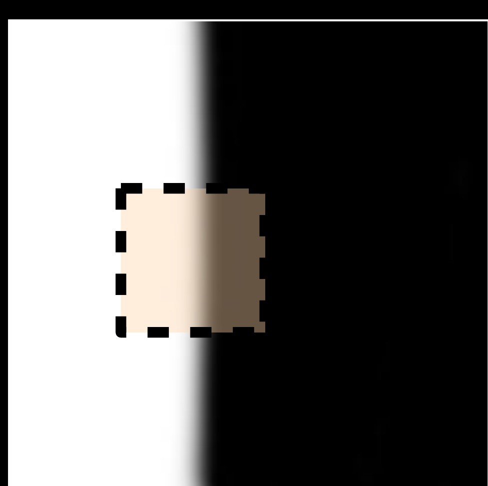


What is the response output?

Derivation

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

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



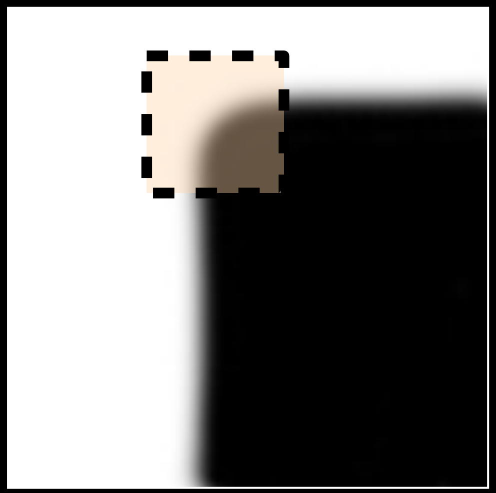
"edge" patches

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

large variation along one orientation

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

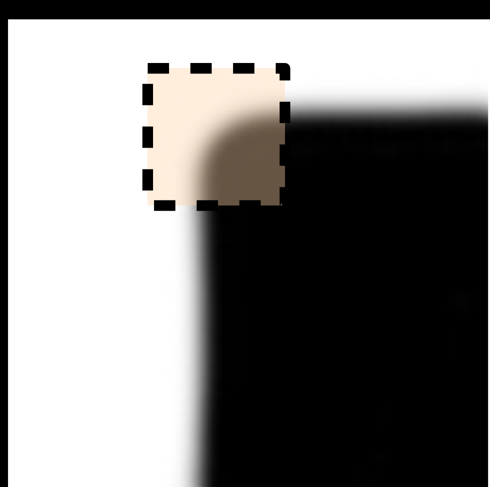
$$E(\Delta x, \Delta y) = \sum_{x,y} w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$



Derivation

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

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

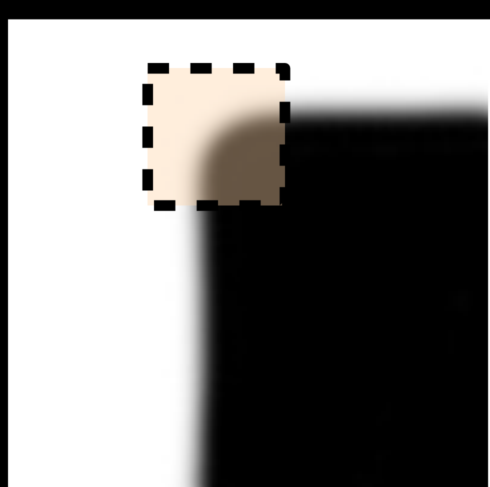


What is the response output?

Derivation

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

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



"corner" patches

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

large variation along two orientations

Derivation

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

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

Derivation

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

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

Review: Taylor series

1D case

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

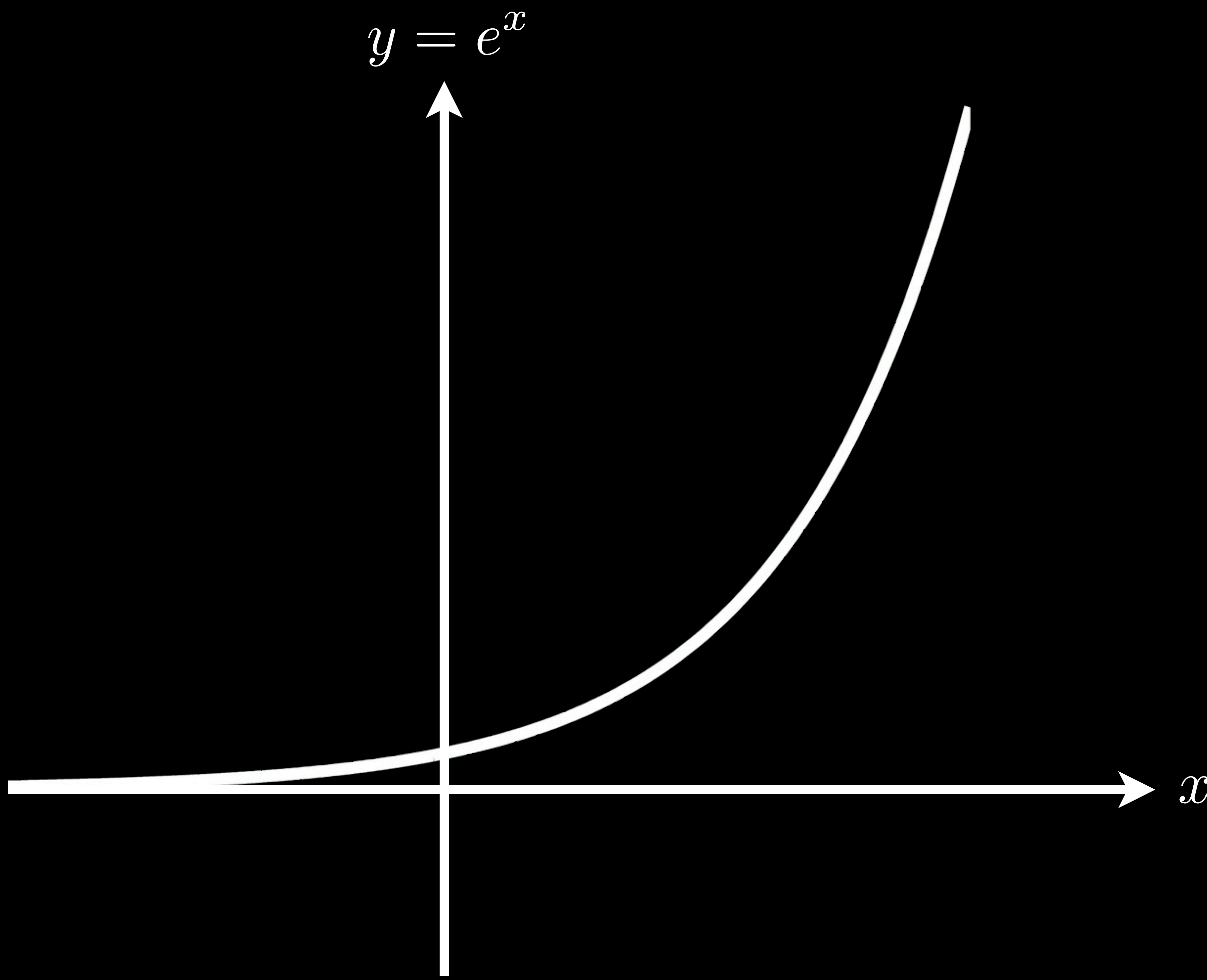
1D case

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

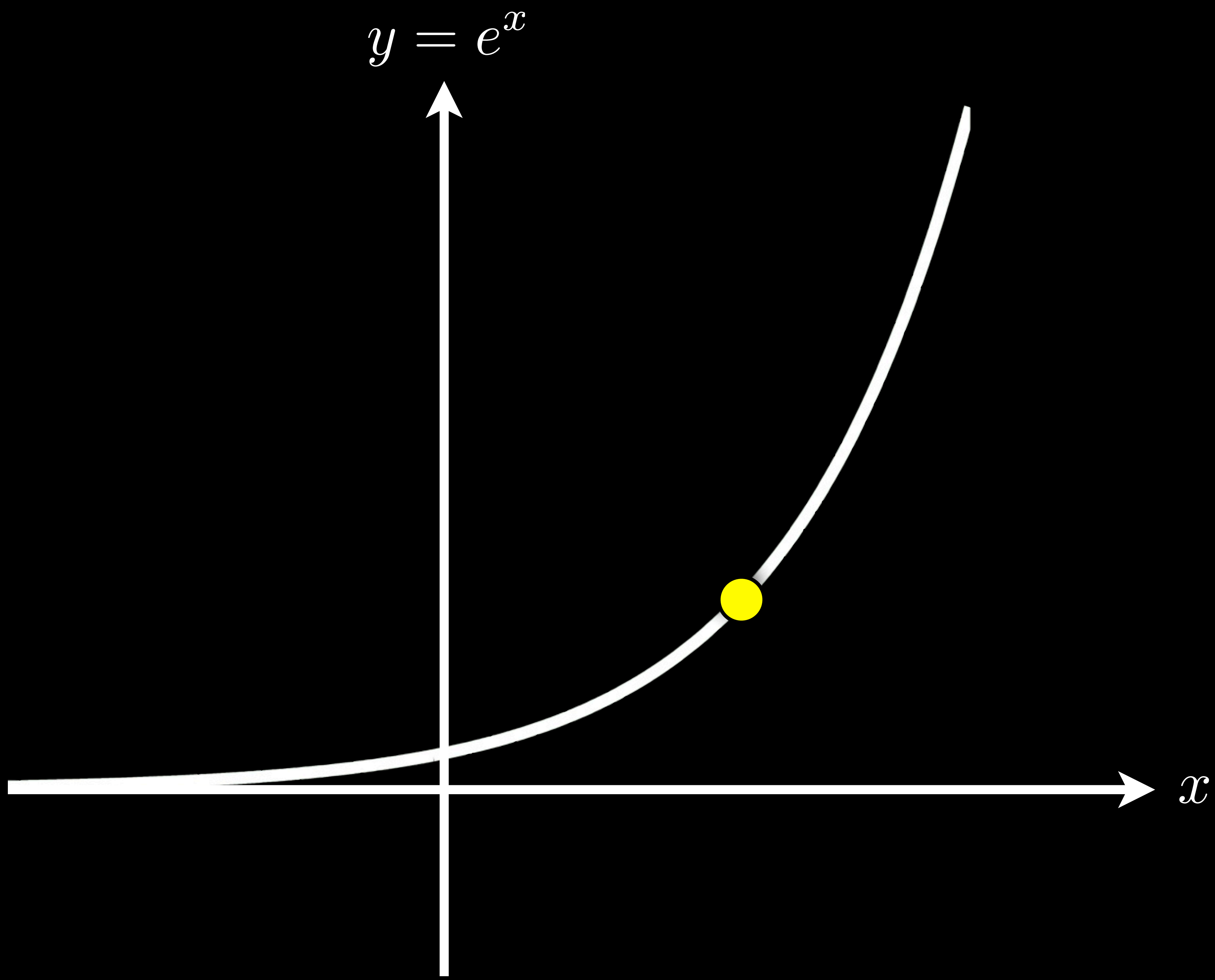
1D case

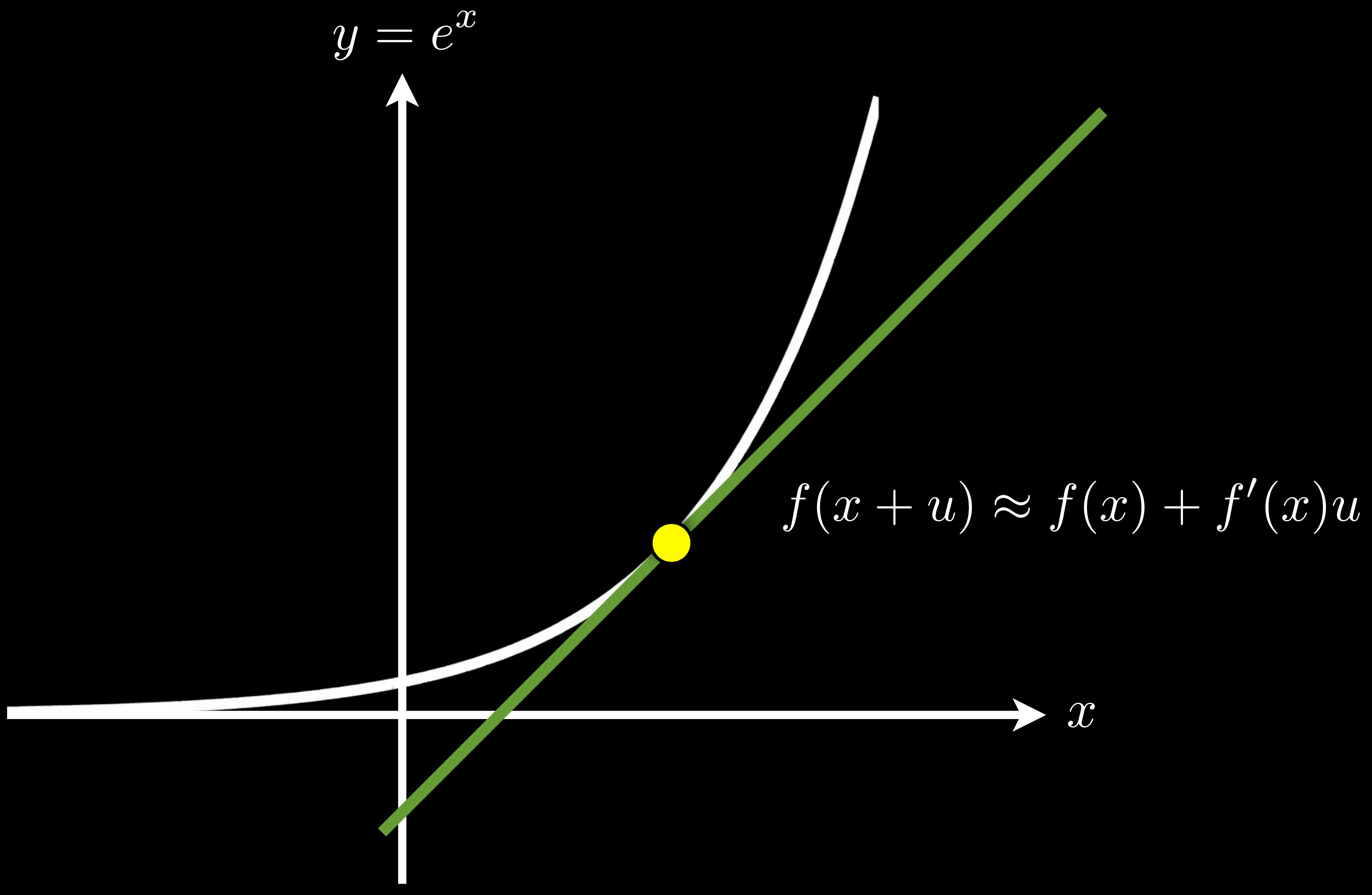
$$f(x+u) \approx f(x) + f'(x)u \cancel{+ \frac{1}{2} f''(x)u^2 + \text{h.o.t.}}$$

first-order approximation



$y = e^x$
x


$y = e^x$
x



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

2D case

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

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

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

2D case

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

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

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

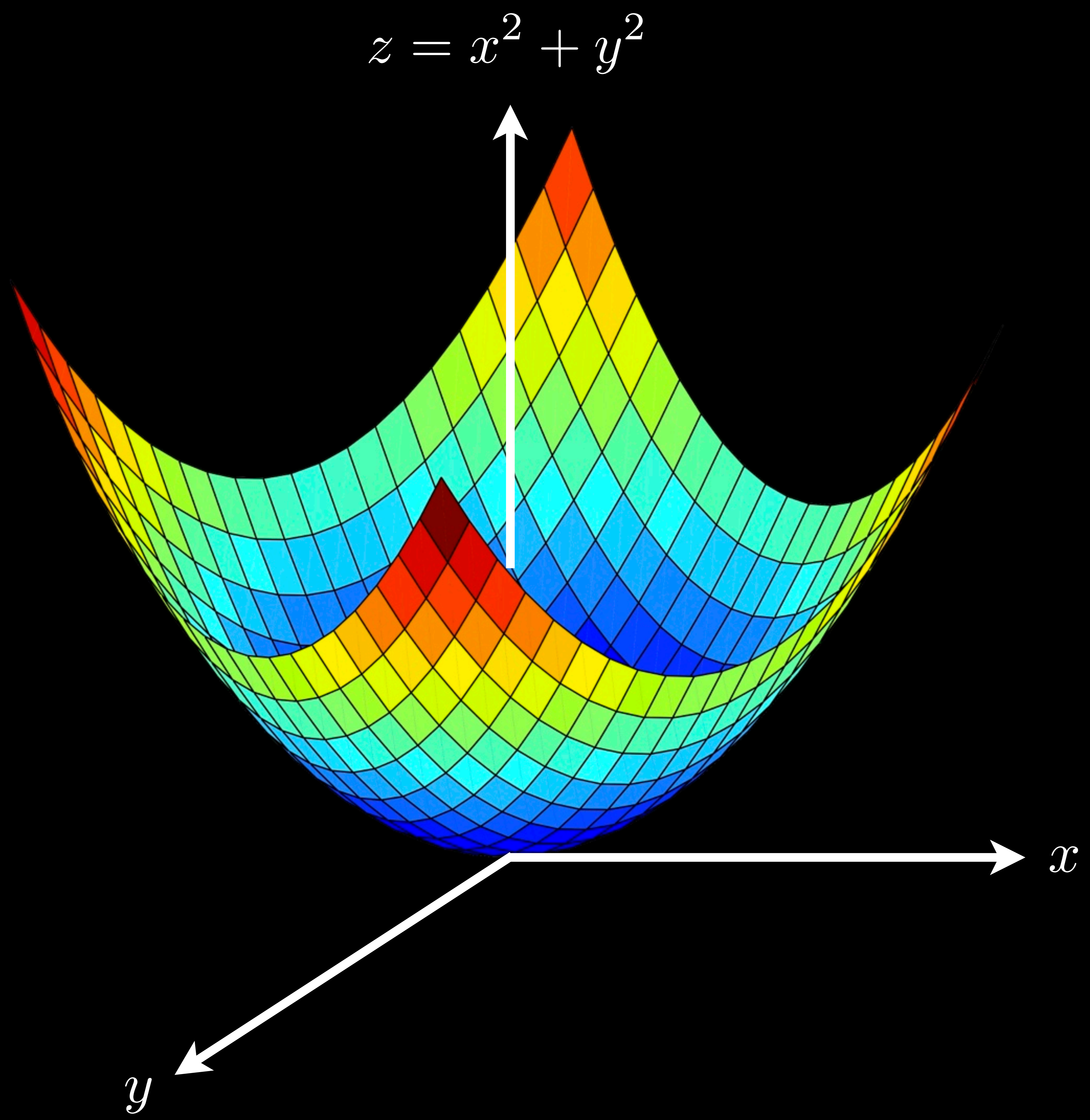
2D case

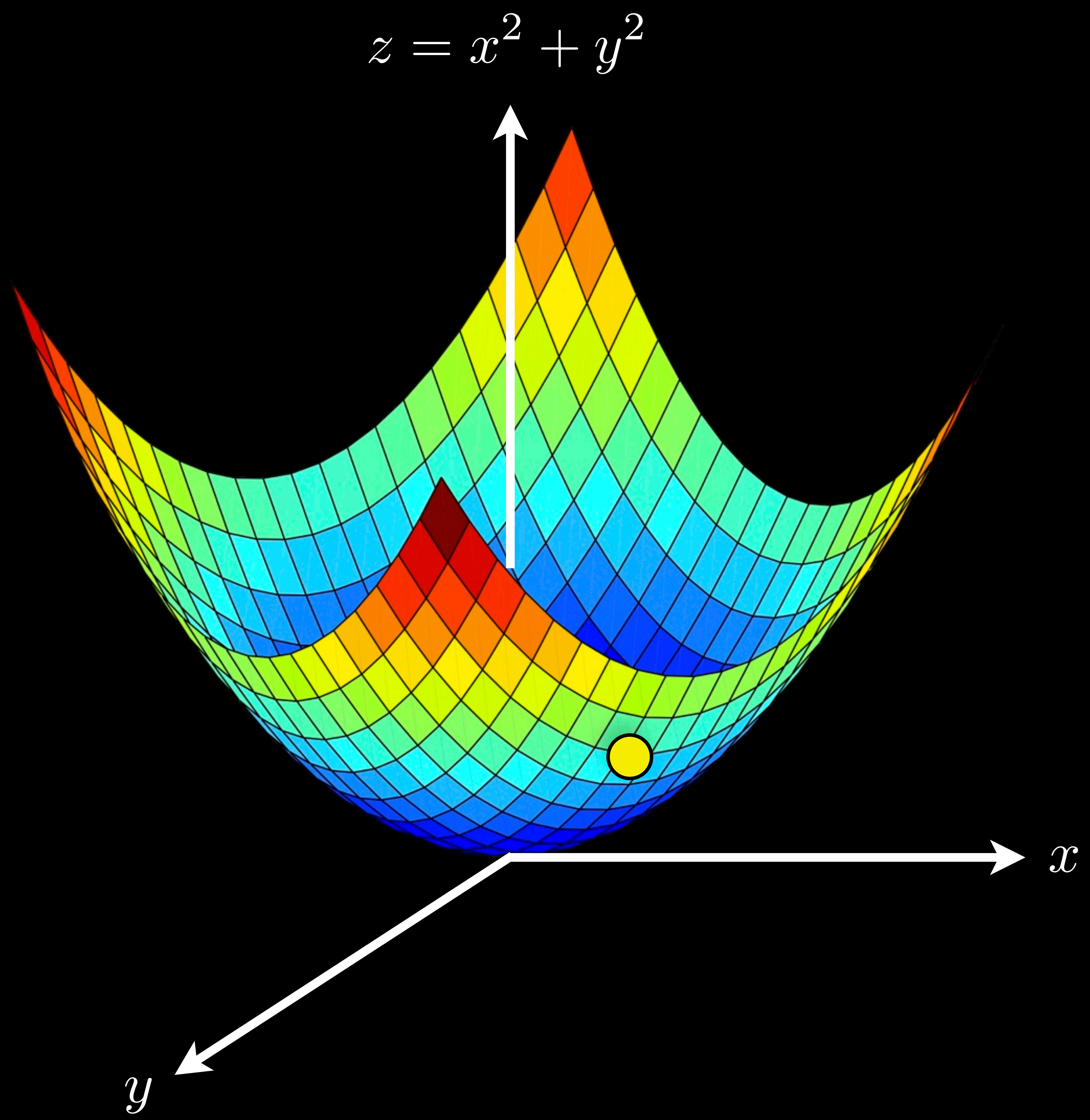
$$f(x+u, y+v) \approx f(x,y) + f_x(x,y)u + f_y(x,y)v$$

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

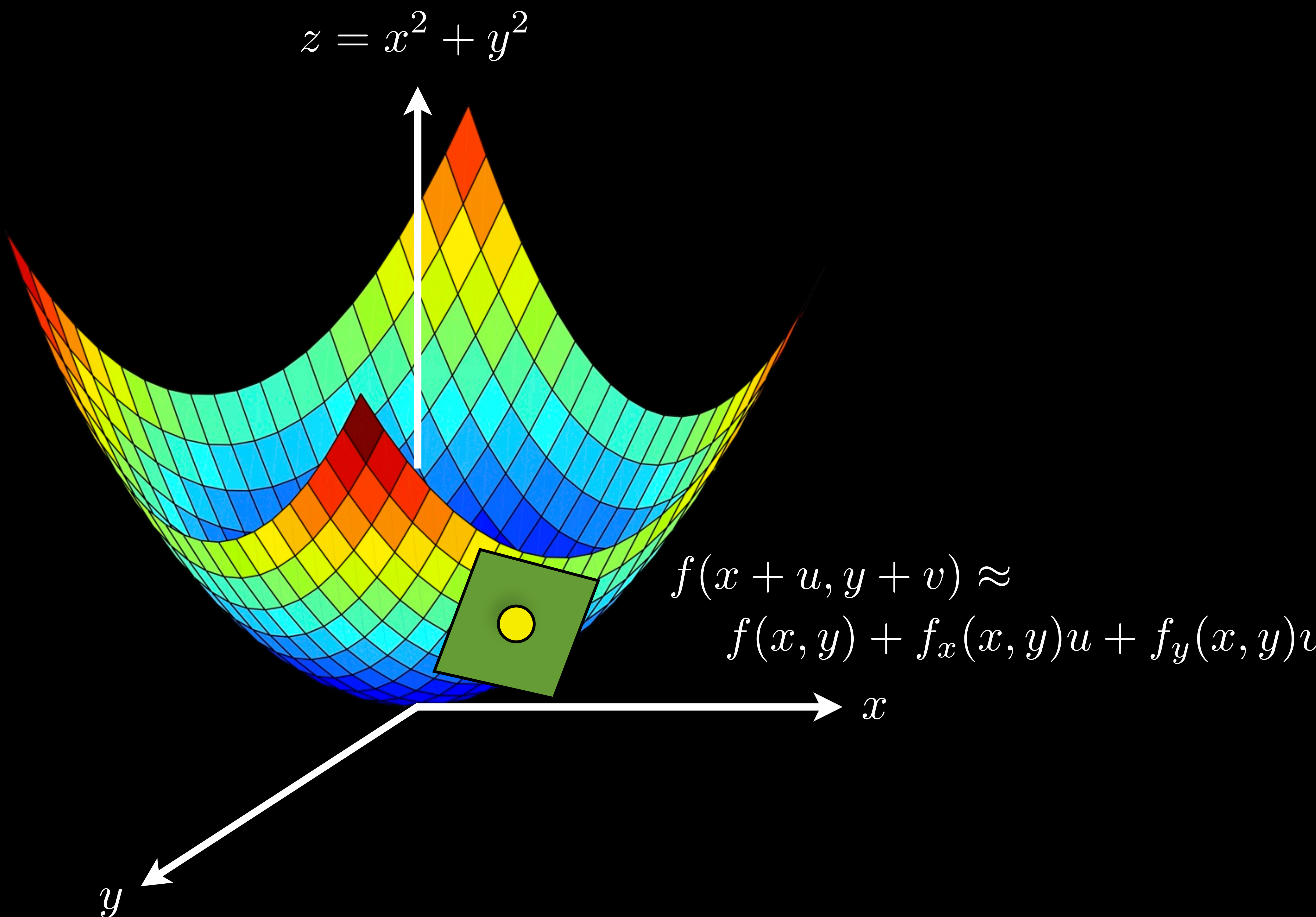
$$\cancel{+ \text{h.o.t.}}$$

first-order approximation


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



$z = x^2 + y^2$
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

Review: Taylor series over

Derivation

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

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

Derivation

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

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

How can we expand this term?

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

where

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

Review: Linear Algebra

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

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

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

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 $\mathbf{B} = \mathbf{B}^\top$.

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

$$\text{e.g.,} \quad \begin{pmatrix} 1 & 7 & 3 \\ 7 & 4 & -5 \\ 3 & -5 & 6 \end{pmatrix}$$

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:

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:

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

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:

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

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

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

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

Review: Linear Algebra over

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

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

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

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

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**

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

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

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

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

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

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

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

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

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

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

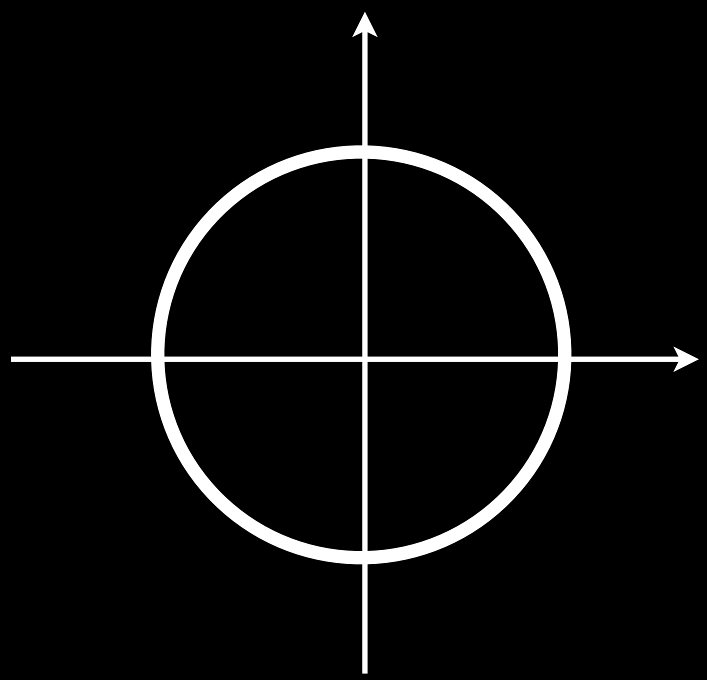
$$\mathbf{R} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \mathbf{R}^\top$$

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

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

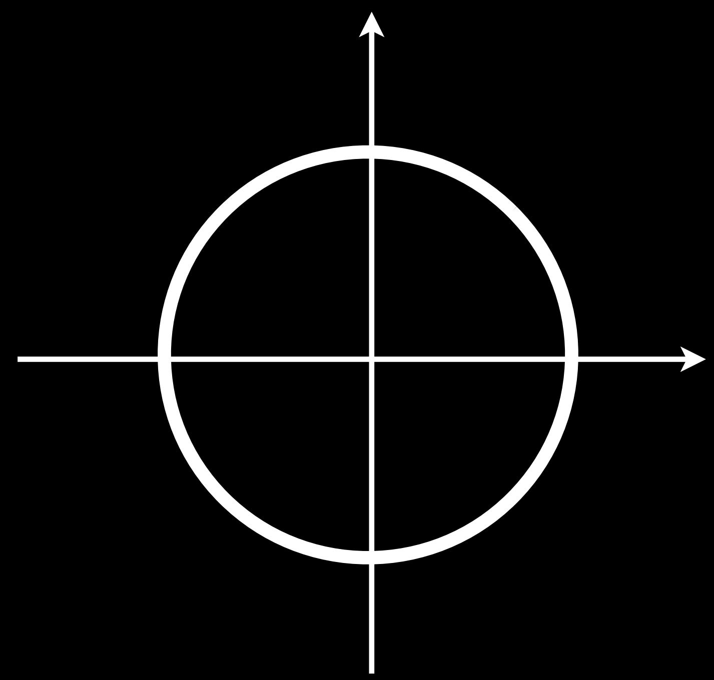
$$\mathbf{x}^\top \mathbf{R} \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \mathbf{R}^\top \mathbf{x}$$

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



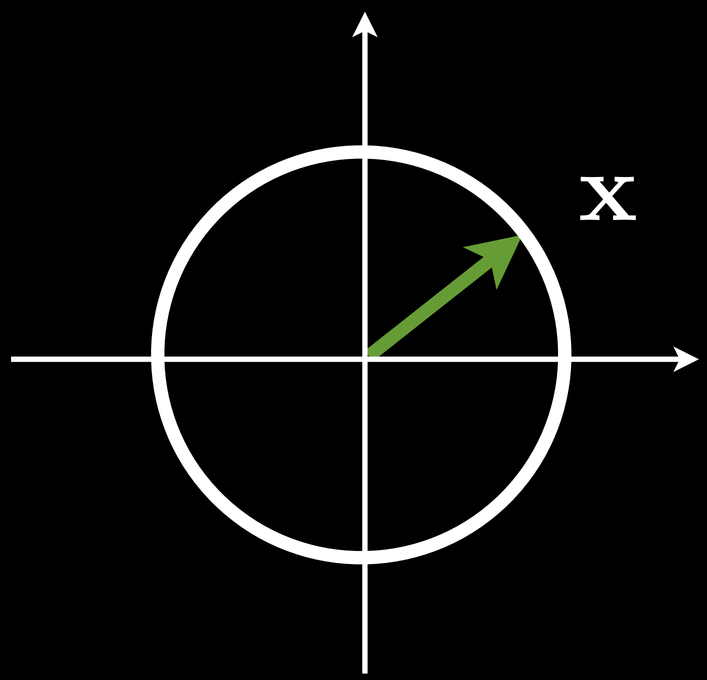
unit circle

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



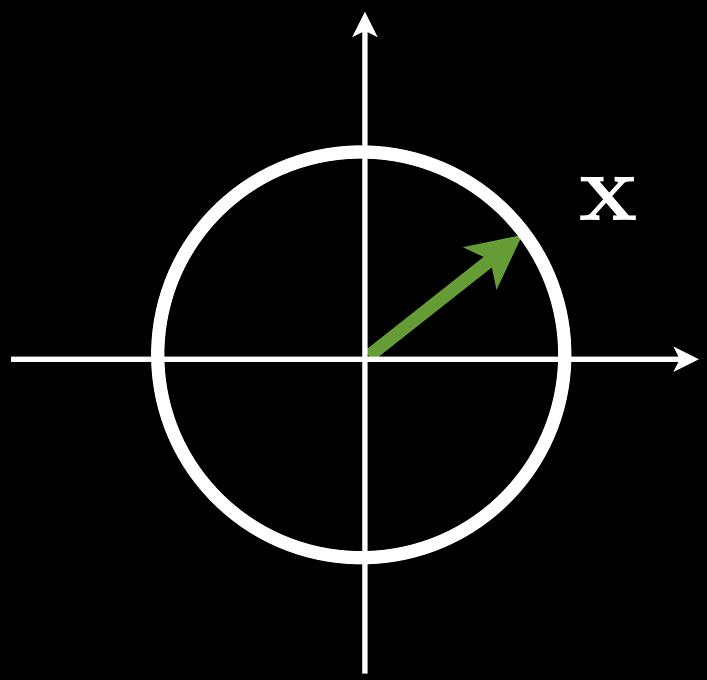
unit circle

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



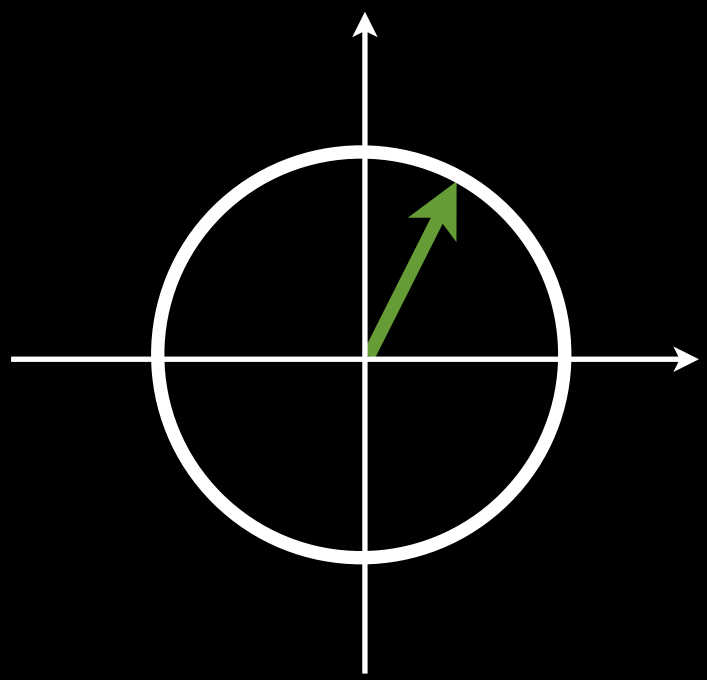
unit circle

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



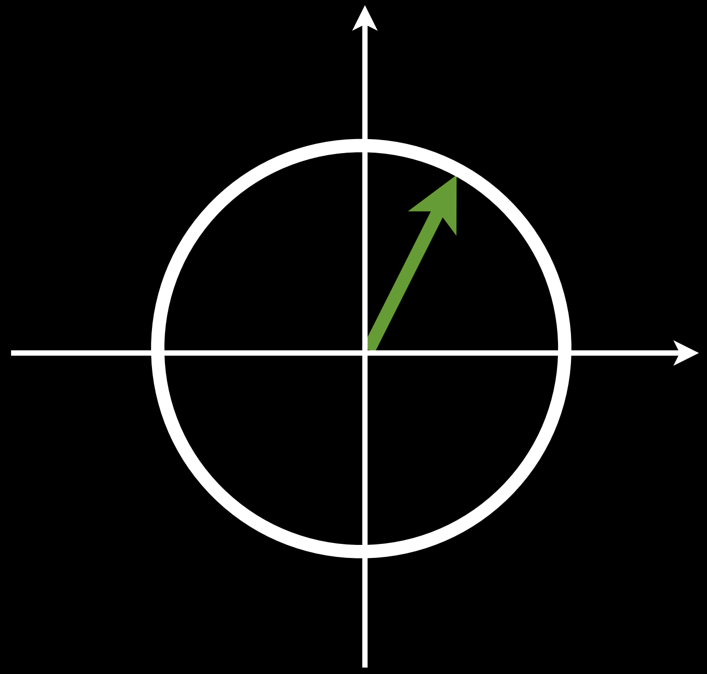
$$\mathbf{R}^\top \mathbf{x}$$

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

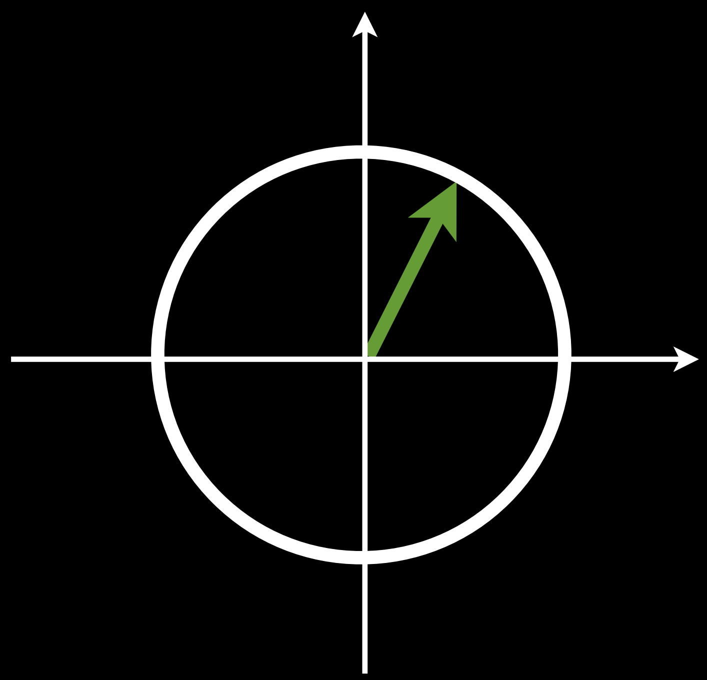


$\mathbf{R}^\top \mathbf{x}$

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

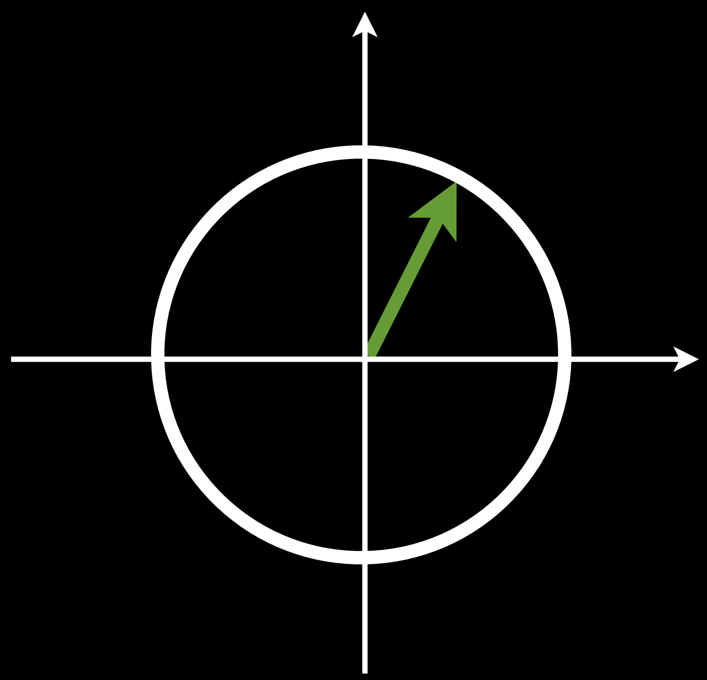


$$\mathbf{x}^\top \mathbf{R} \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \mathbf{R}^\top \mathbf{x}$$



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

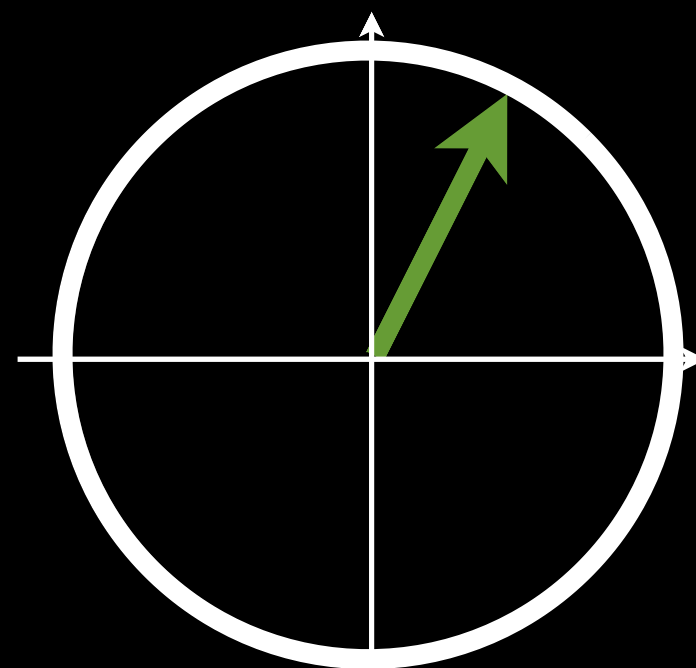
$$\mathbf{x}^\top \mathbf{R} \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \underline{\begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \mathbf{R}^\top \mathbf{x}}$$



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

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

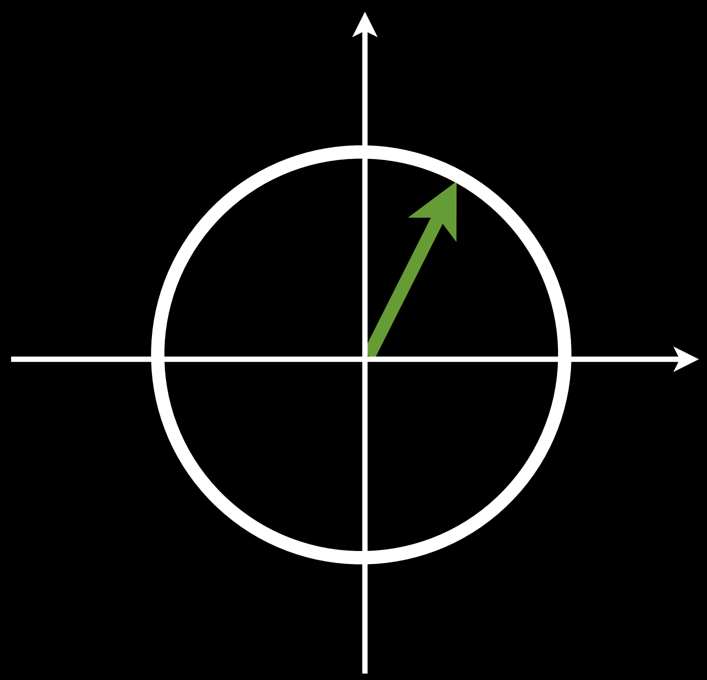
$$\mathbf{x}^\top \mathbf{R} \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \mathbf{R}^\top \mathbf{x}$$



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

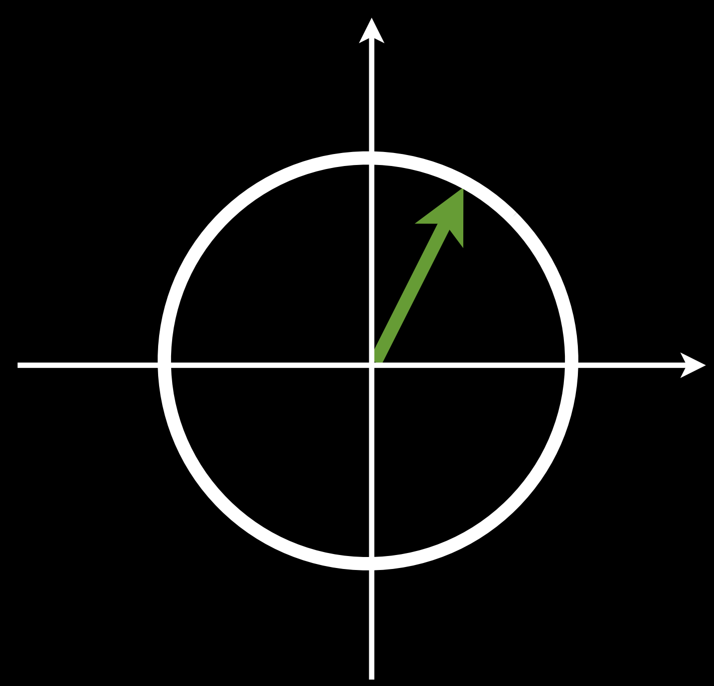
if $\lambda_1 = \lambda_2$ **and large**

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



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

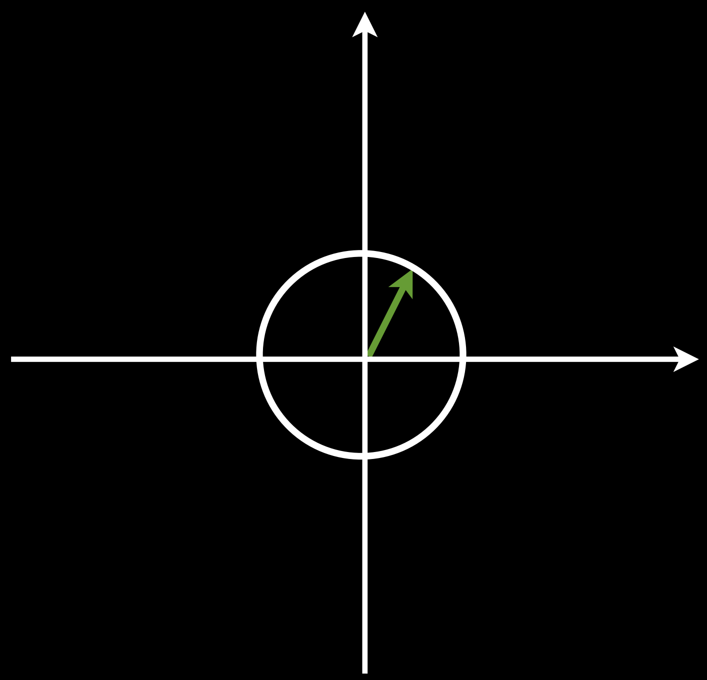
$$\mathbf{x}^\top \mathbf{R} \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \mathbf{R}^\top \mathbf{x}$$



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

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

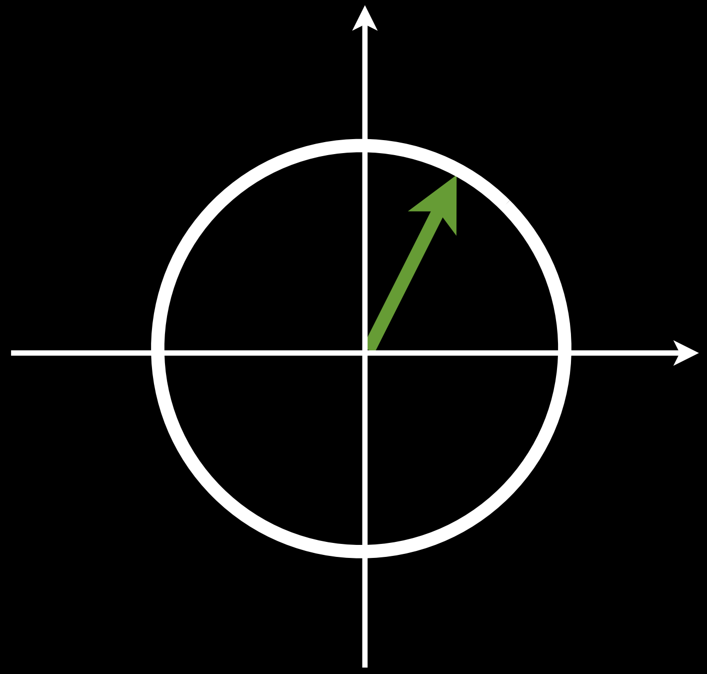
$$\mathbf{x}^\top \mathbf{R} \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \mathbf{R}^\top \mathbf{x}$$



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

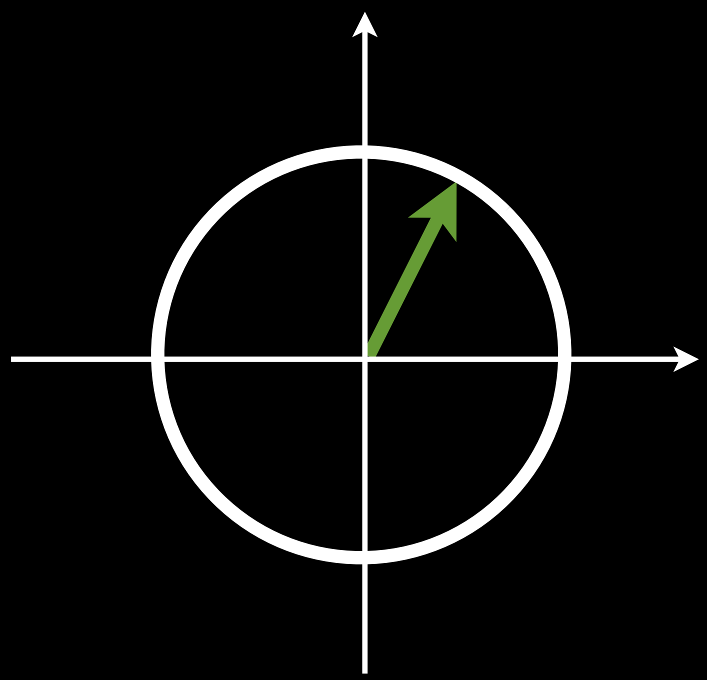
if $\lambda_1 = \lambda_2$ **and small**

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



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

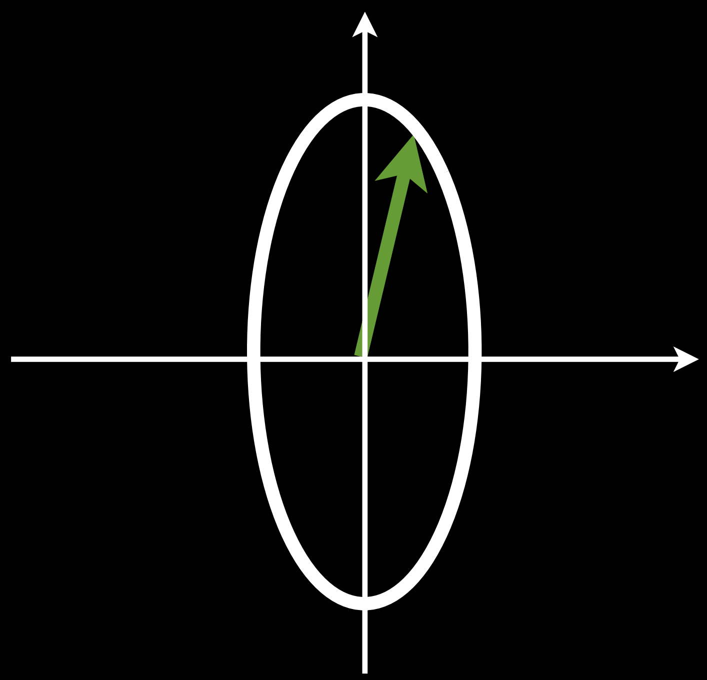
$$\mathbf{x}^\top \mathbf{R} \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \mathbf{R}^\top \mathbf{x}$$



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

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

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



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

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

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

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

Have we seen this before?

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

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

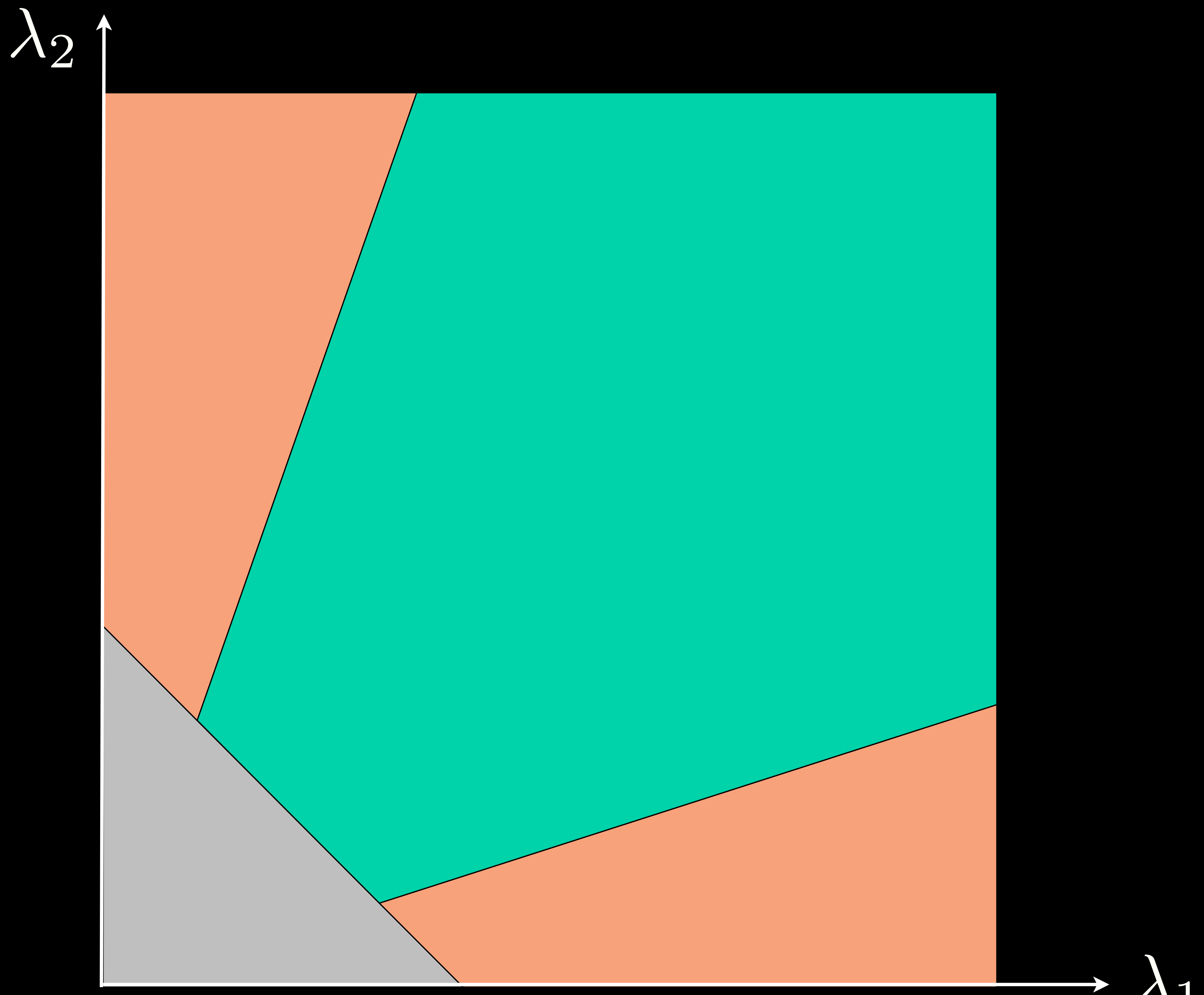


Image point classification using eigenvalues of M

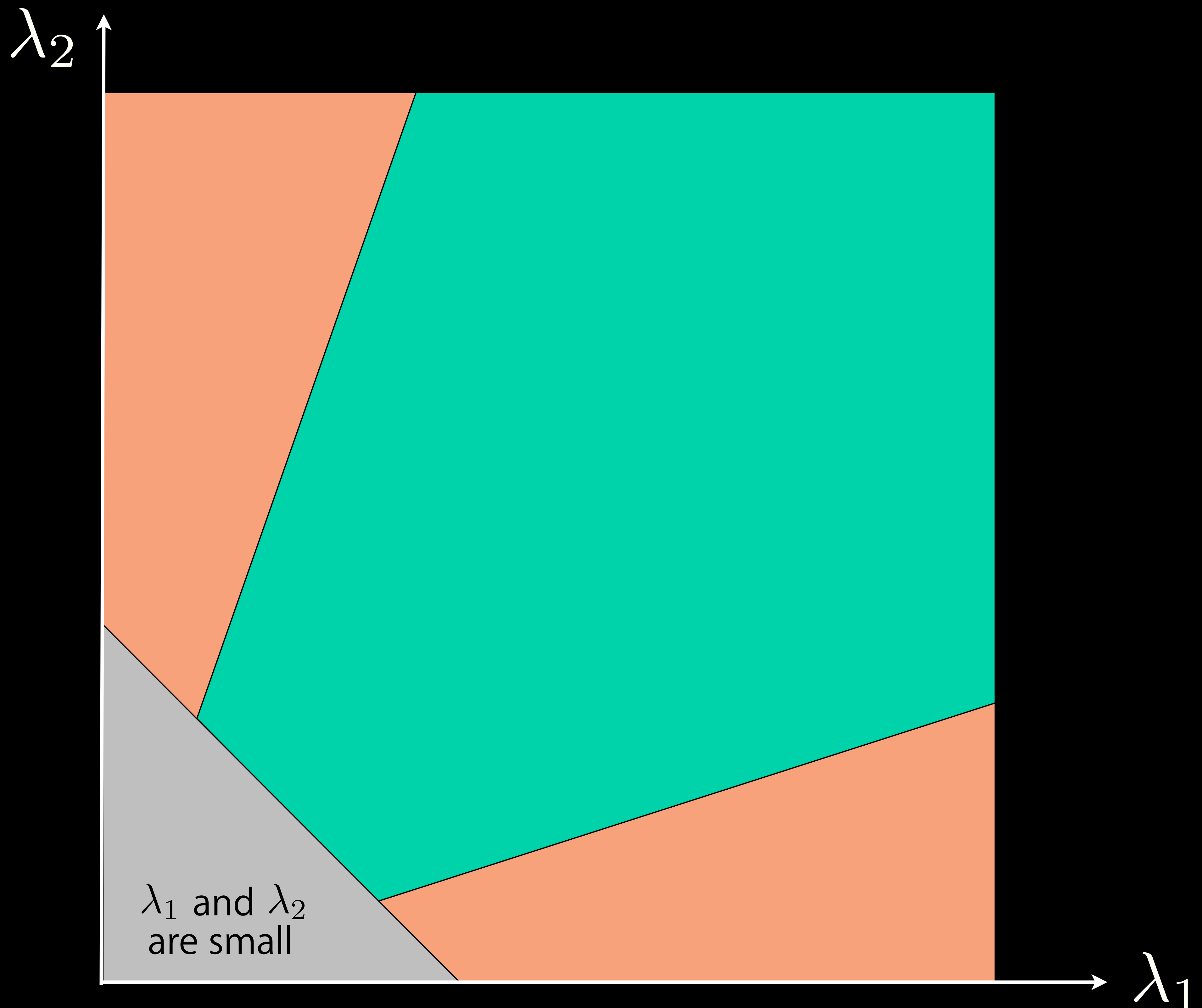


Image point classification using eigenvalues of M

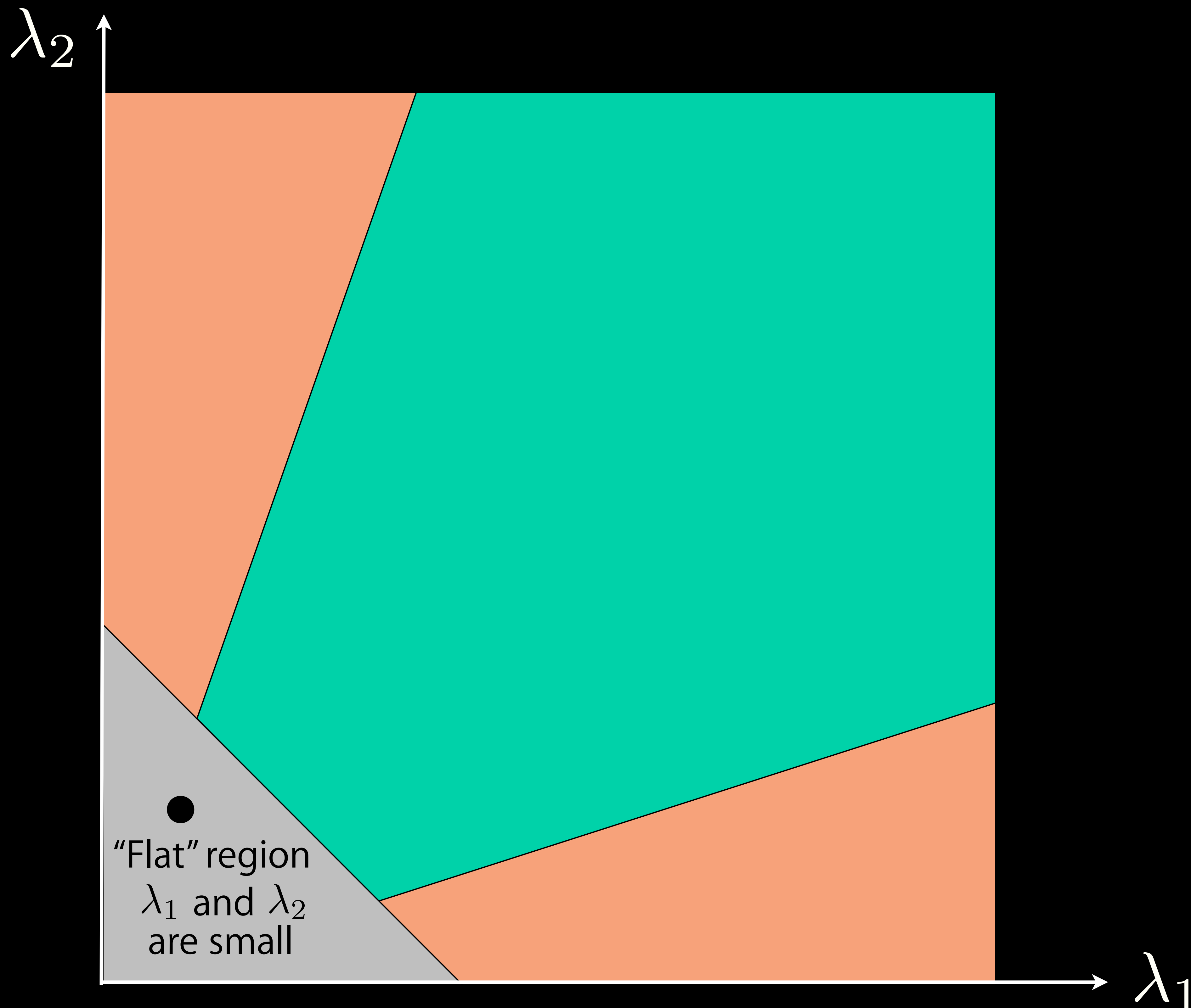


Image point classification using eigenvalues of M

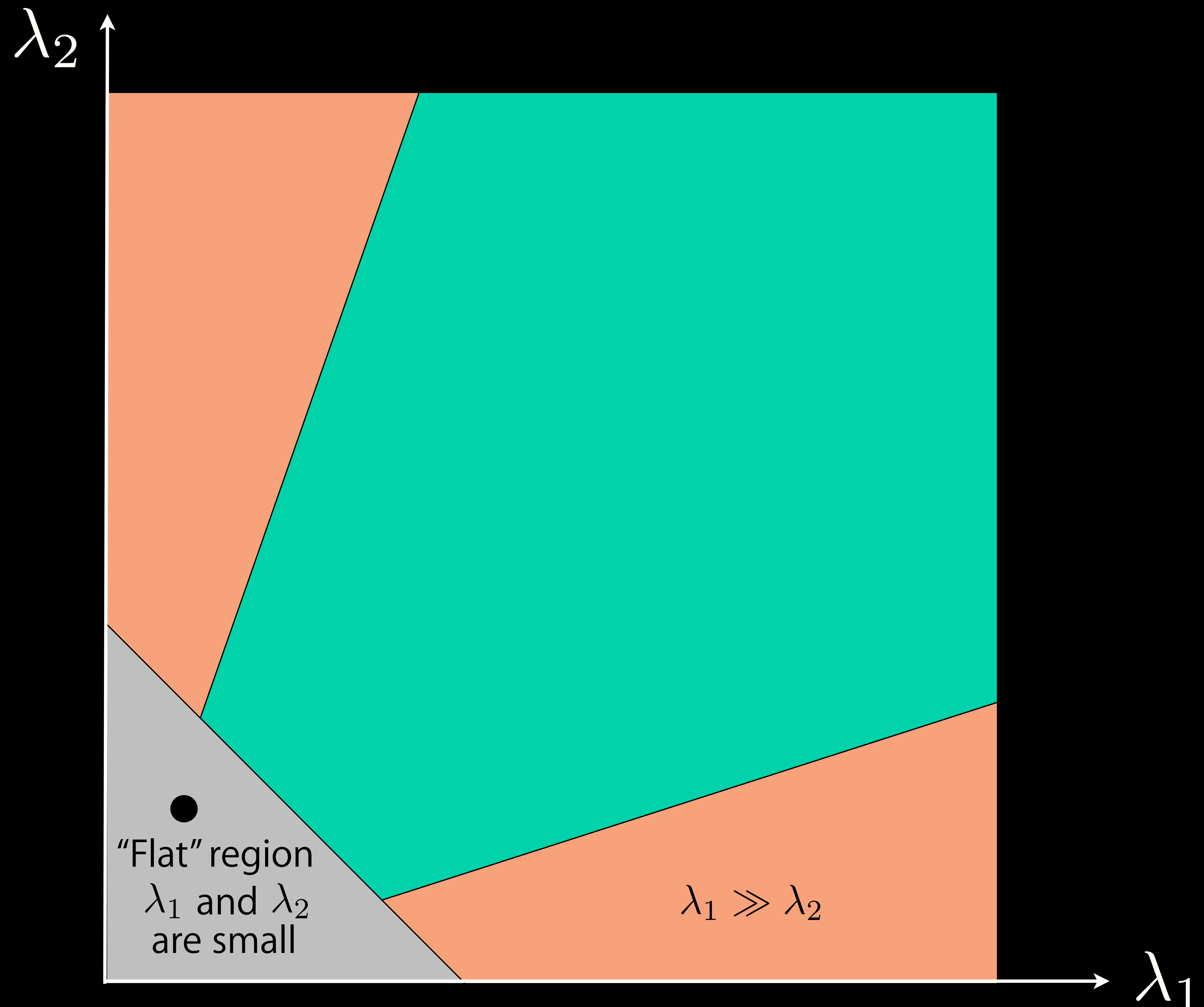


Image point classification using eigenvalues of M

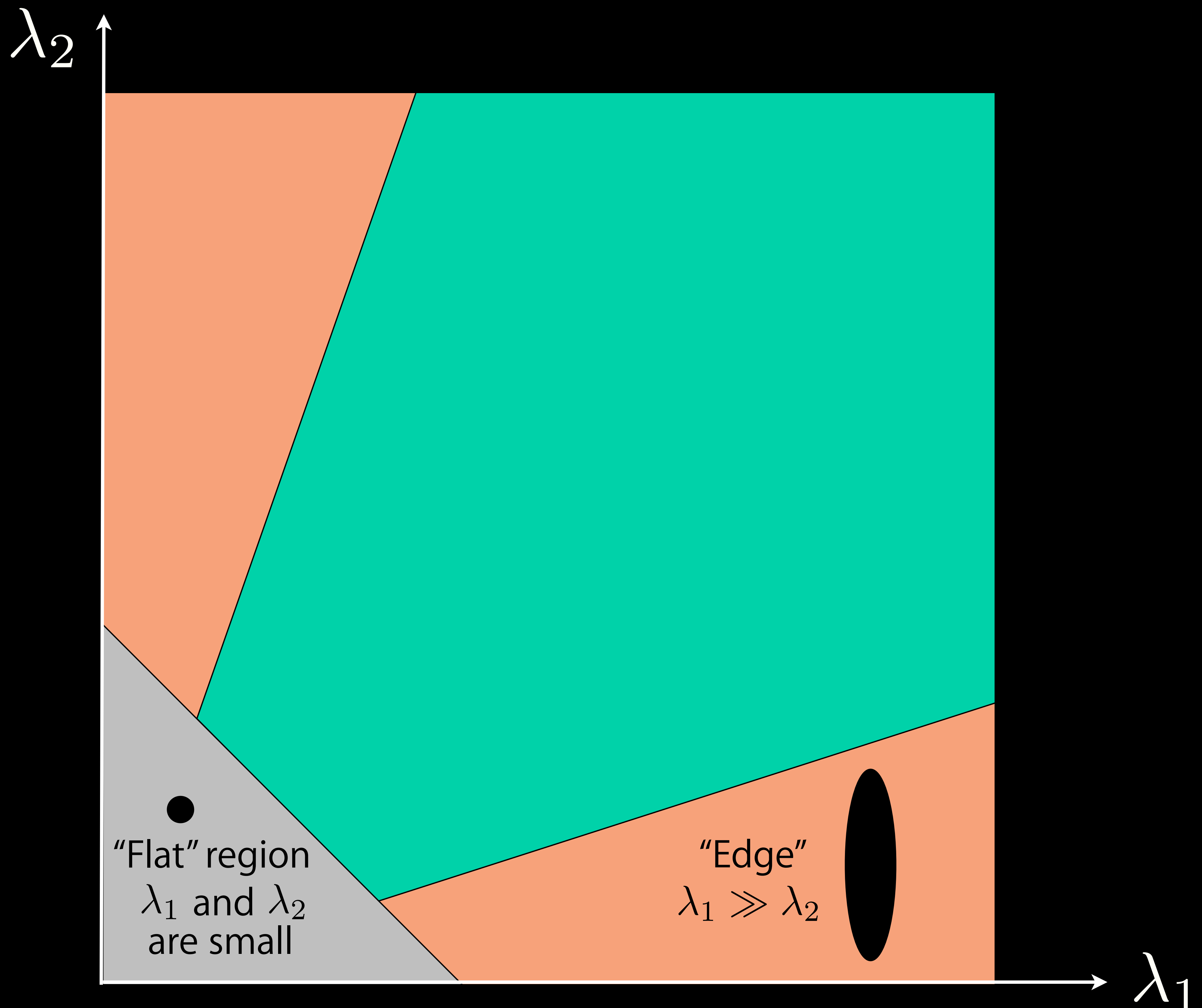


Image point classification using eigenvalues of M

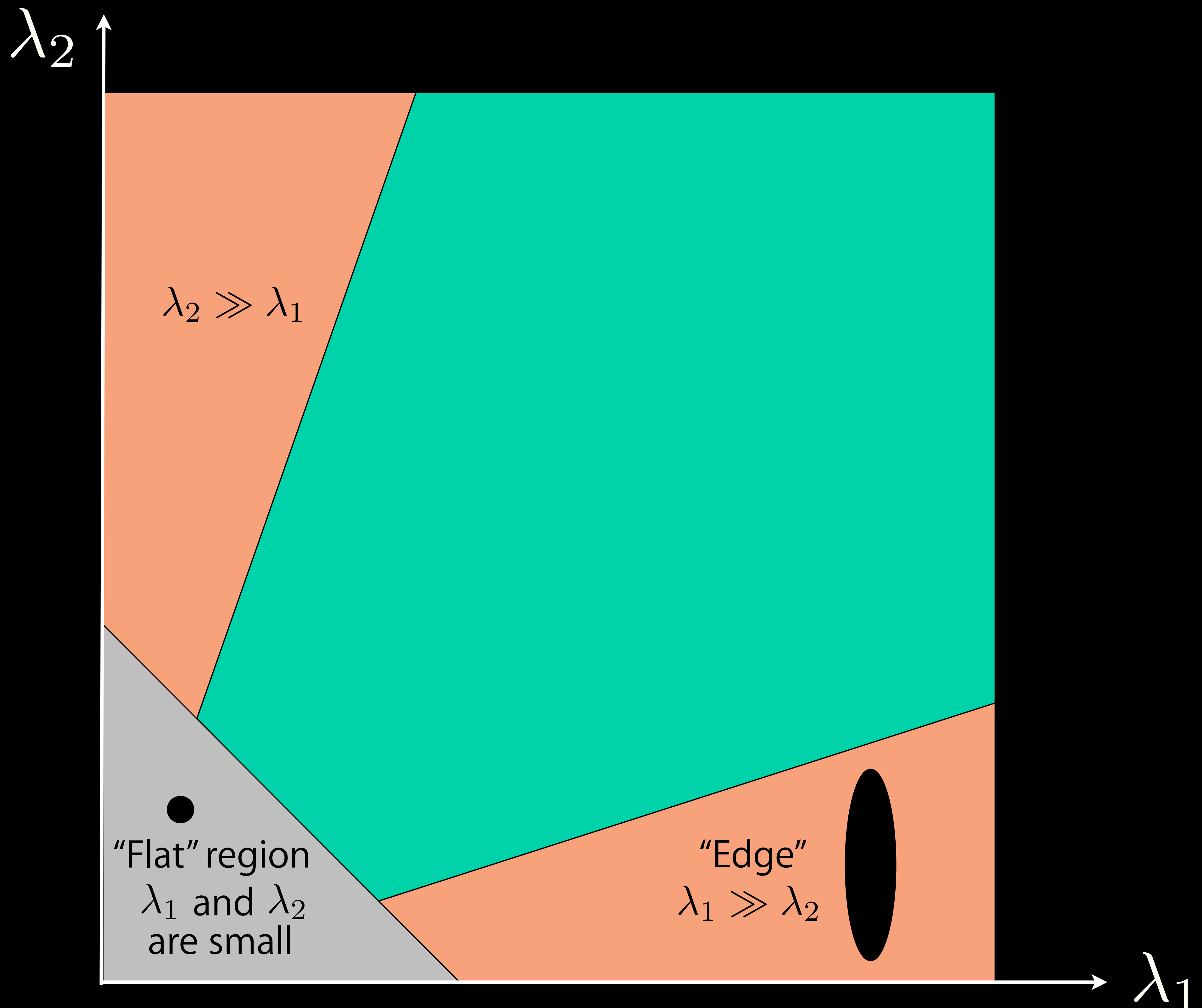


Image point classification using eigenvalues of M

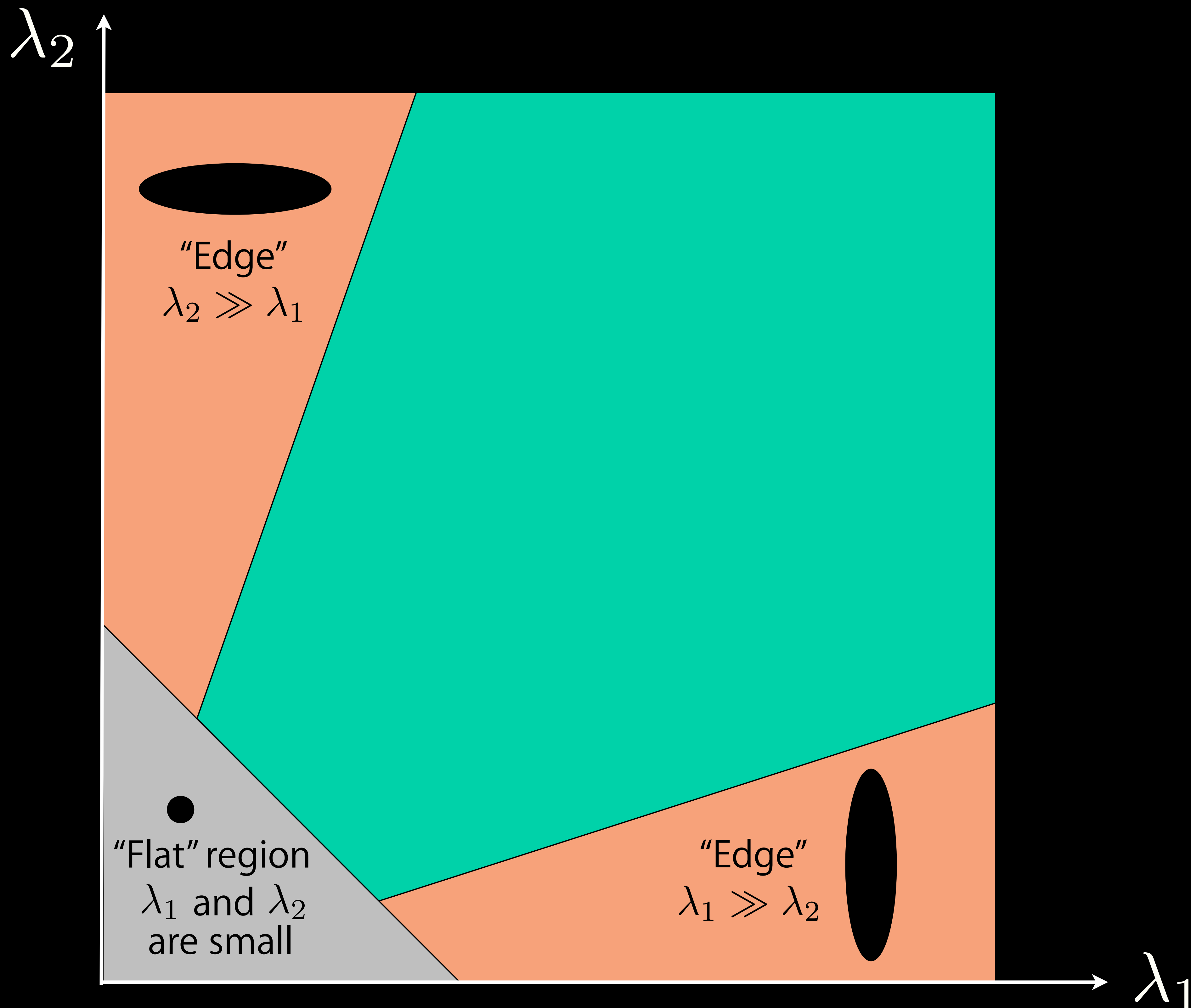


Image point classification using eigenvalues of M

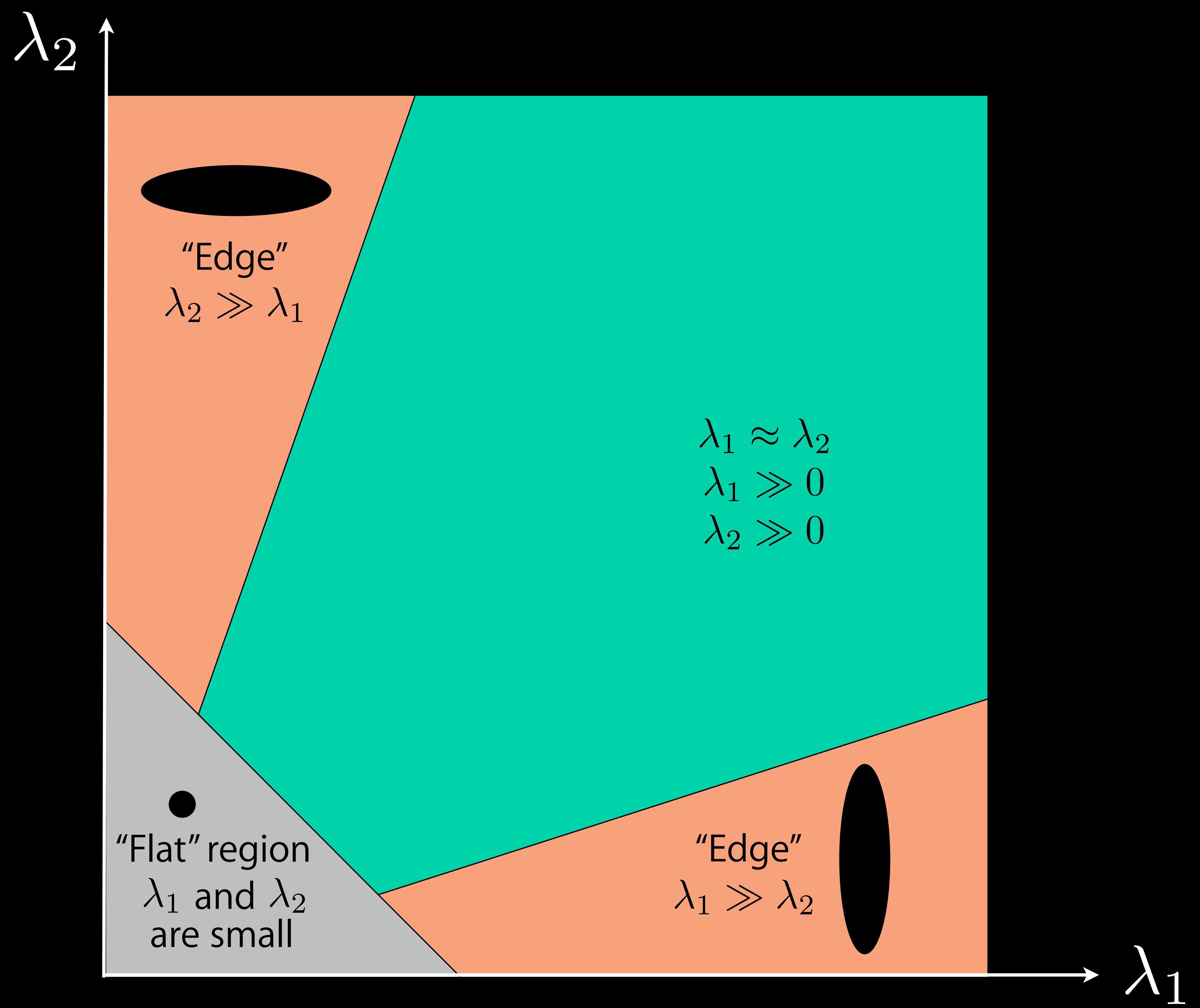
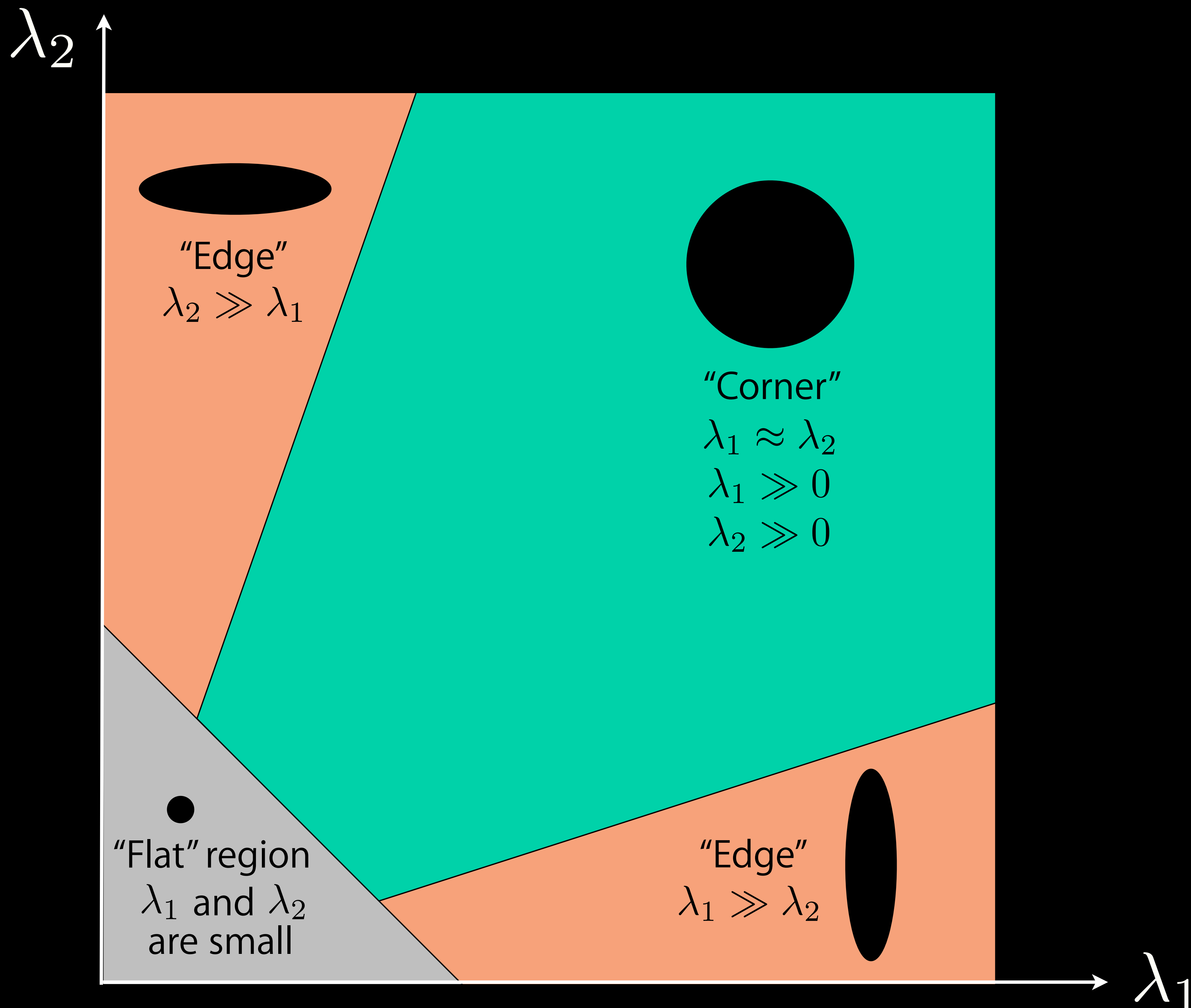


Image point classification using eigenvalues of $\mathbf{M}$



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

Harris Corner Response Function

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

k is empirically set constant: 0.04 - 0.06

implicit way of using eigenvalues

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

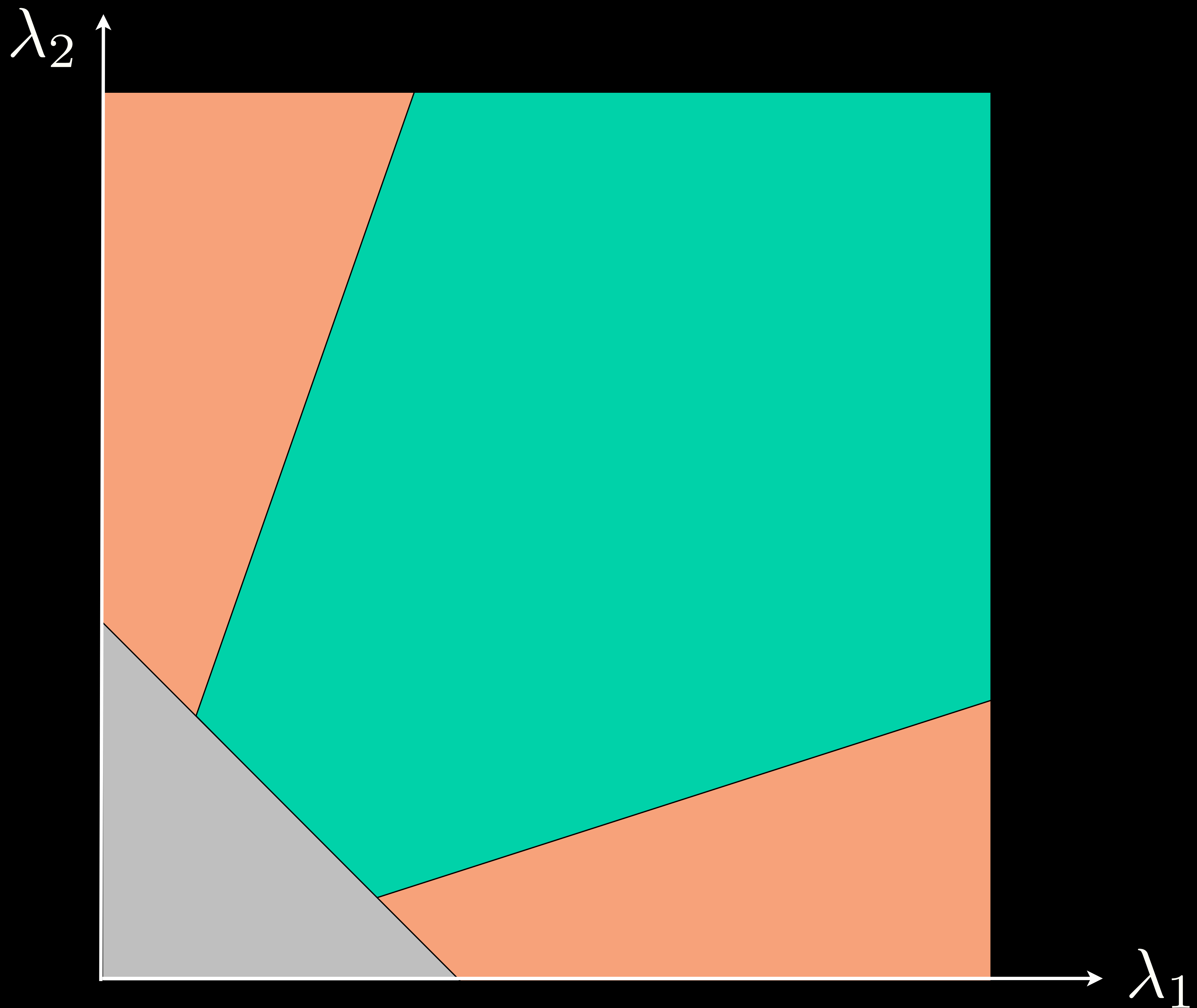


Image point classification using r

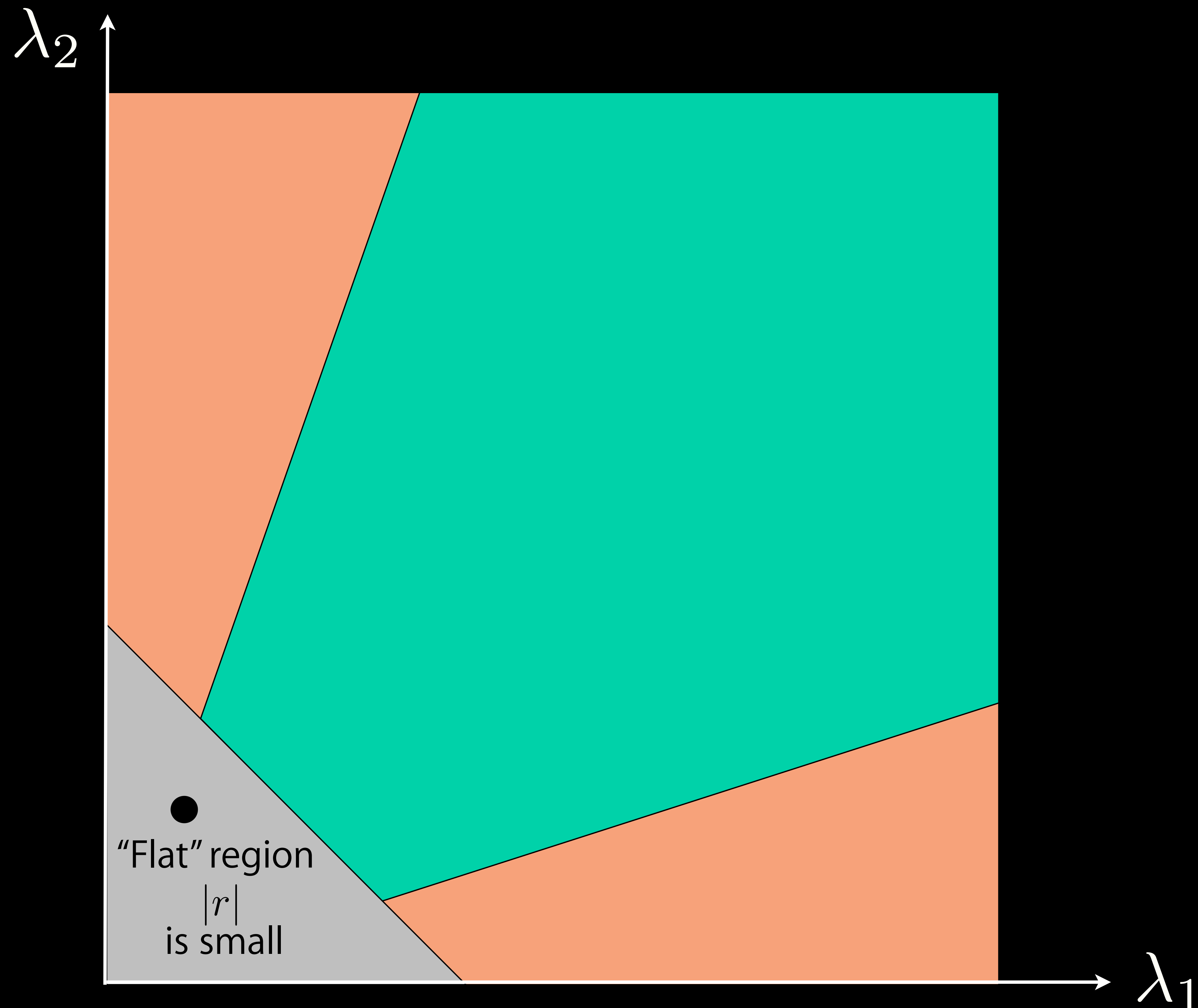


Image point classification using r

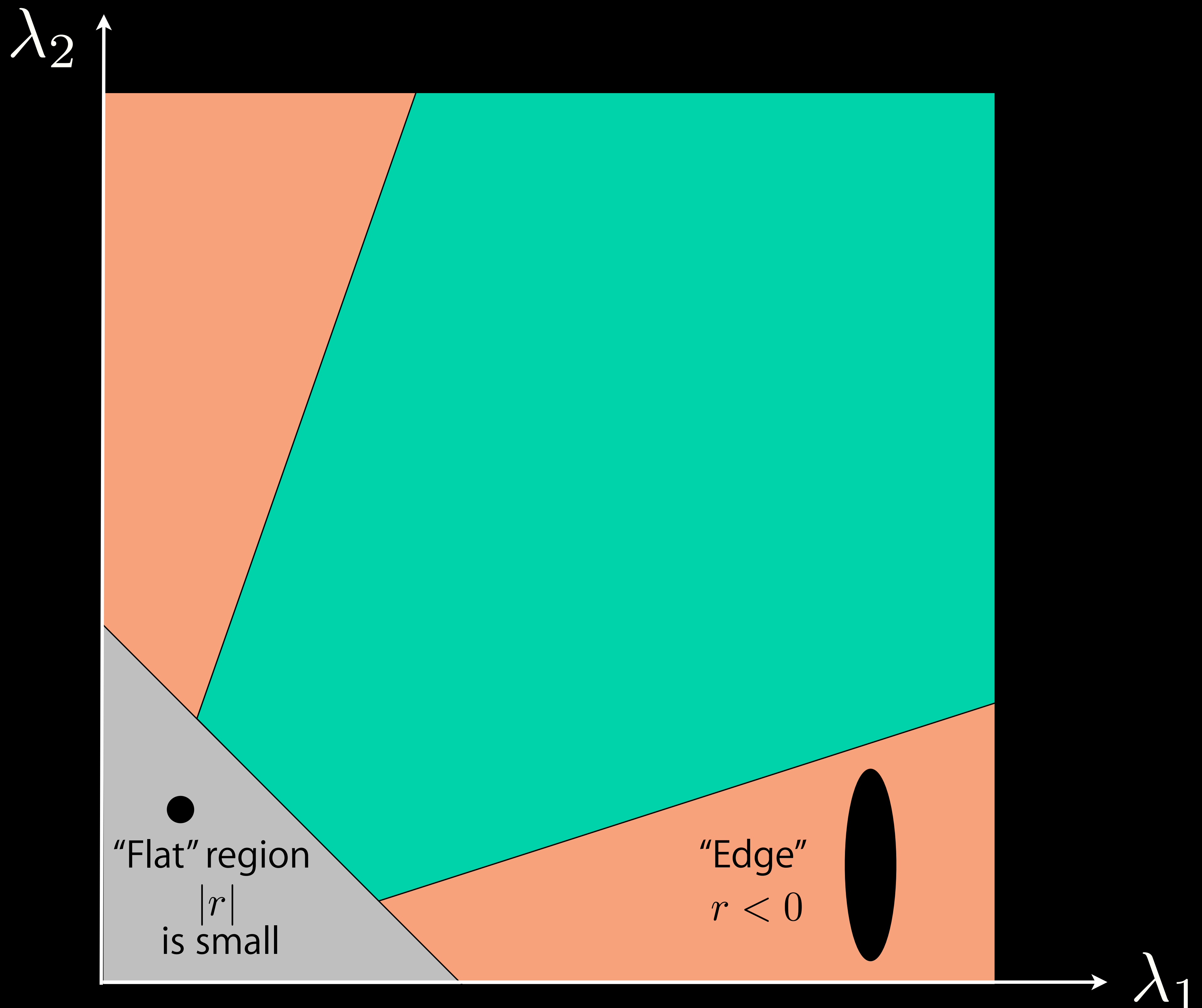


Image point classification using r

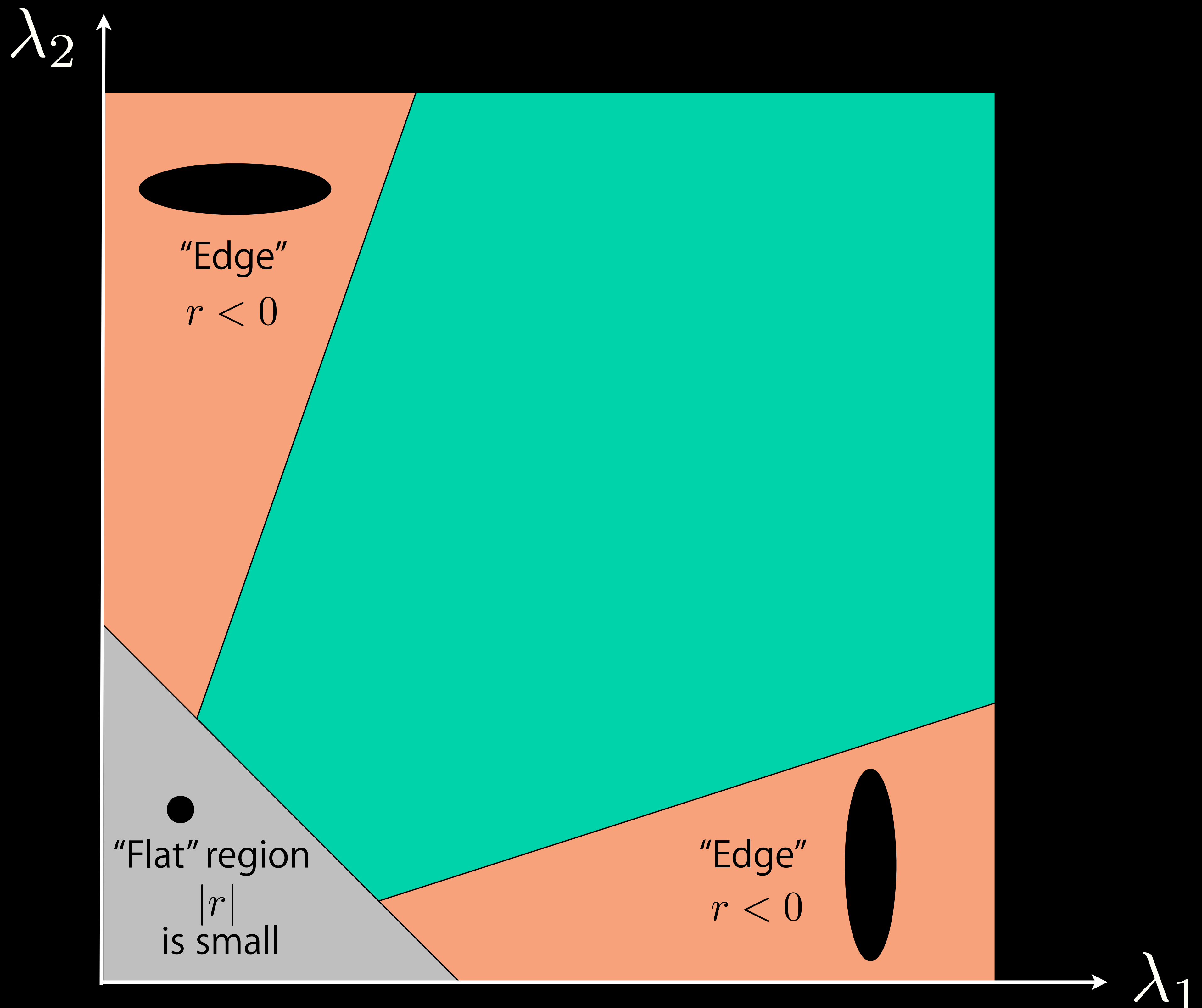
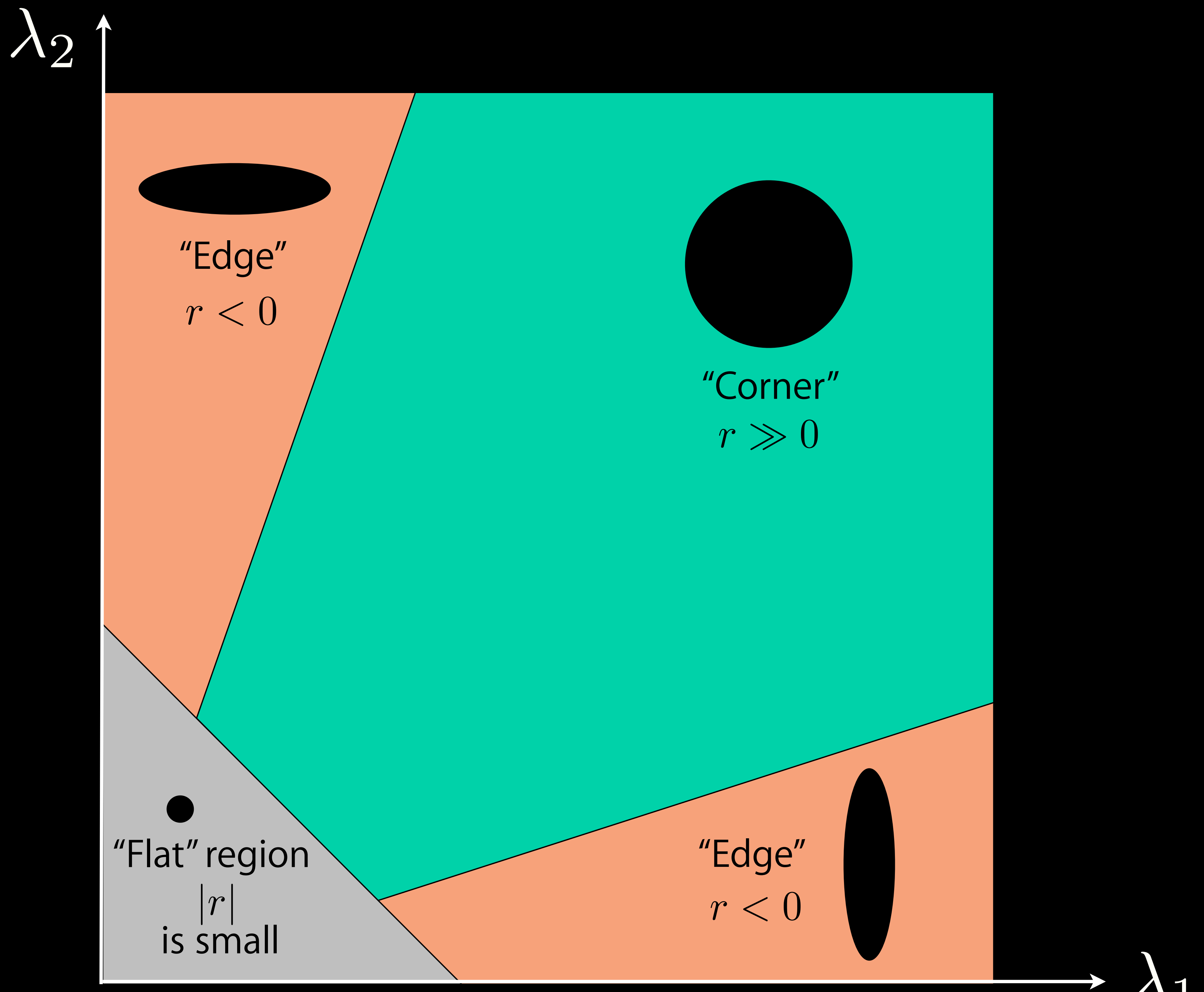


Image point classification using r



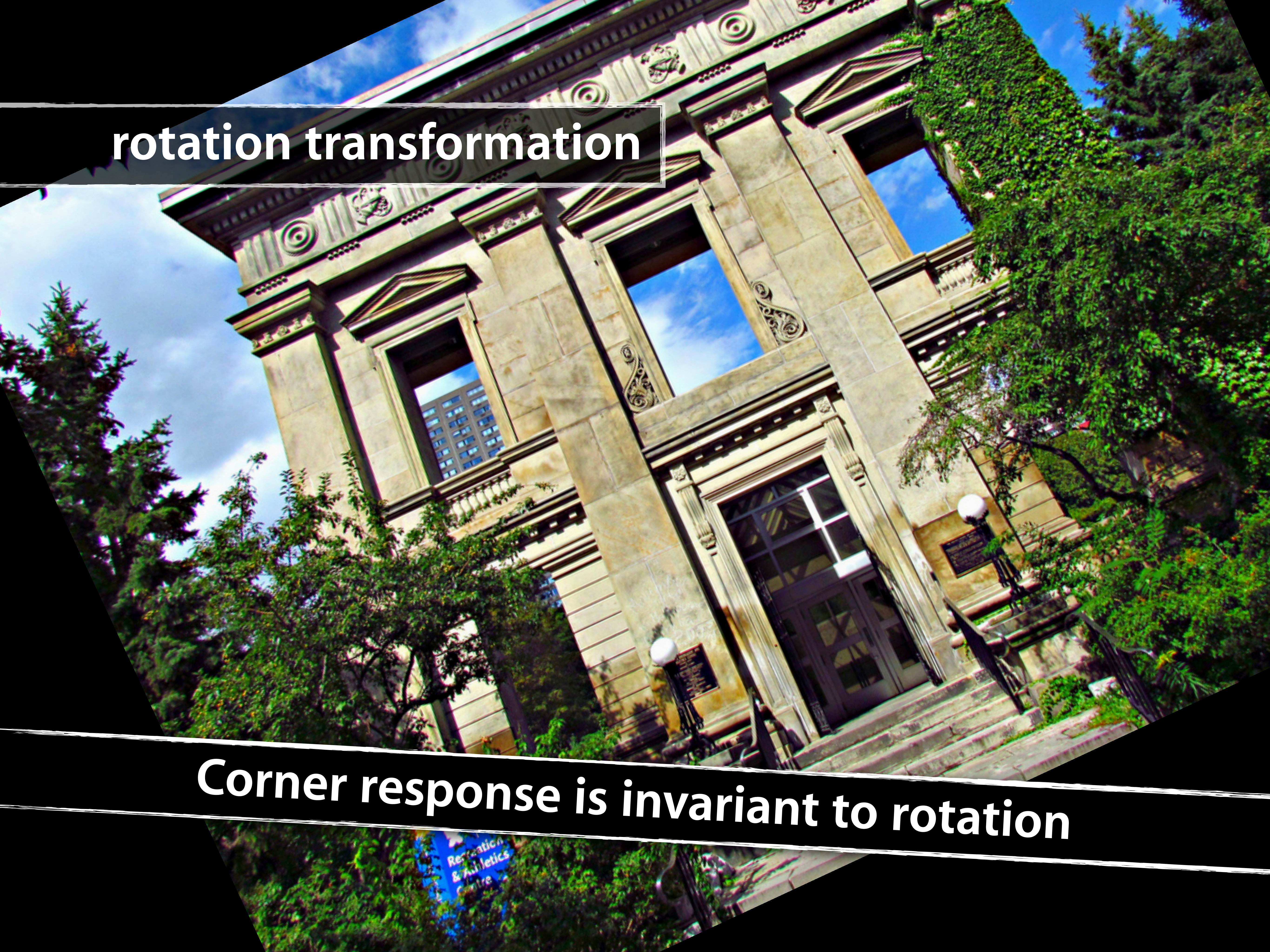
Harris Corner
Properties

Harris Corner Properties

Rotation invariant

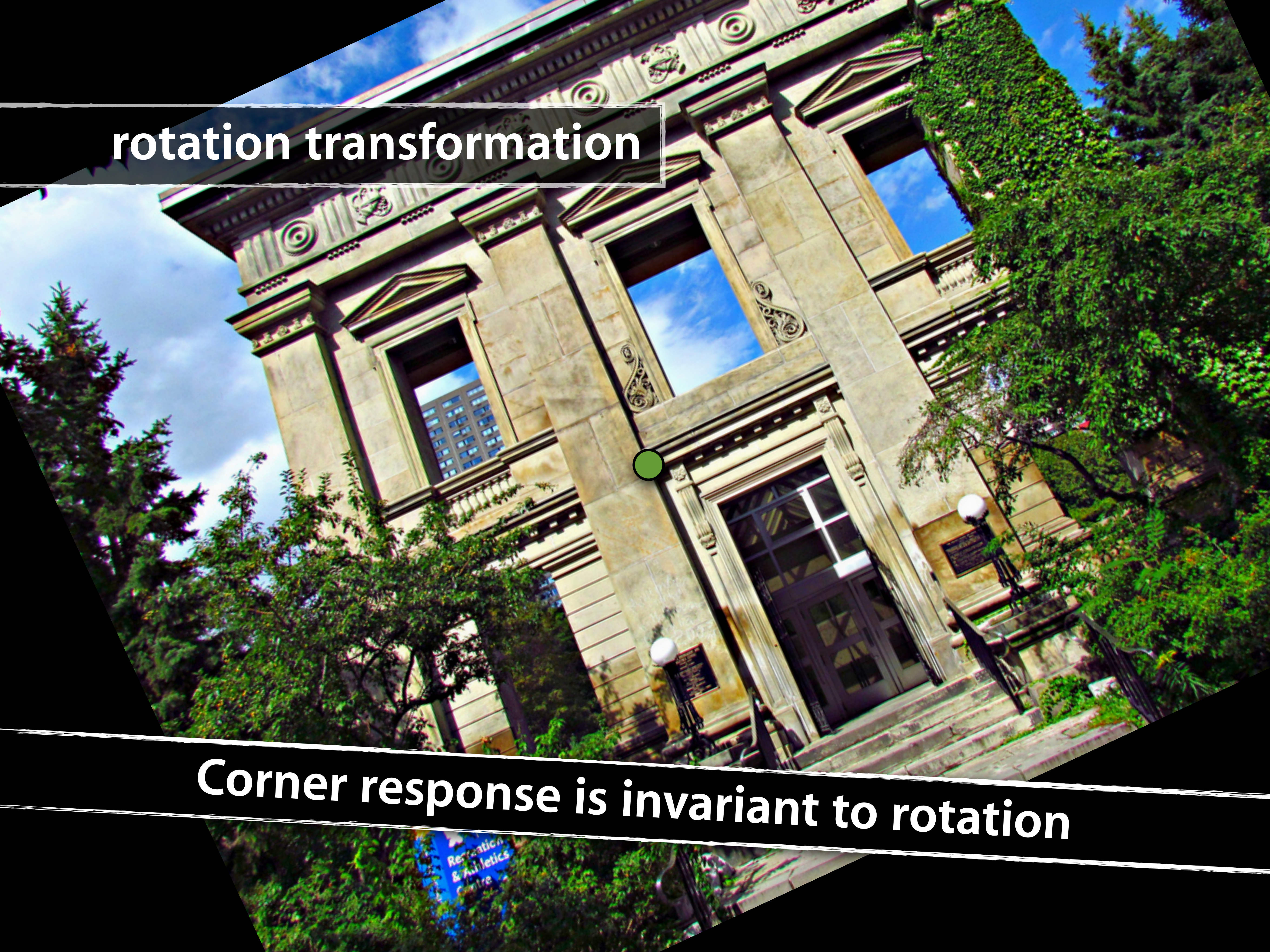

input image

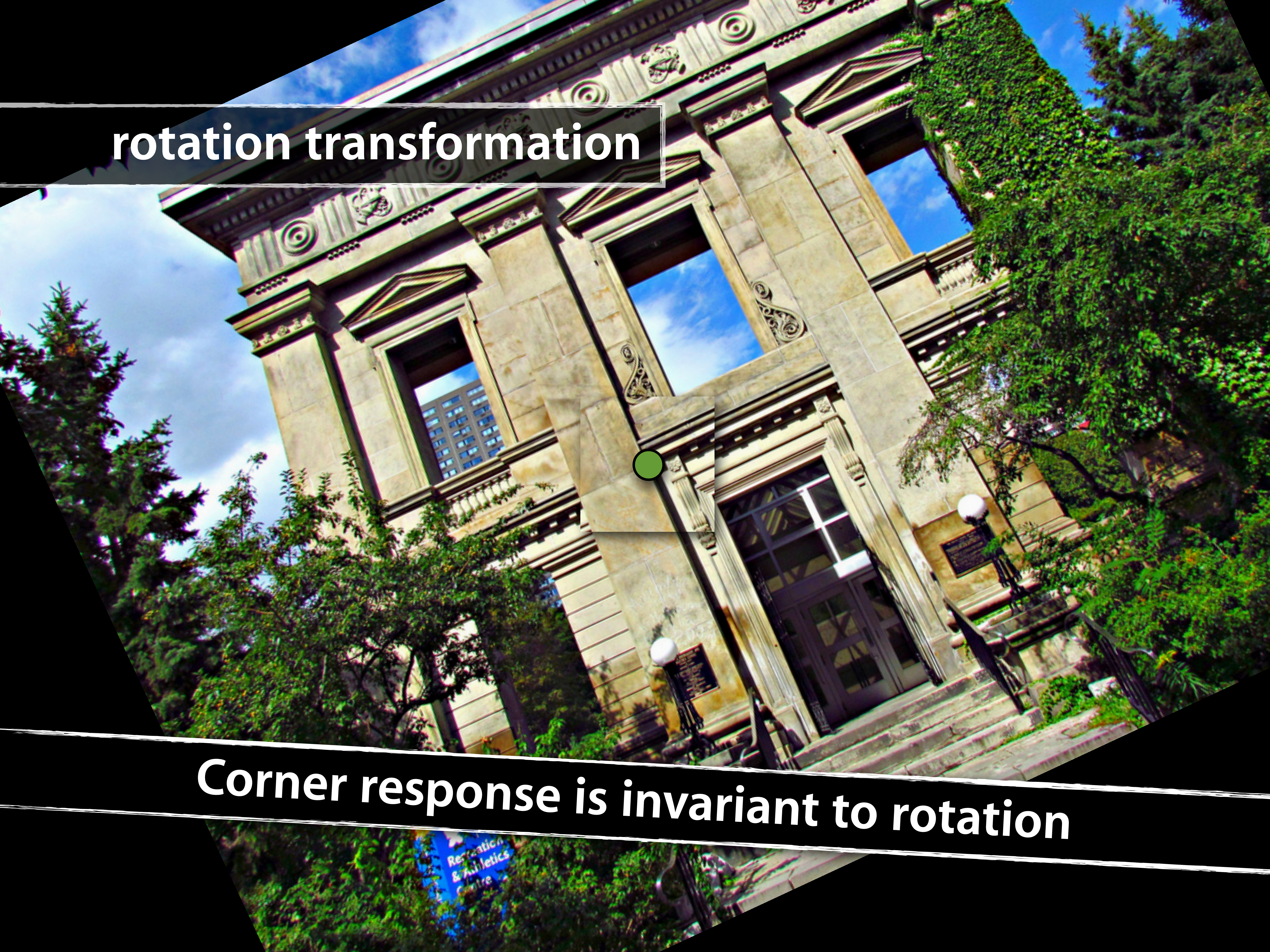
rotation transformation

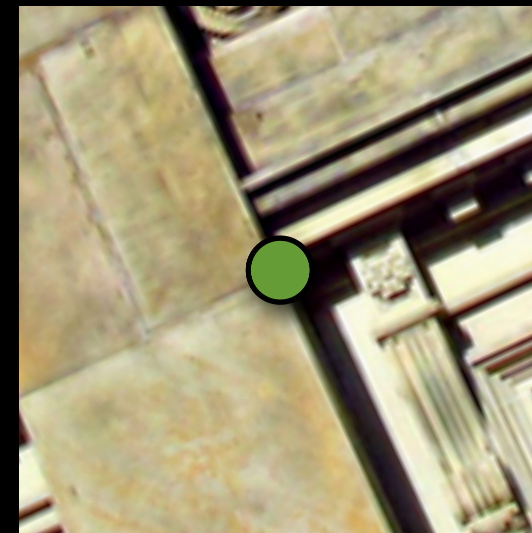


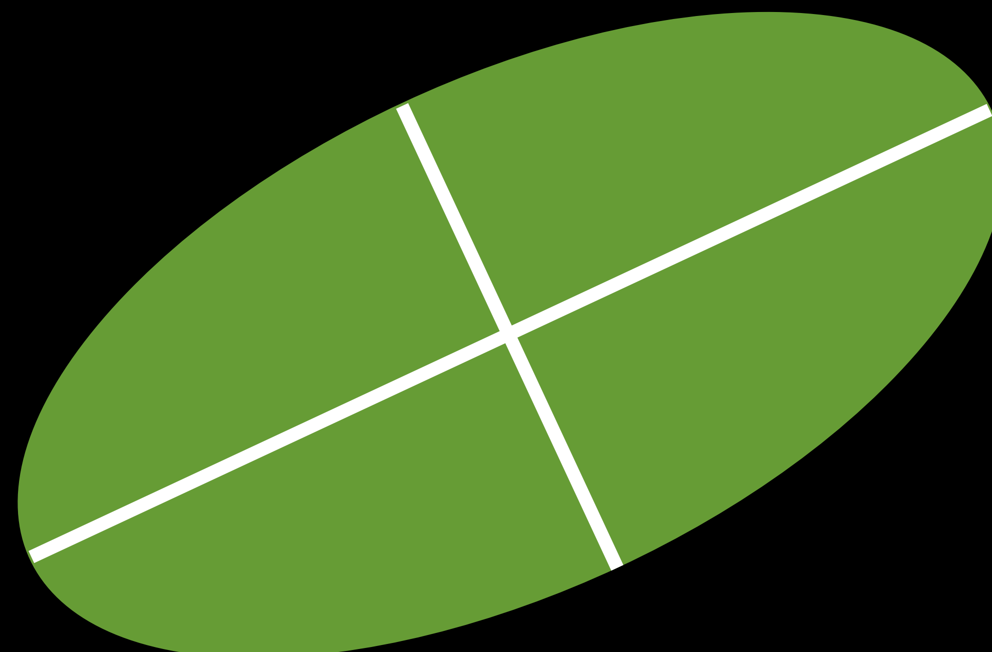
rotation transformation

Corner response is invariant to rotation

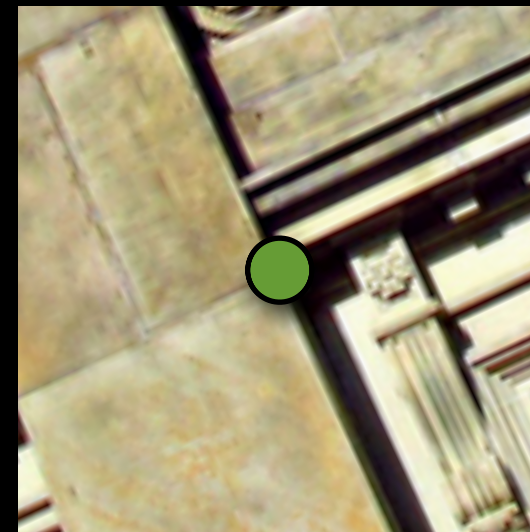

rotation transformation
Corner response is invariant to rotation

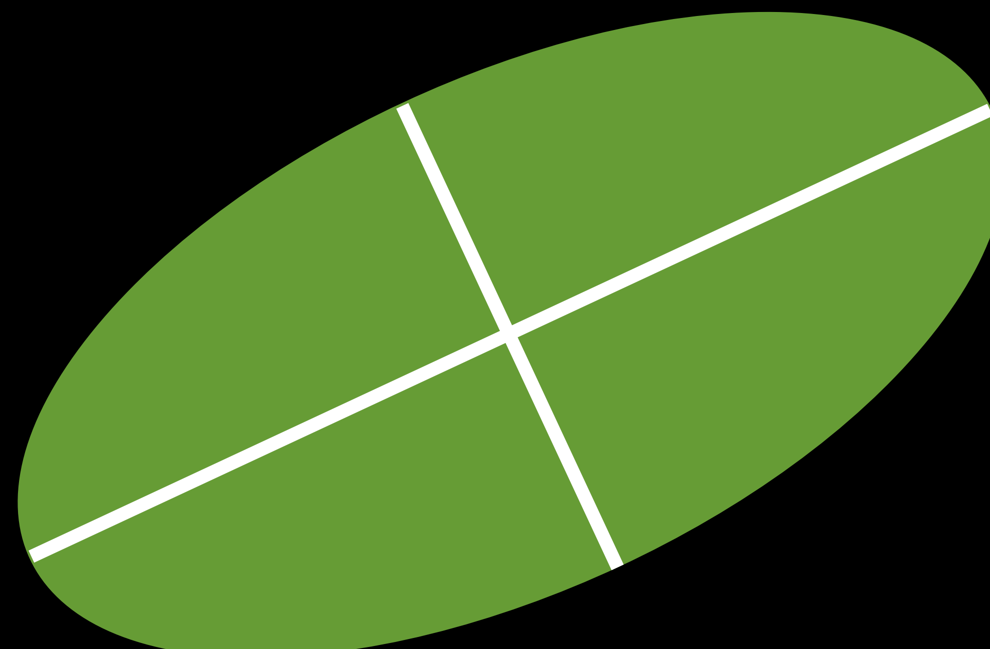

rotation transformation
Corner response is invariant to rotation



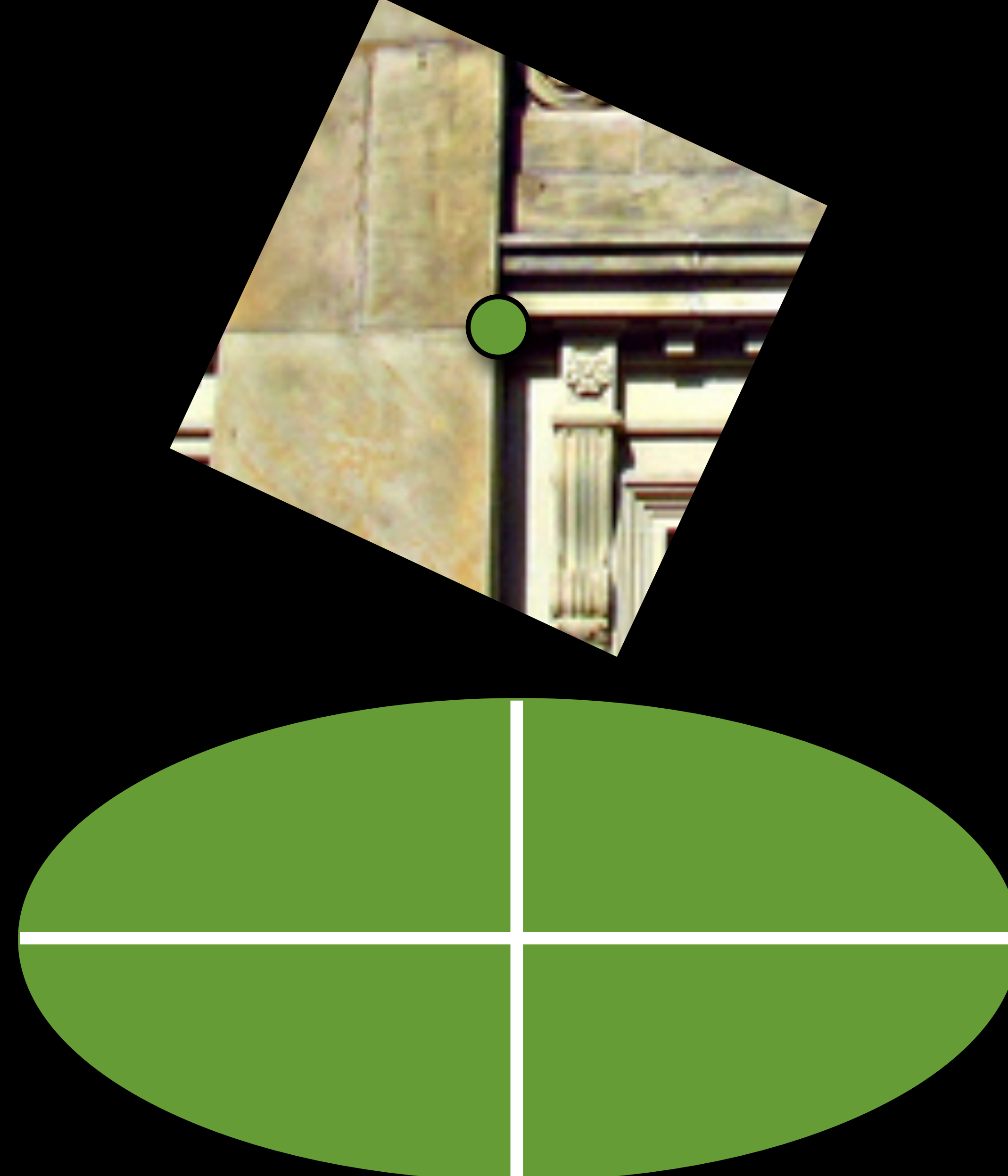


Rotation Invariant?



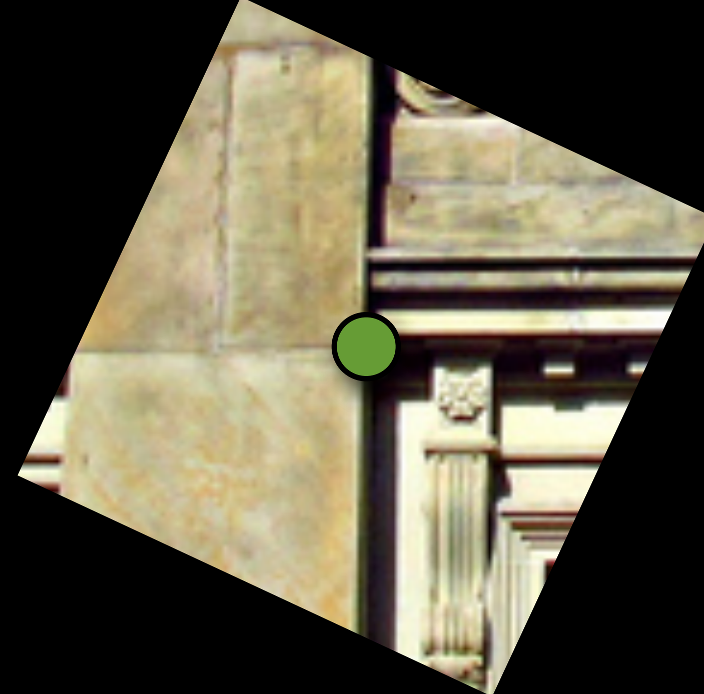


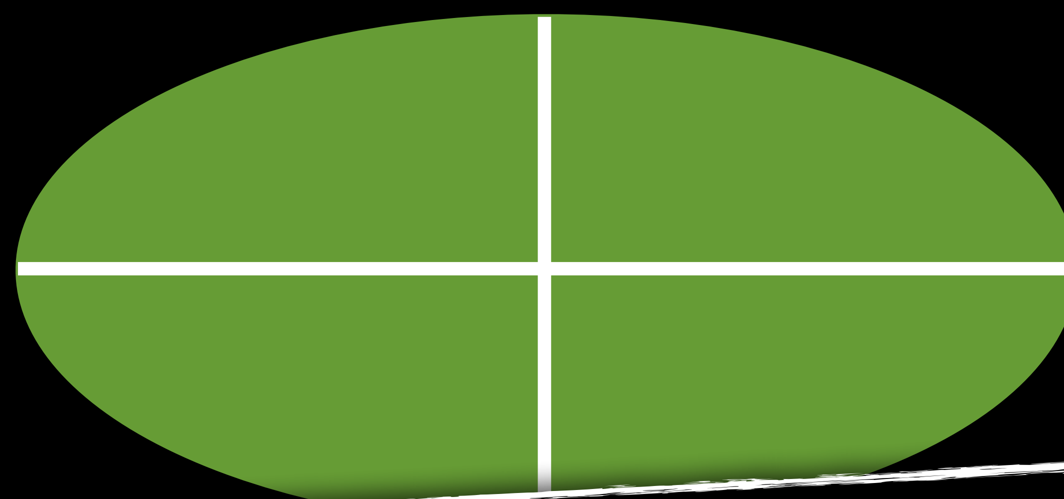
Rotation Invariant?



Rotation Invariant?

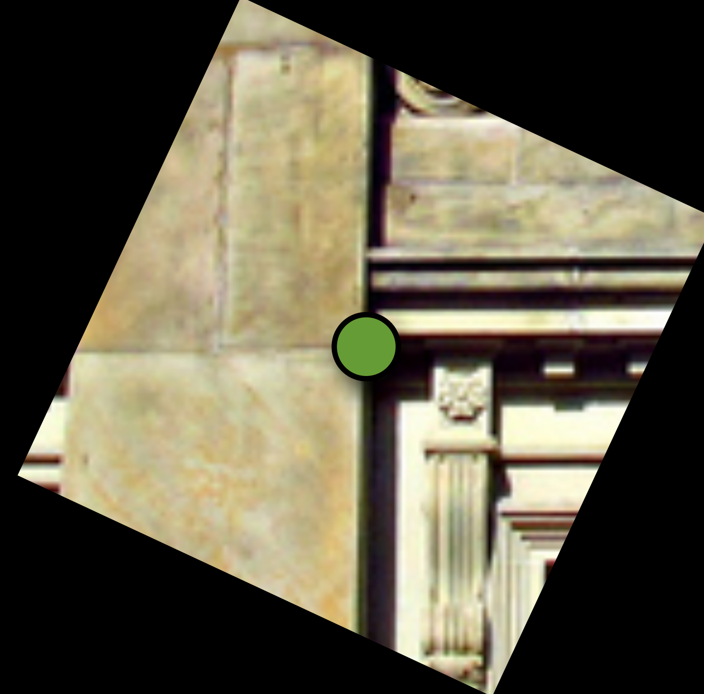
Rotation Invariant?

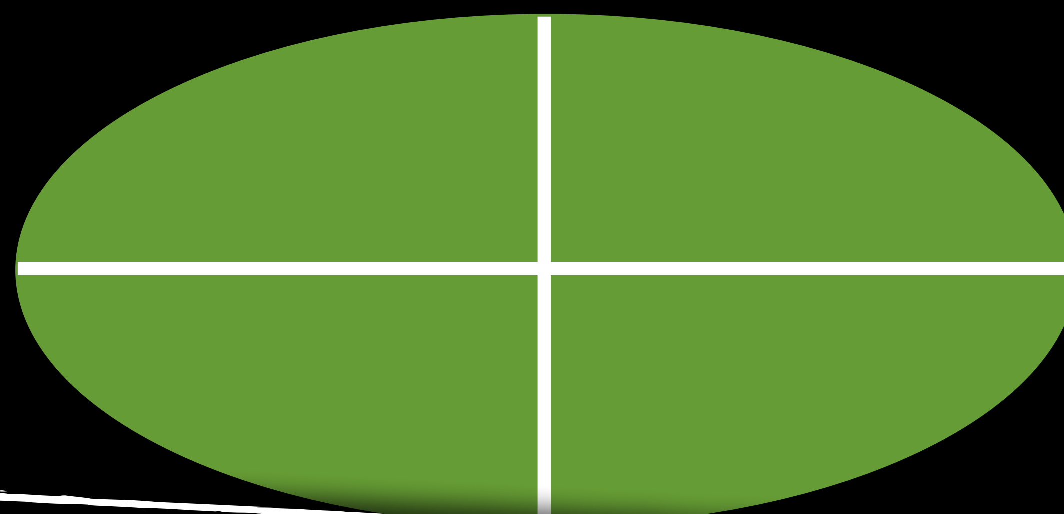




the shape remained the same

Rotation Invariant?





What else remained the same?

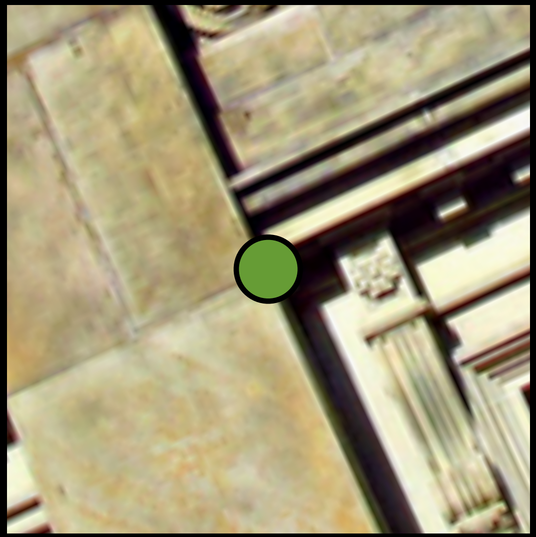
Local Intensity Variation

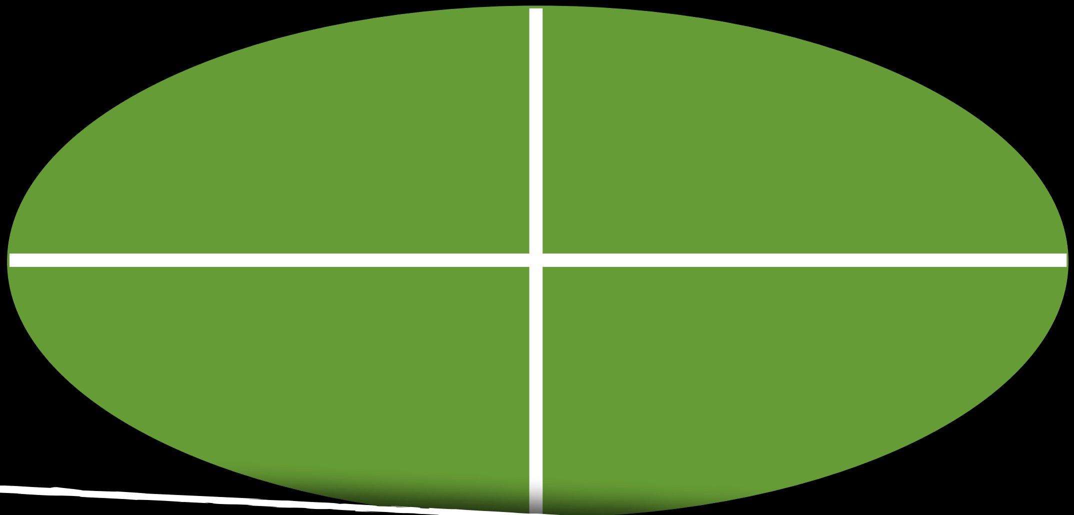
$$E(\Delta x, \Delta y) = \begin{pmatrix}\Delta x & \Delta y\end{pmatrix} \mathbf{M} \begin{pmatrix}\Delta x \\ \Delta y\end{pmatrix}$$

where

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

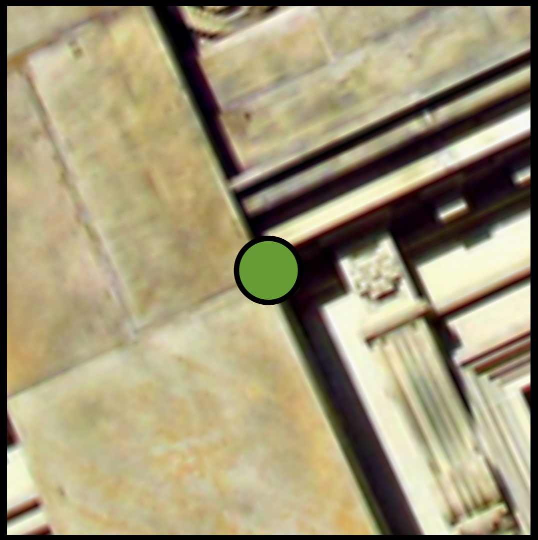
$$= \mathbf{R} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \mathbf{R}^\top$$

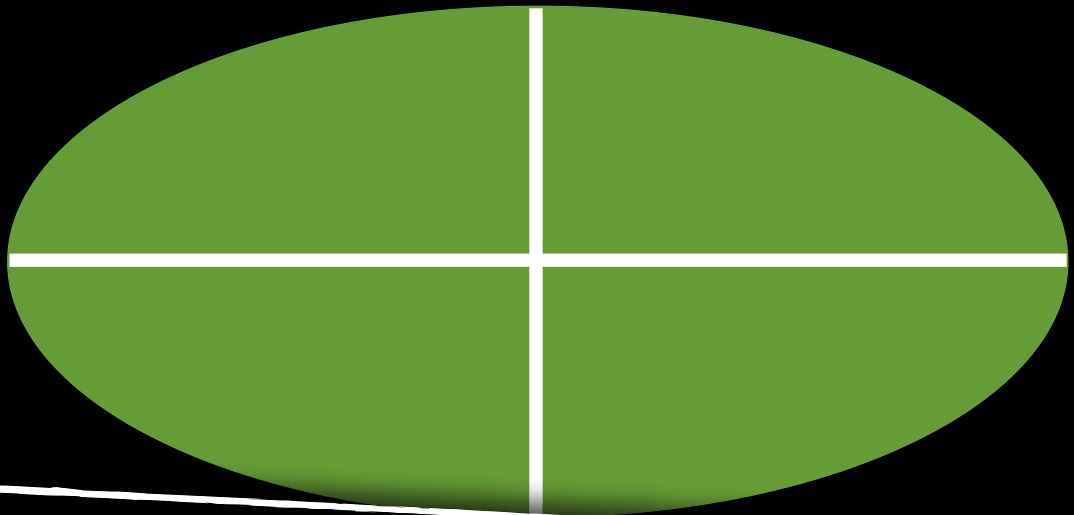




Rotation Invariant?

What else remained the same?

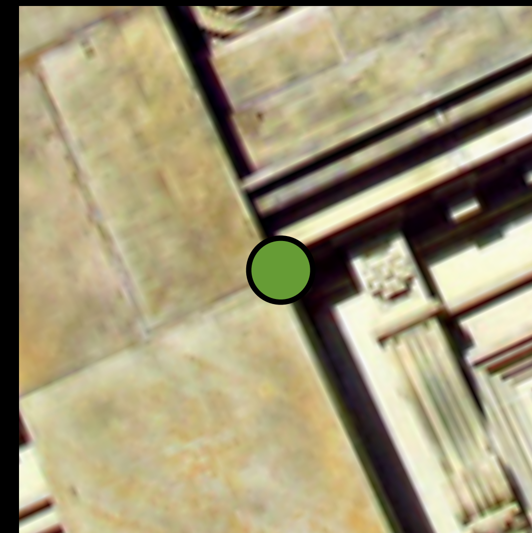


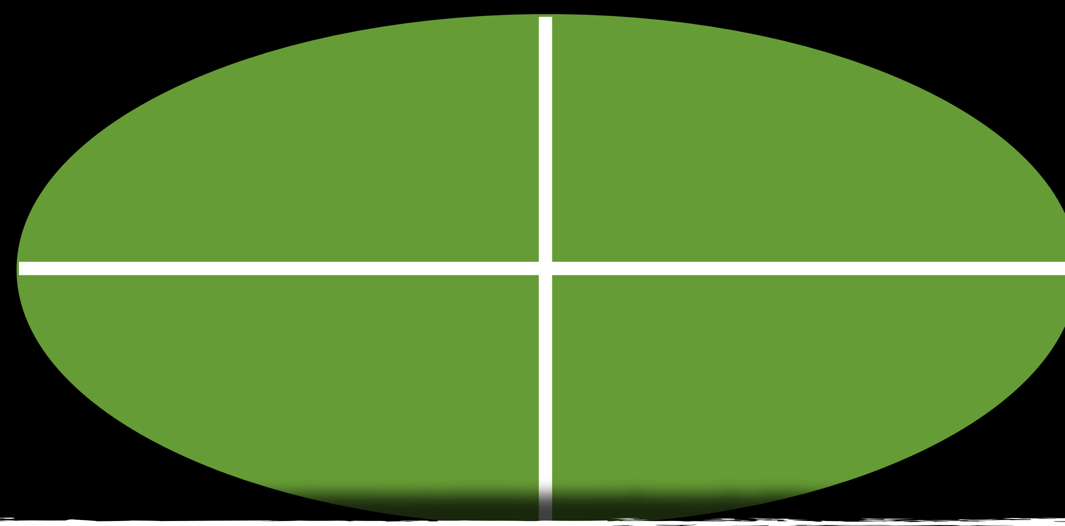


Rotation Invariant?

What else remained the same?

eigenvalues





Rotation Invariant?

Harris corner response is invariant to rotation

Harris Corner Properties

Rotation invariant

Harris Corner Properties

Rotation invariant

Partially invariant to illumination variation


input image


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

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

Intensity Invariant?

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

Corner response is a function of derivatives

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

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

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

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

Corner response is a function of derivatives

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

Corner response is a function of derivatives

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

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

Corner response is a function of derivatives

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

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

intensity shift

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

Corner response is a function of derivatives

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

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

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?

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

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

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

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?

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?

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?

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

Harris Corner Properties

Rotation invariant

Partially invariant to illumination variation

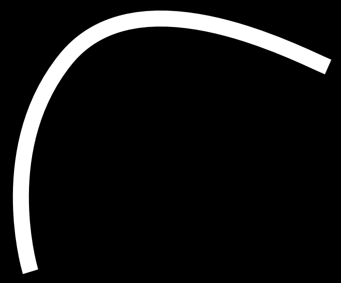
Harris Corner Properties

Rotation invariant

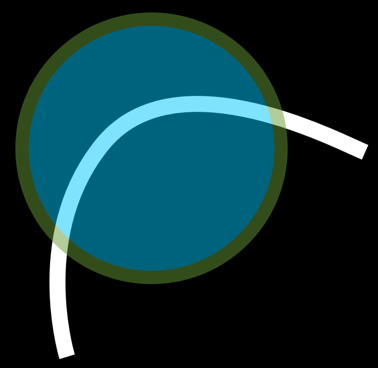
Partially invariant to illumination variation

NOT scale invariant

NOT scale invariant

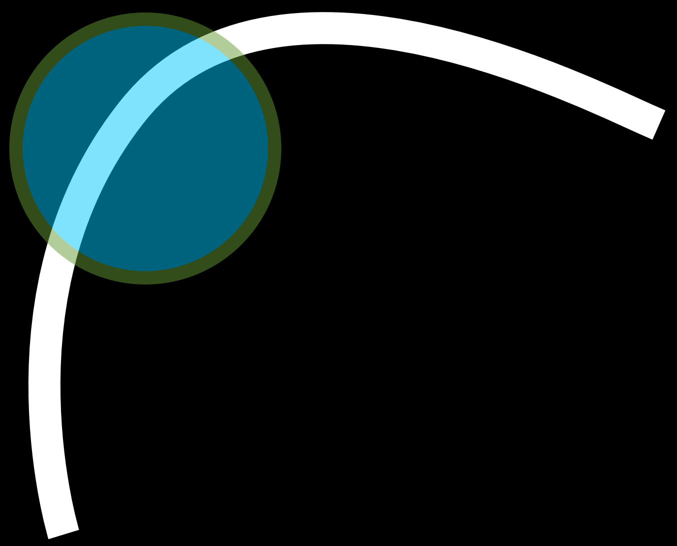


NOT scale invariant

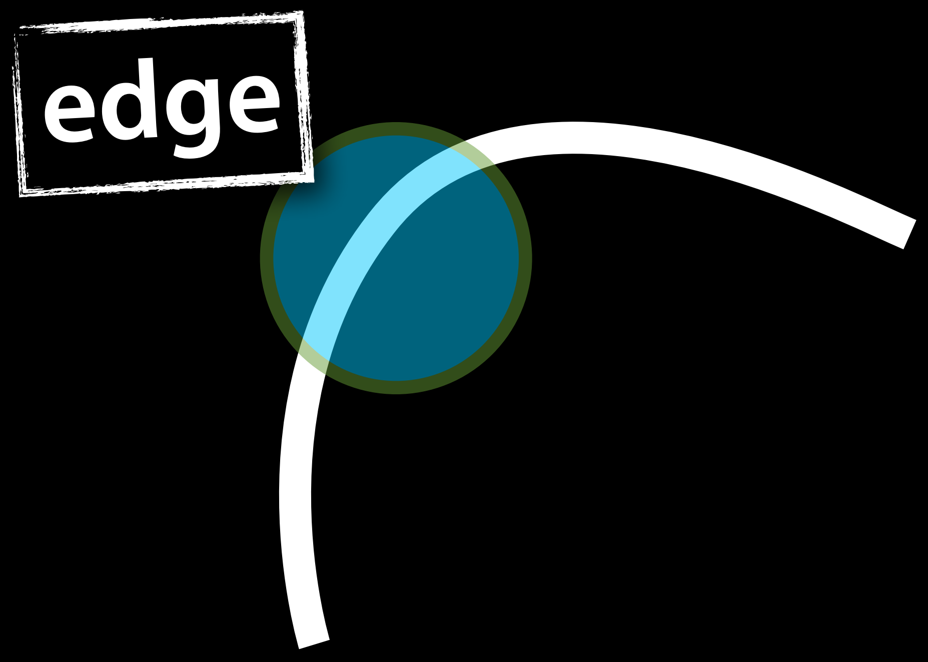


NOT scale invariant

NOT scale invariant



NOT scale invariant



Harris Corner Properties

Rotation invariant

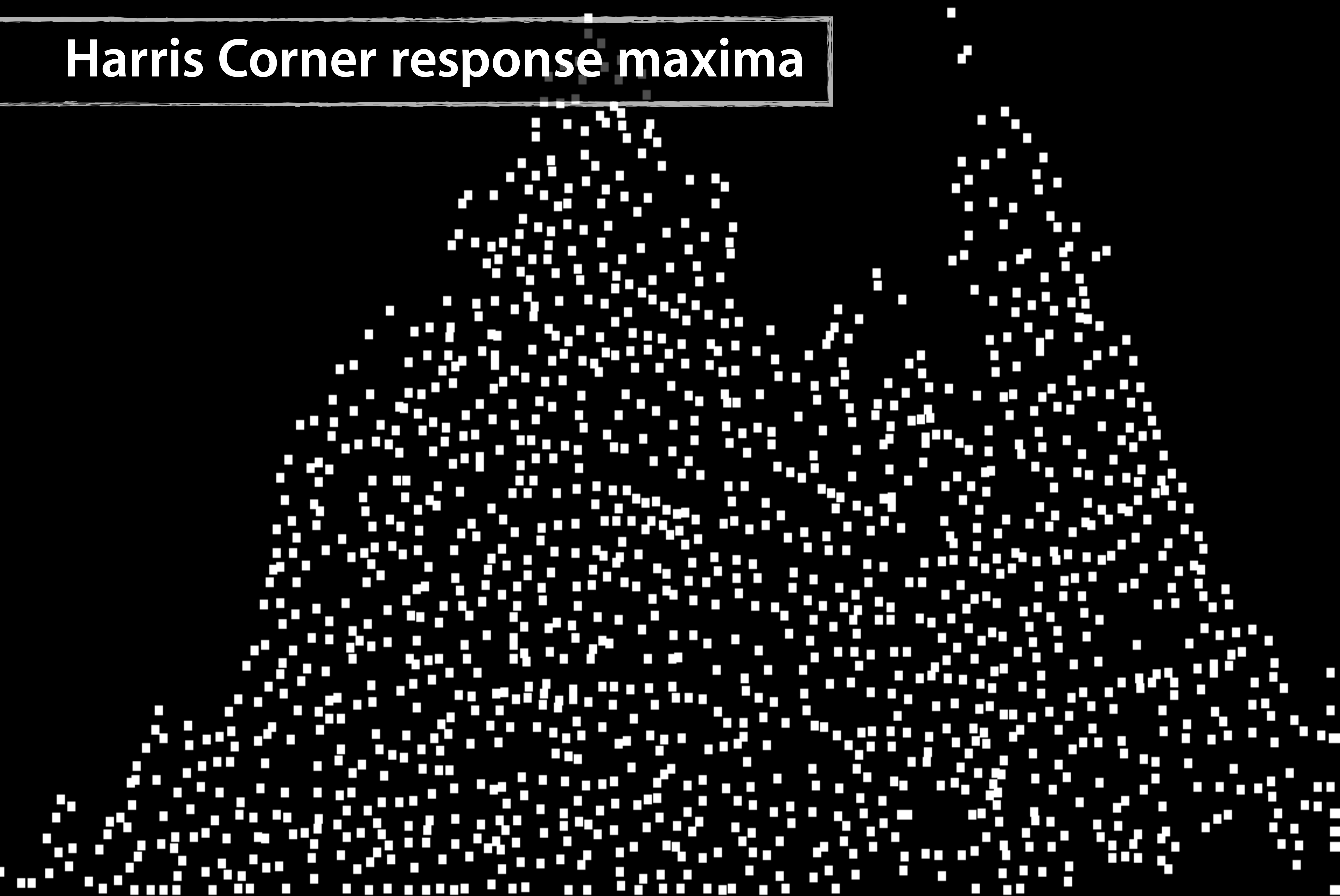
Partially invariant to illumination variation

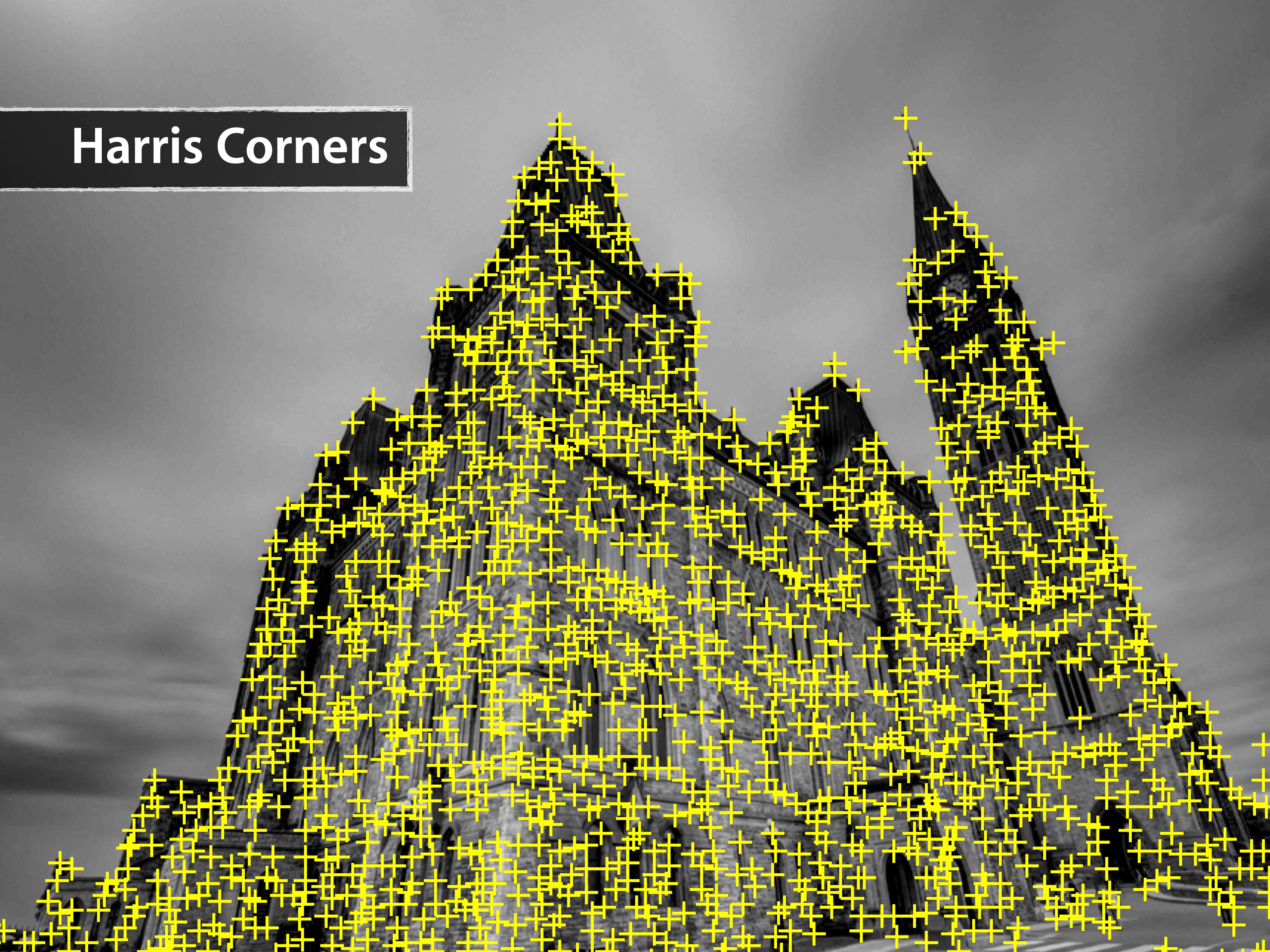
NOT scale invariant

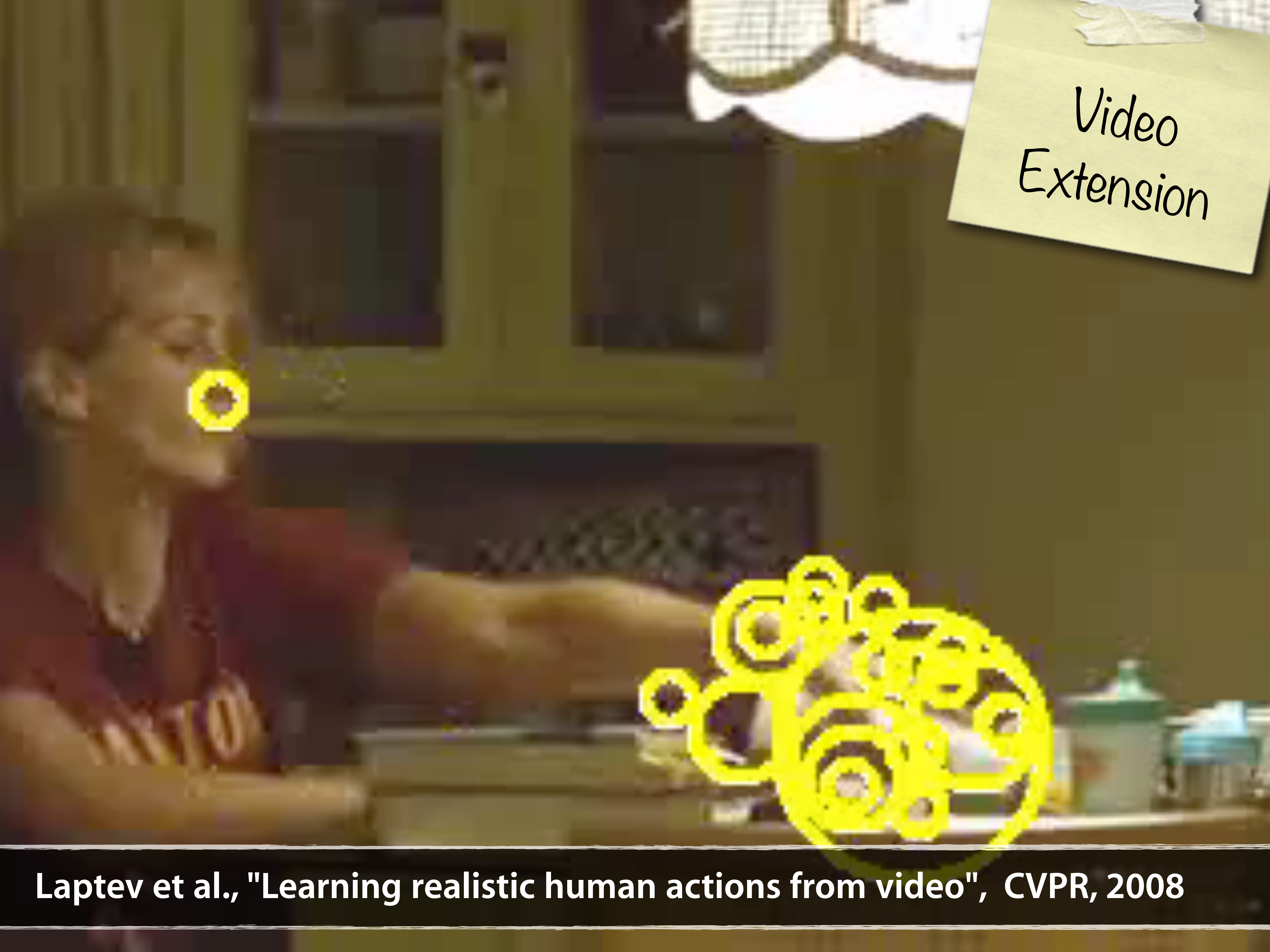

input image


Harris Corner responses

Harris Corner response maxima




Harris Corners



Laptev et al., "Learning realistic human actions from video", CVPR, 2008

Laptev et al., "Learning realistic human actions from video", CVPR, 2008