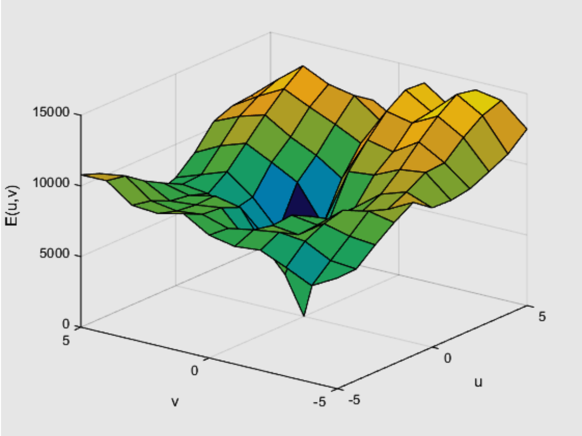
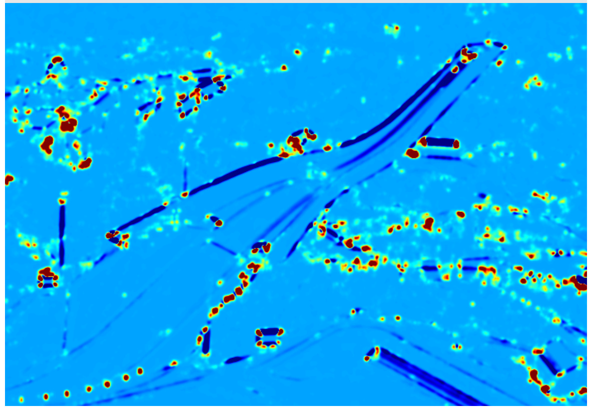
UiO : Department of Technology Systems
University of Oslo





Corner features

Trym Vegard Haavardsholm

2023

With illustrations from Rick Szeliski, S. Seitz. Svetlana Lazebnik, Derek Hoiem, Grauman&Leibe, James Hayes and Noah Snavely

TEK5030

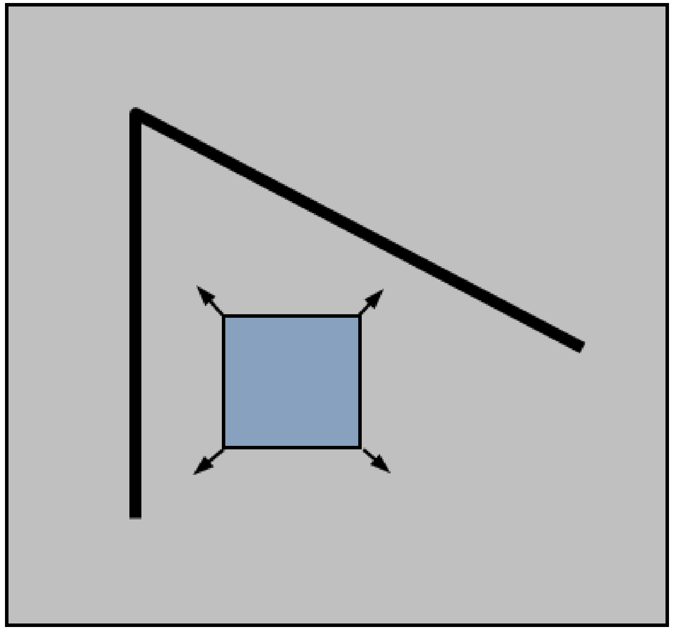
Characteristics of good features

- Repeatability
- Distinctiveness
- Efficiency
- Locality

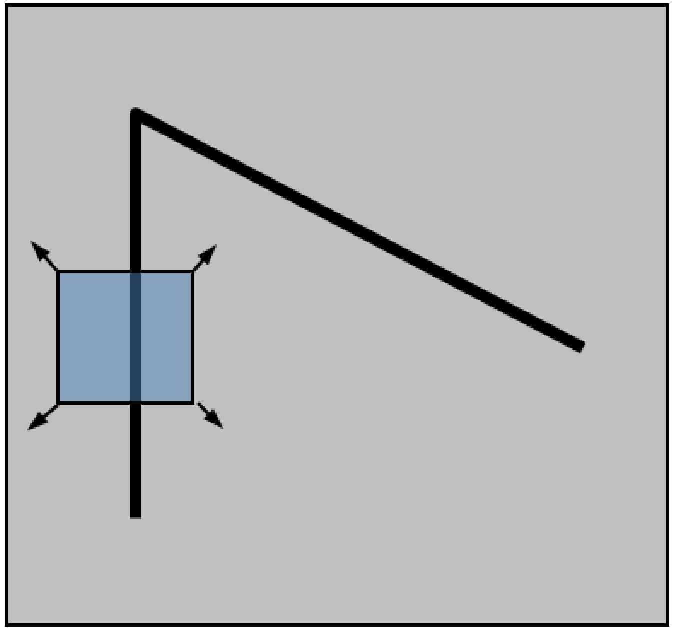
Local measure of feature distinctiveness

Consider a small window of pixels around a feature:

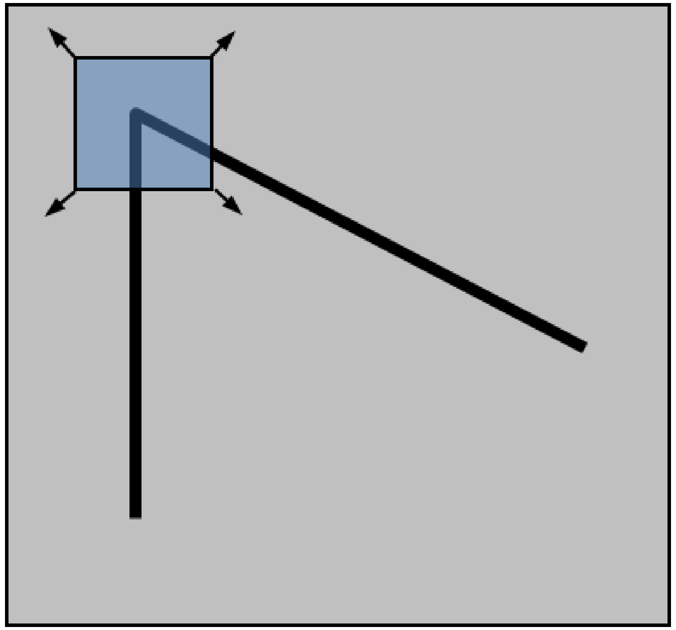
- How does the window change when you shift it?



“Flat” region:
No change in all directions



“Edge”:
No change along edge



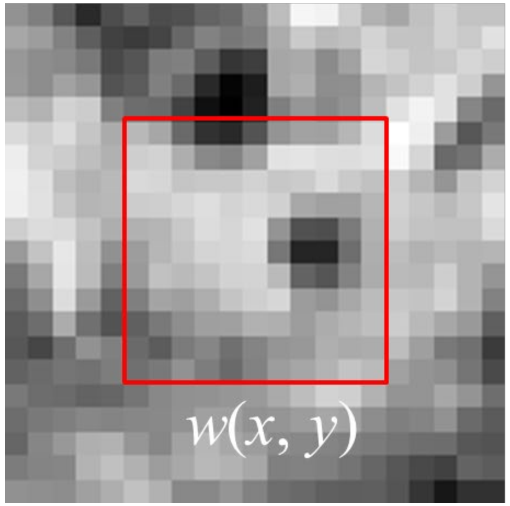
“Corner”:
Change in all directions

Local measure of feature distinctiveness

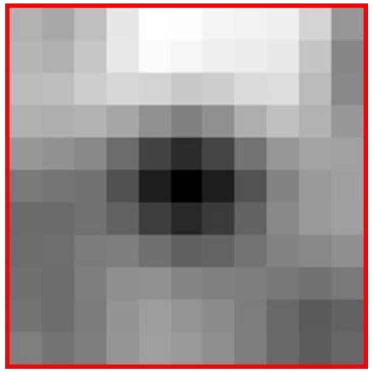
Change in appearance of window *w*(*x,y*) for the shift [*u,v*]:

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

$I(x, y)$



$E(u, v)$

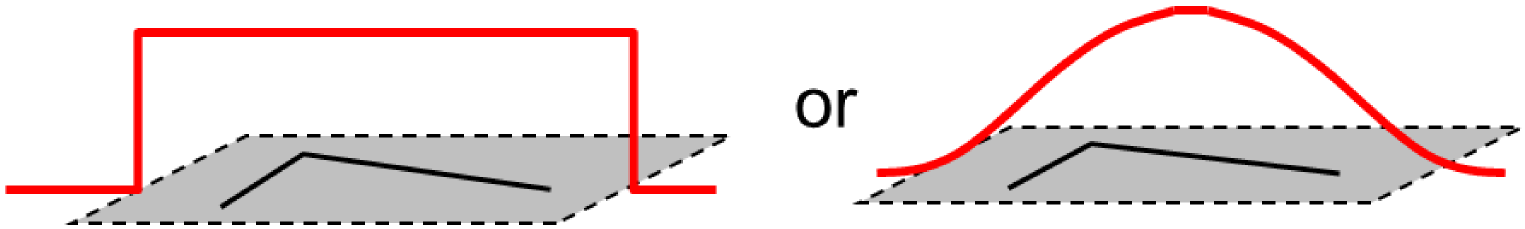


Local measure of feature distinctiveness

Change in appearance of window $w(x,y)$ for the shift $[u,v]$:

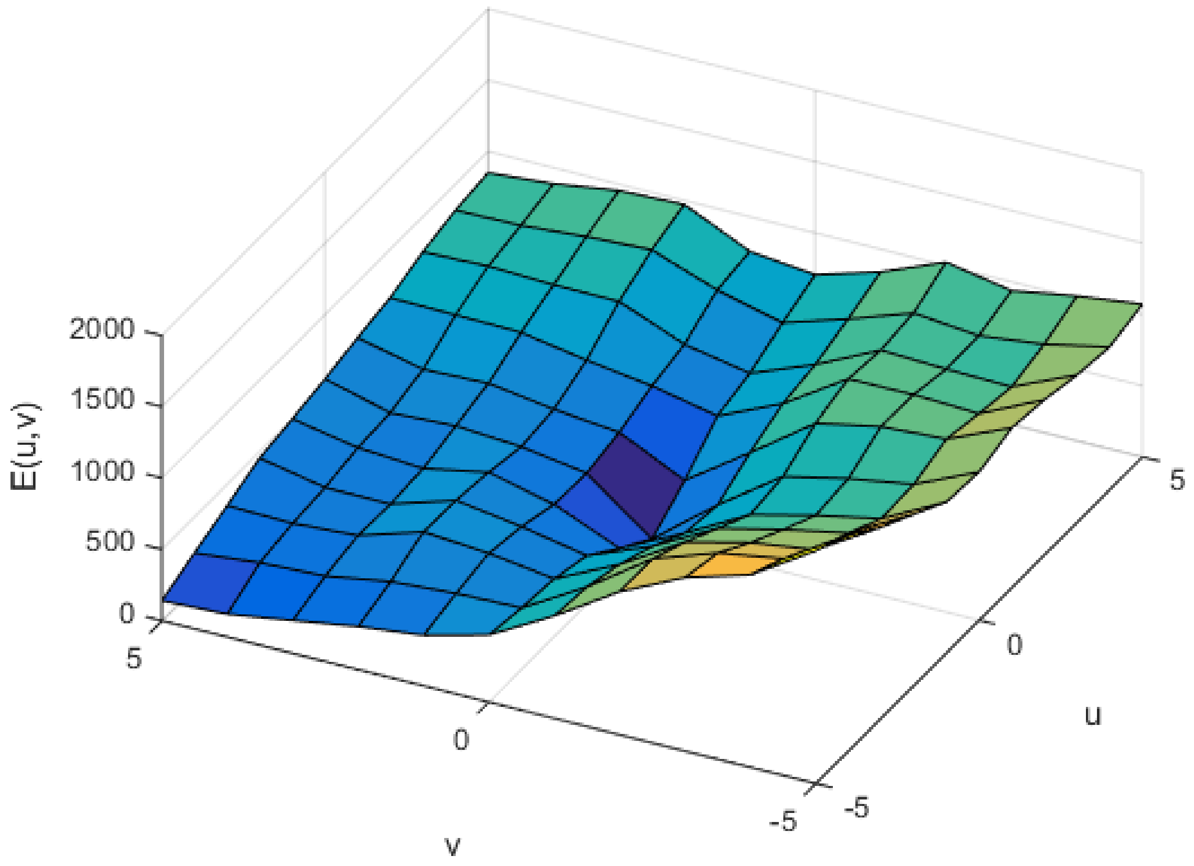
$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u, y+v) - I(x,y)\right]^2$$

Window function $w(x,y) =$

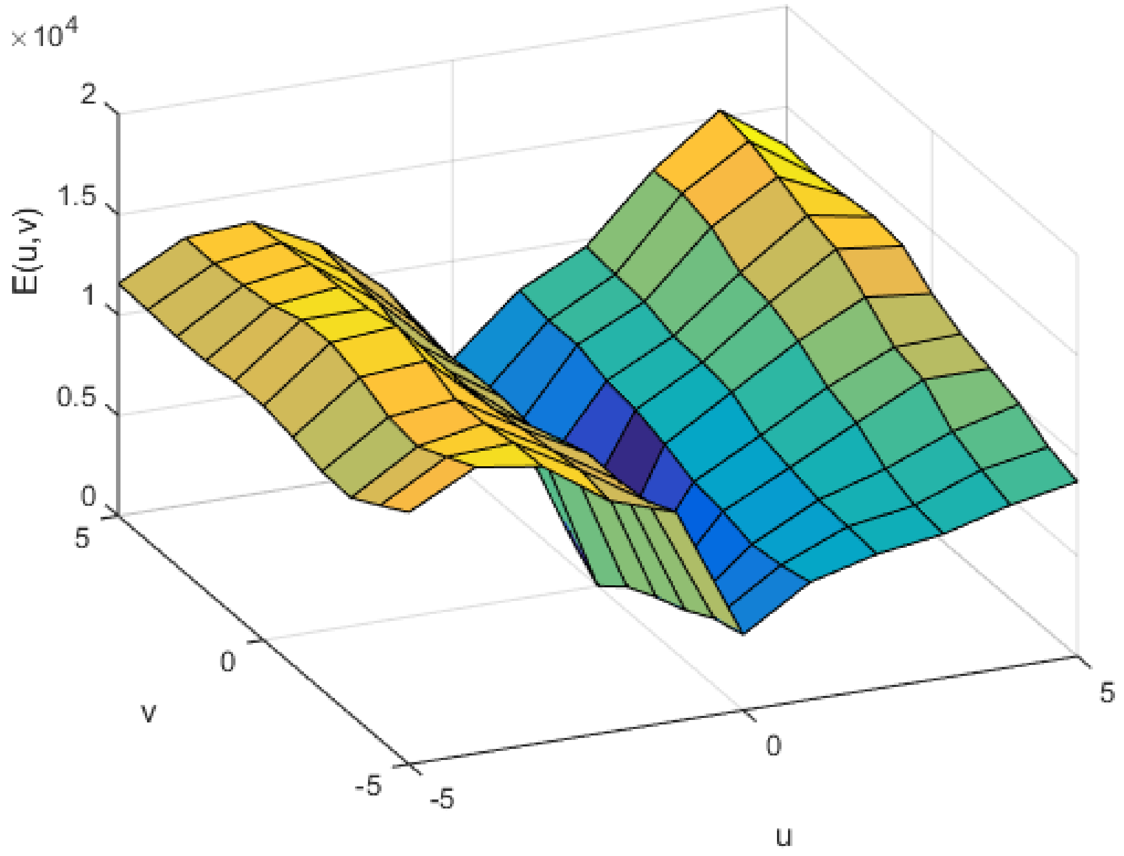


1 in window, 0 outside   Gaussian

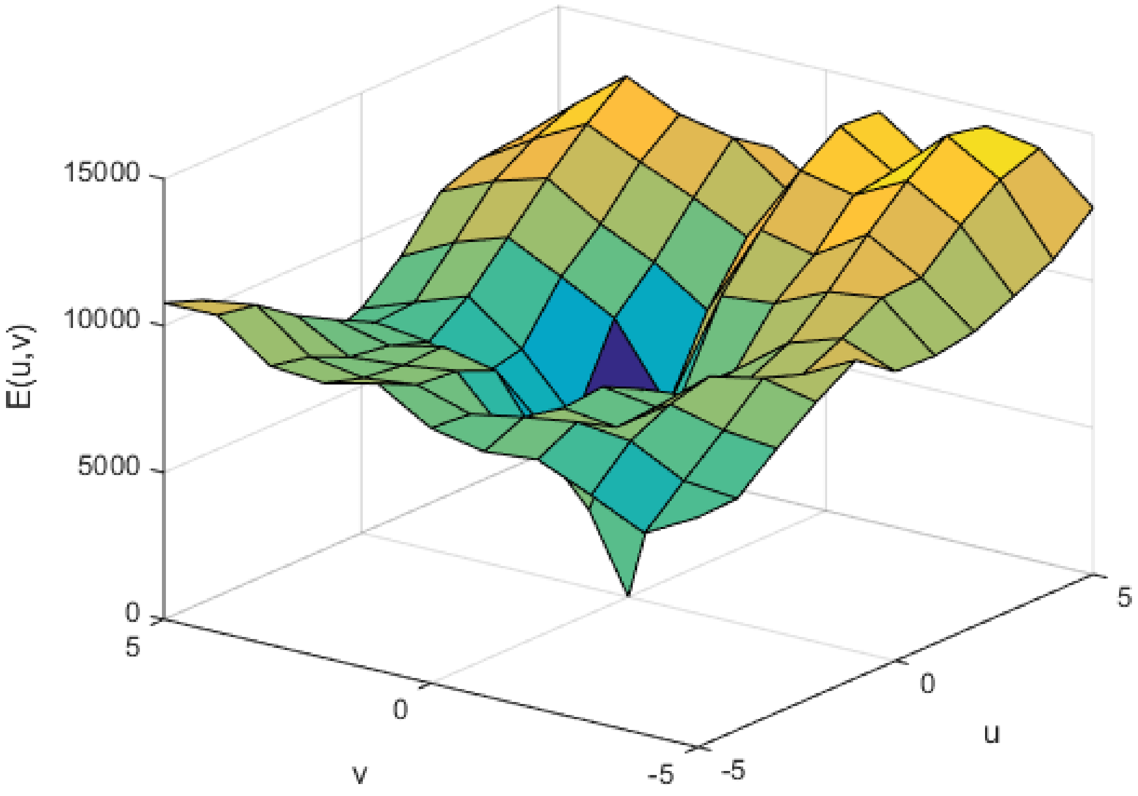
Examples from Holmenkollen



Examples from Holmenkollen



Examples from Holmenkollen



Simplifying the measure

Local first order Taylor Series expansion of *I*(*x,y*):

$$I(x+u, y+v) \approx I(x, y) + I_x u + I_y v$$

Local quadratic approximation of *E*(*u,v*):

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

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

$$= Au^2 + 2Buv + Cv^2$$

Simplifying the measure

Local quadratic approximation of the surface *E*(*u*,*v*):

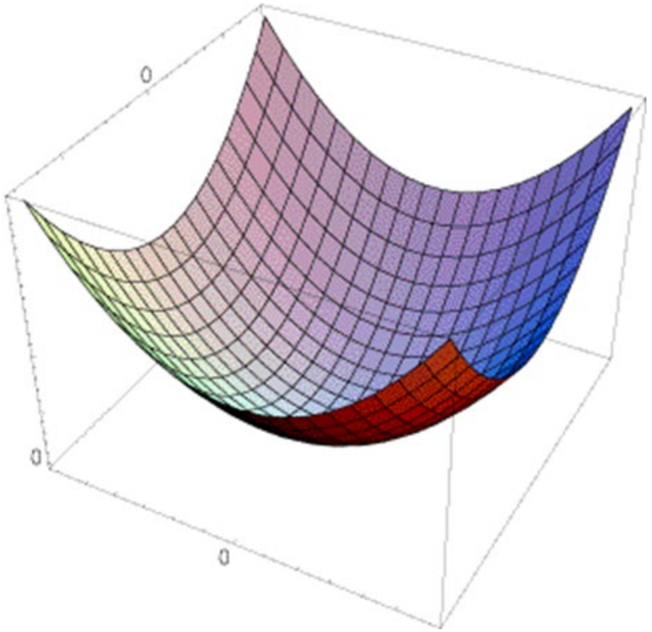
$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

$$A = \sum_{x,y} w(x,y) I_x^2$$

$$B = \sum_{x,y} w(x,y) I_x I_y$$

$$C = \sum_{x,y} w(x,y) I_y^2$$

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



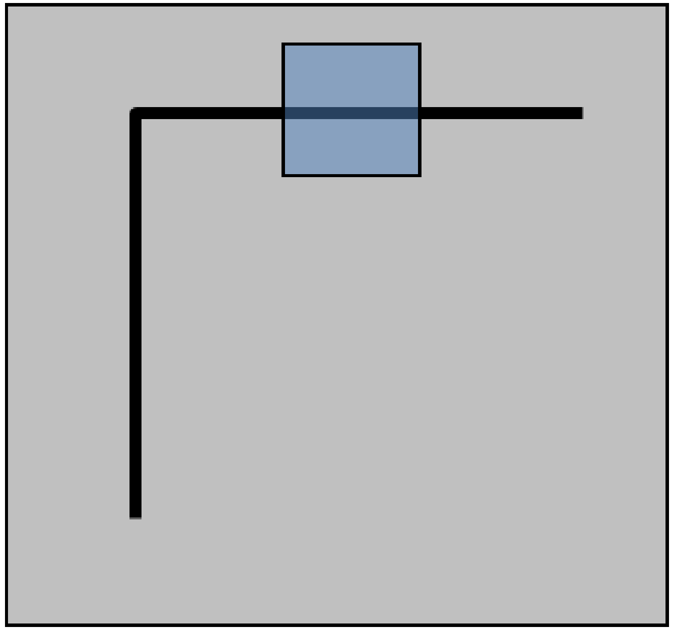
Interpreting the quadratic surface

$$E(u,v) \approx [u \quad v]\begin{bmatrix} A & B \\ B & C \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = [u \quad v]\, M \begin{bmatrix} u \\ v \end{bmatrix}$$

$$A = \sum_{x,y} w(x,y) I_x^2$$

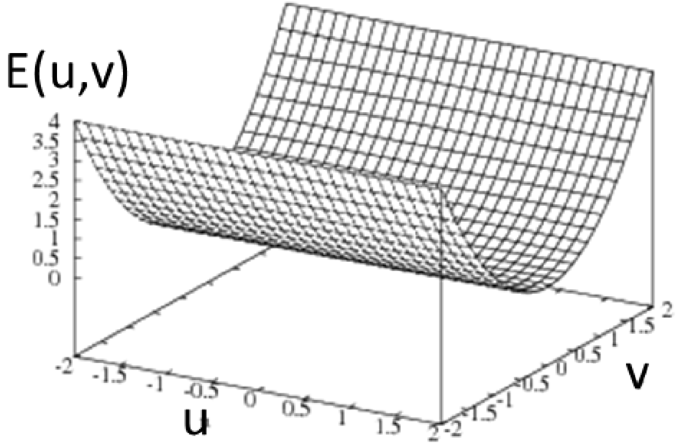
$$B = \sum_{x,y} w(x,y) I_x I_y$$

$$C = \sum_{x,y} w(x,y) I_y^2$$



Horizontal edge: $I_x = 0$

$$M = \begin{bmatrix} 0 & 0 \\ 0 & C \end{bmatrix}$$



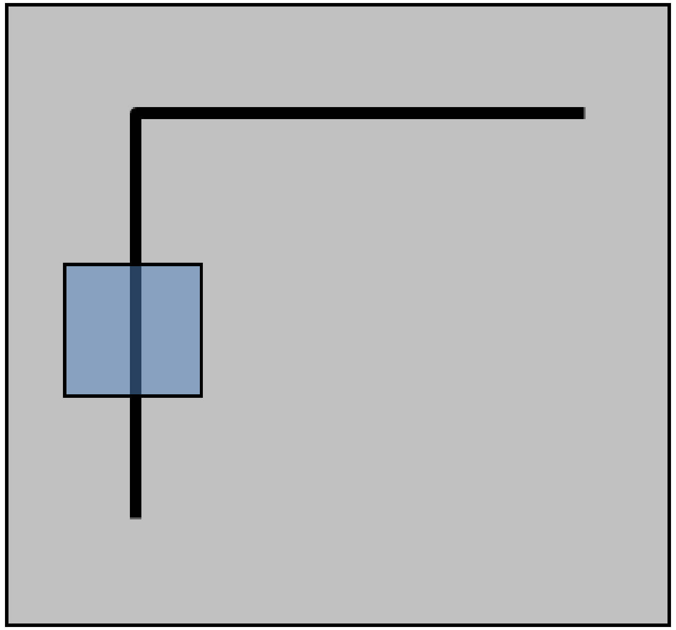
Interpreting the quadratic surface

$$E(u,v) \approx [u \quad v] \begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = [u \quad v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

$$A = \sum_{x,y} w(x,y) I_x^2$$

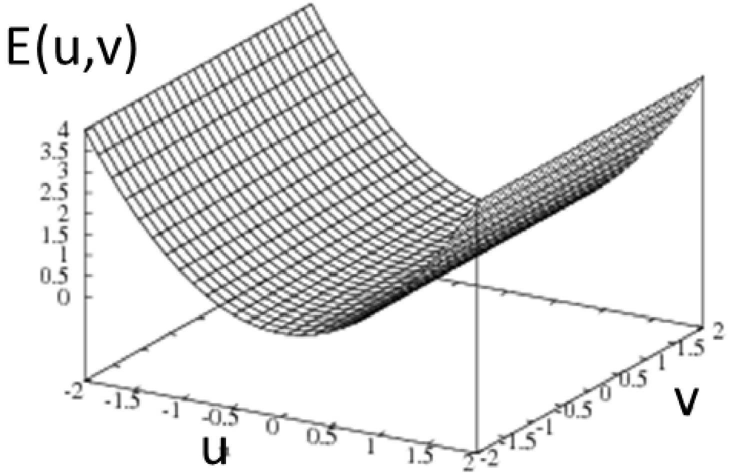
$$B = \sum_{x,y} w(x,y) I_x I_y$$

$$C = \sum_{x,y} w(x,y) I_y^2$$

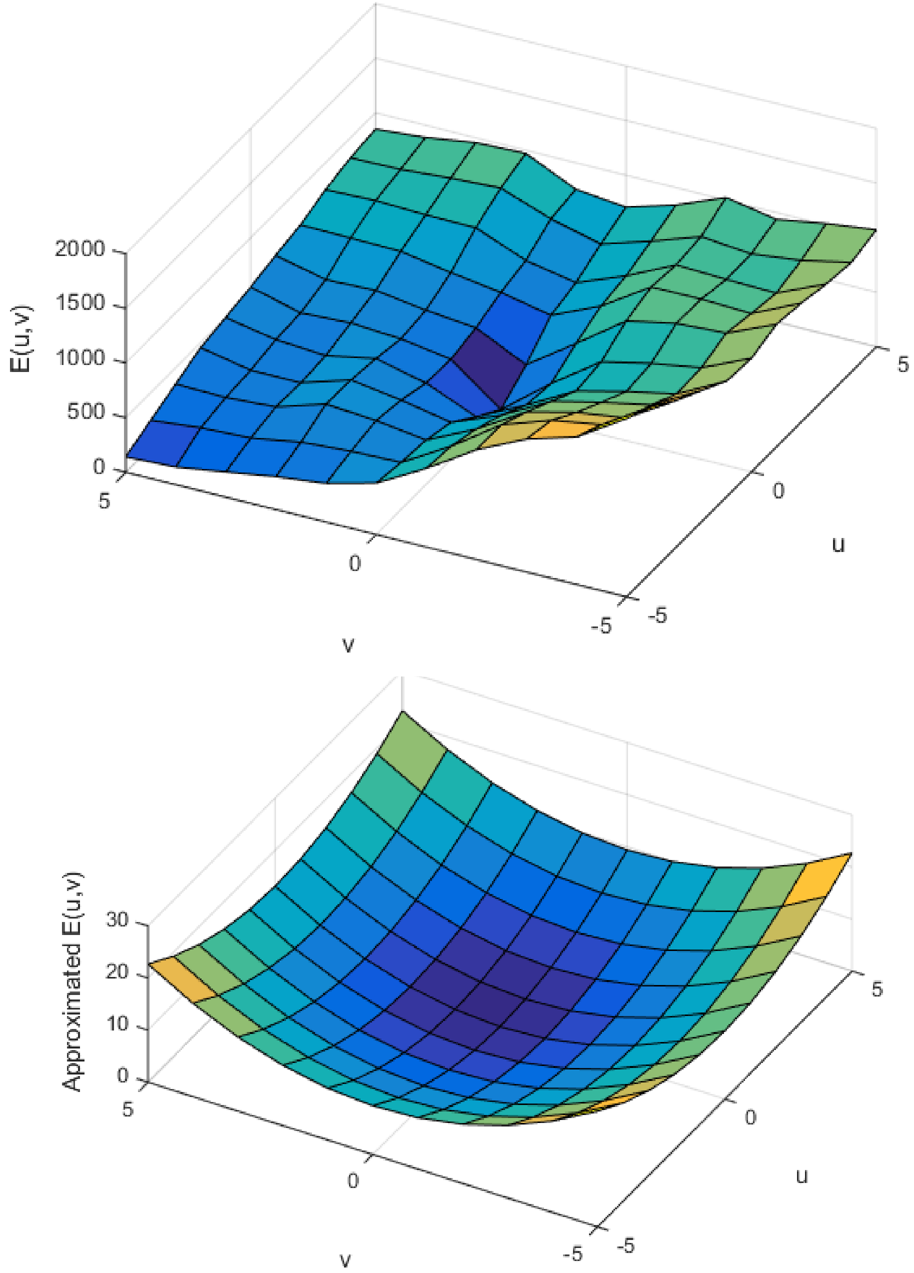


Vertical edge: $I_y = 0$

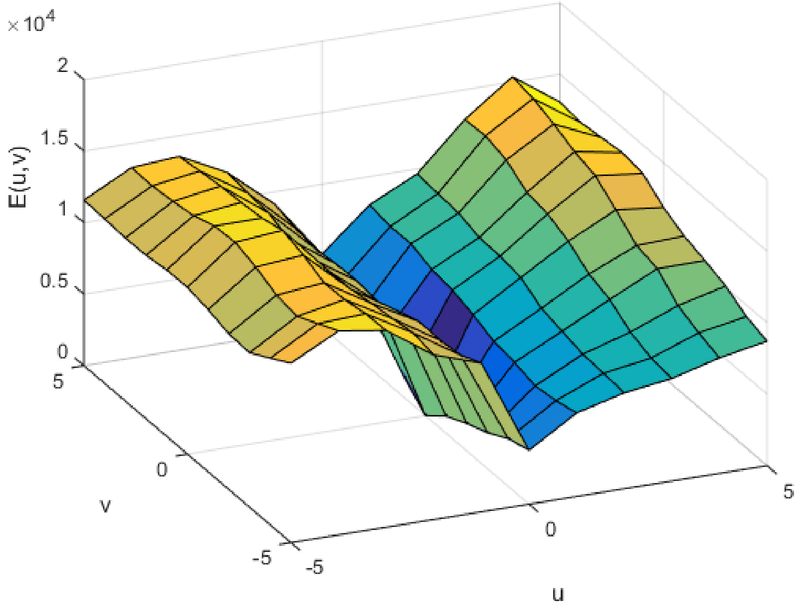
$$M = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}$$

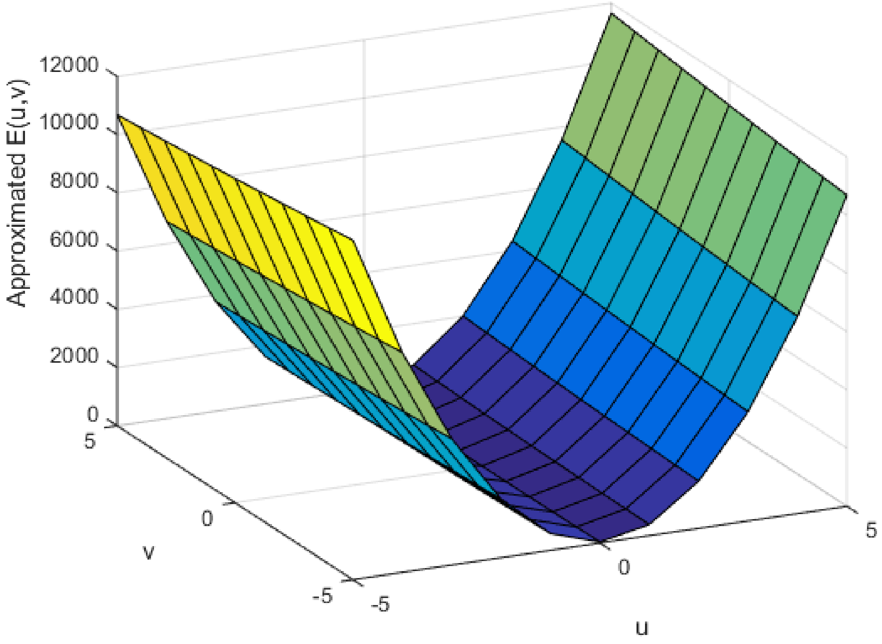


Examples from Holmenkollen

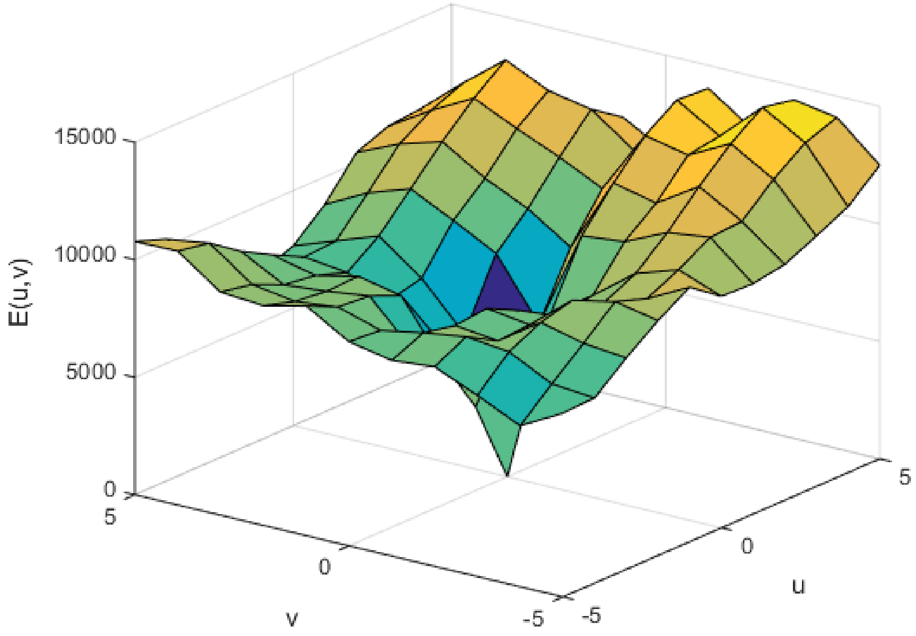


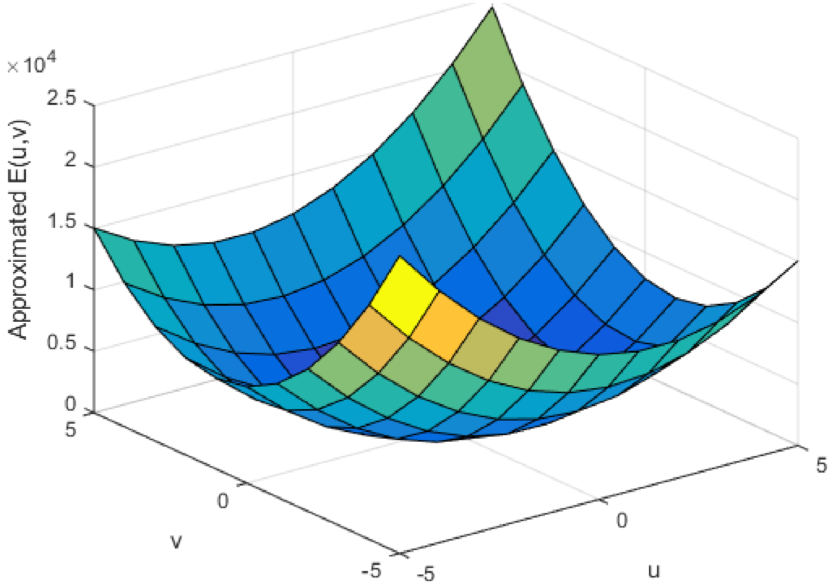
Examples from Holmenkollen





Examples from Holmenkollen



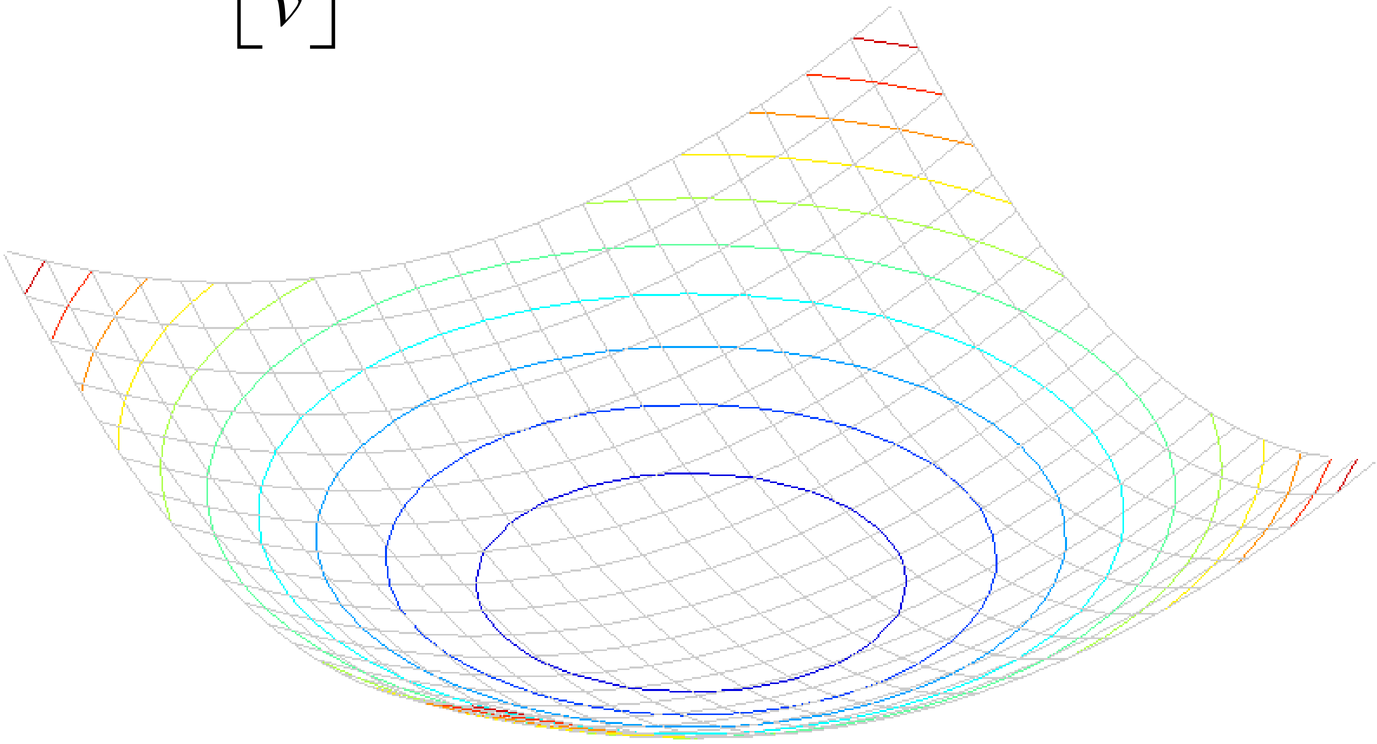


Simplifying the measure even further

Consider a horizontal “slice” of *E*(*u*,*v*):

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} = const$$

This is the equation of an ellipse



Simplifying the measure even further

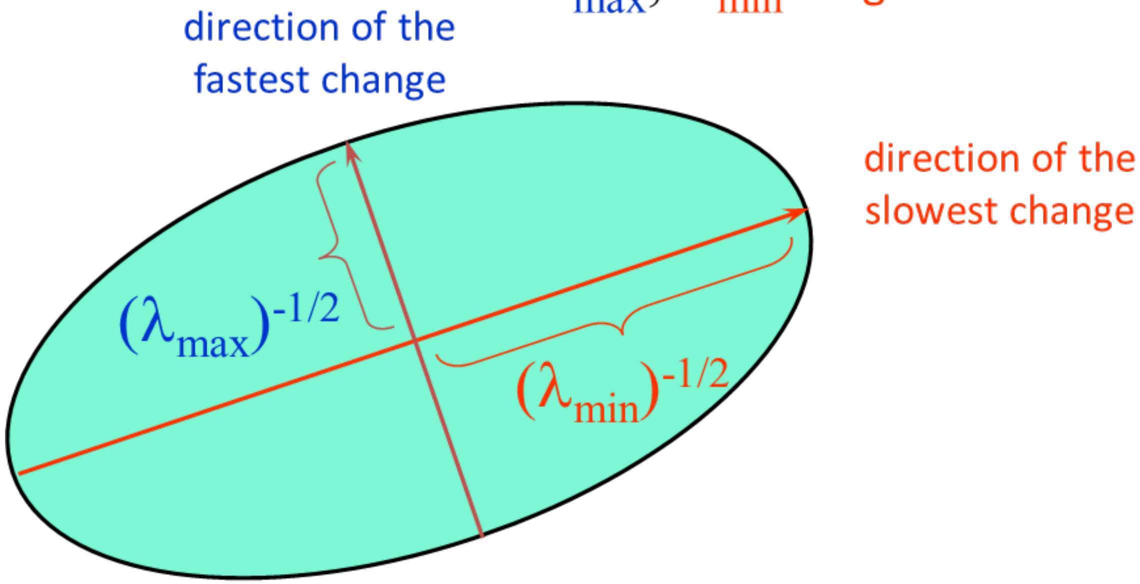
Consider a horizontal "slice" of $E(u,v)$:

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} = const$$

This is the equation of an ellipse

- The ellipses indicate the rate and direction of change
- This is described by the eigenvalues of M

λ_{max}, λ_{min} : eigenvalues of M



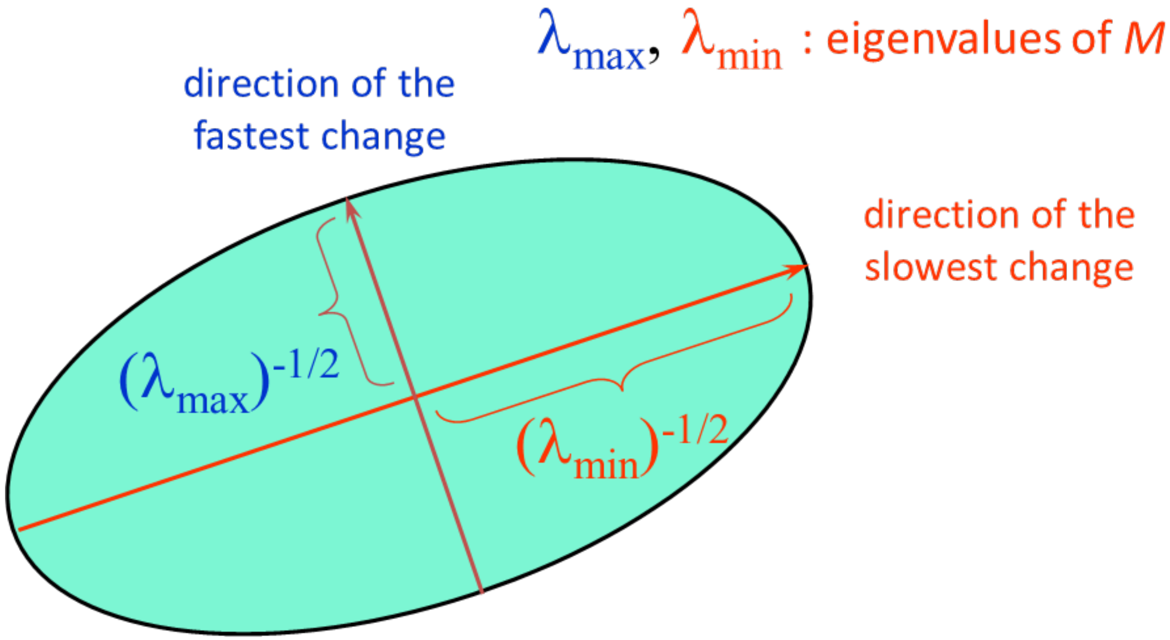
Simplifying the measure even further

Consider a horizontal "slice" of $E(u,v)$:

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} = const$$

This is the equation of an ellipse

- The ellipses indicate the rate and direction of change
- This is described by the eigenvalues of *M*
- ➢ Describe the surface using the eigenvalues!



The eigenvalues and eigenvectors of *M*

The eigenvalues

$$\lambda = \frac{1}{2}\left[(A+C) \pm \sqrt{4B^2 + (A-C)^2}\right]$$

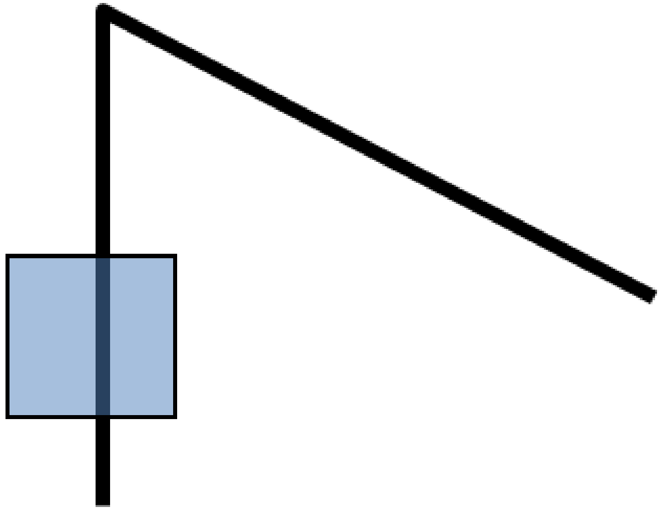
Once you know λ, you find the eigenvectors **x** by solving

$$\begin{bmatrix} A-\lambda & B \\ B & C-\lambda \end{bmatrix}\mathbf{x} = 0$$

The eigenvalues and eigenvectors of *M*

Describe the shift directions with the smallest and largest change in error:

- $\mathbf{x}_{max}$ = direction of largest increase in E
- λ_{max} = amount of increase in direction $\mathbf{x}_{max}$
- $\mathbf{x}_{min}$ = direction of smallest increase in E
- λ_{min} = amount of increase in direction $\mathbf{x}_{min}$



Local measure of feature distinctiveness

How are λ_{max}, $\mathbf{x}_{max}$, λ_{min}, $\mathbf{x}_{min}$ relevant for feature detection?

- What is our feature scoring function?

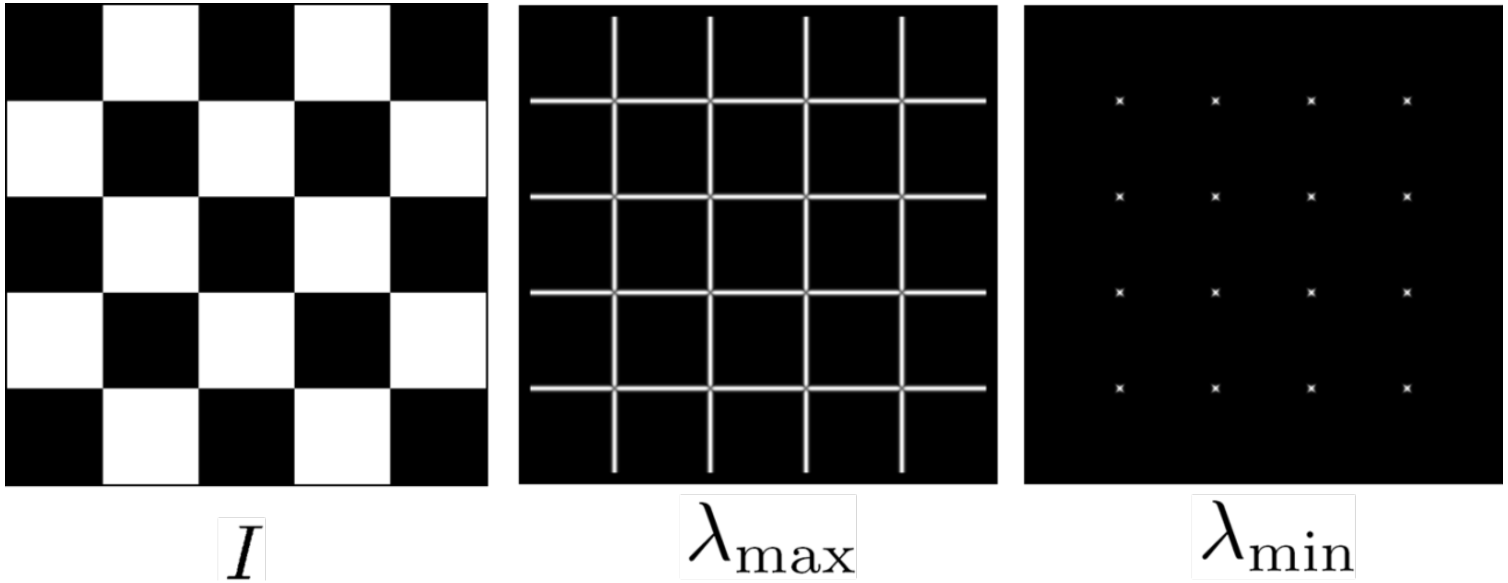
Local measure of feature distinctiveness

How are λ_{max}, $\mathbf{x}_{max}$, λ_{min}, $\mathbf{x}_{min}$ relevant for feature detection?

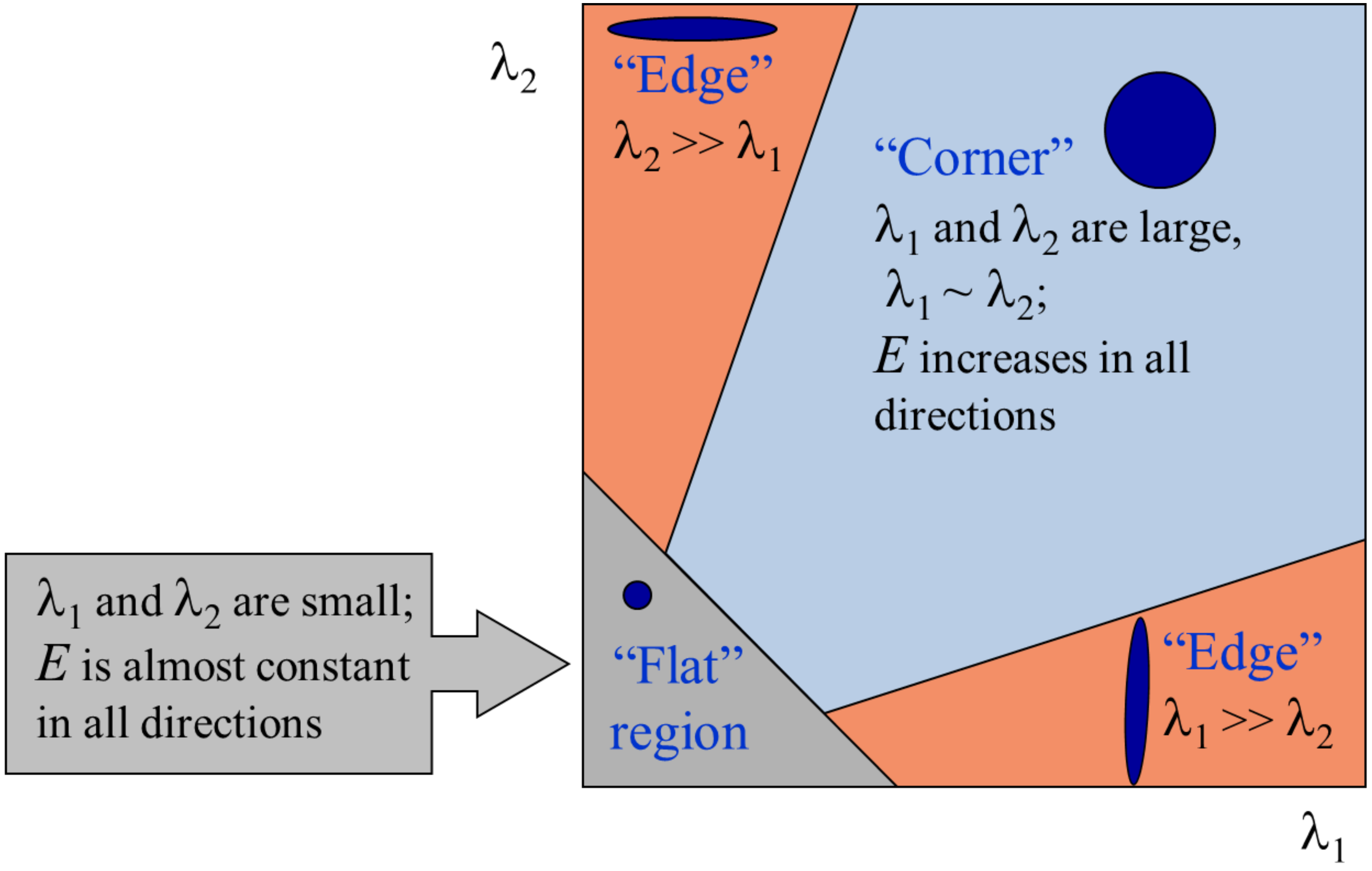
- What is our feature scoring function?

Want $E(u,v)$ to be large for small shifts in all directions

- the minimum of $E(u,v)$ should be large, over all unit vectors $[u\ v]$
- this minimum is given by the smaller eigenvalue (λ_{min}) of M

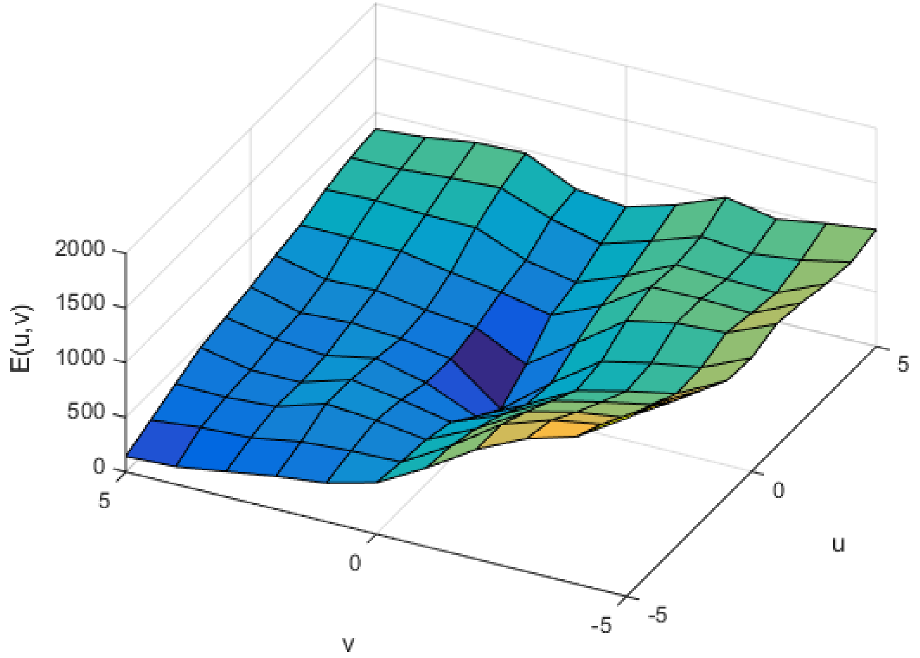


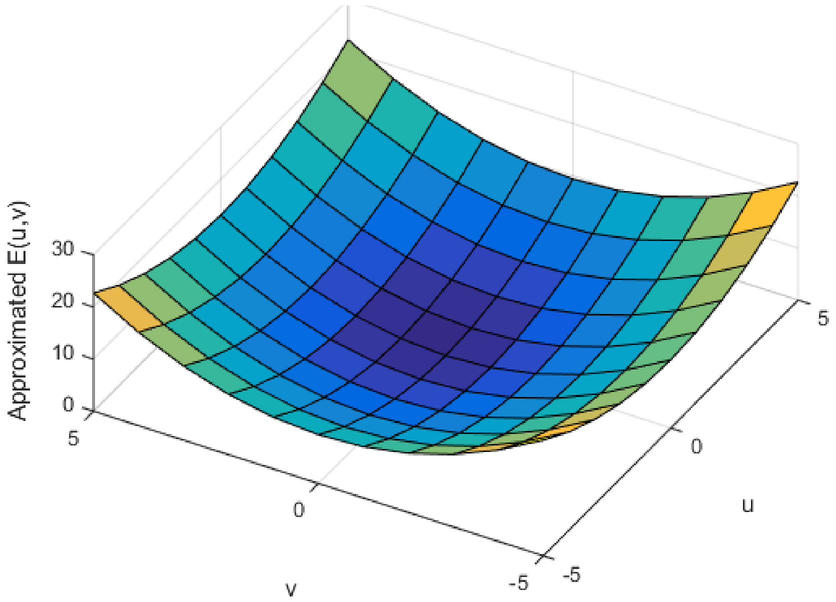
Local measure of feature distinctiveness



Examples from Holmenkollen

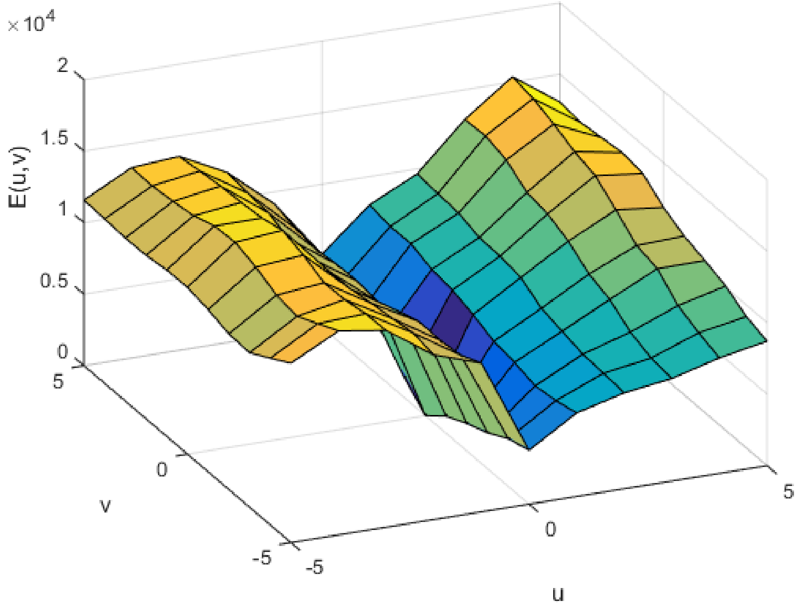
$\lambda_{min} = 0.4$

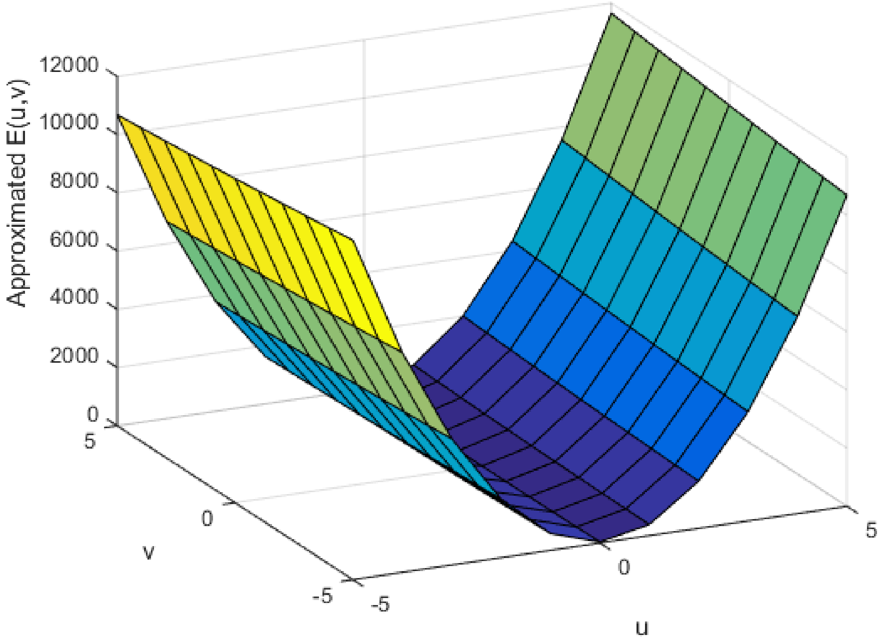




Examples from Holmenkollen

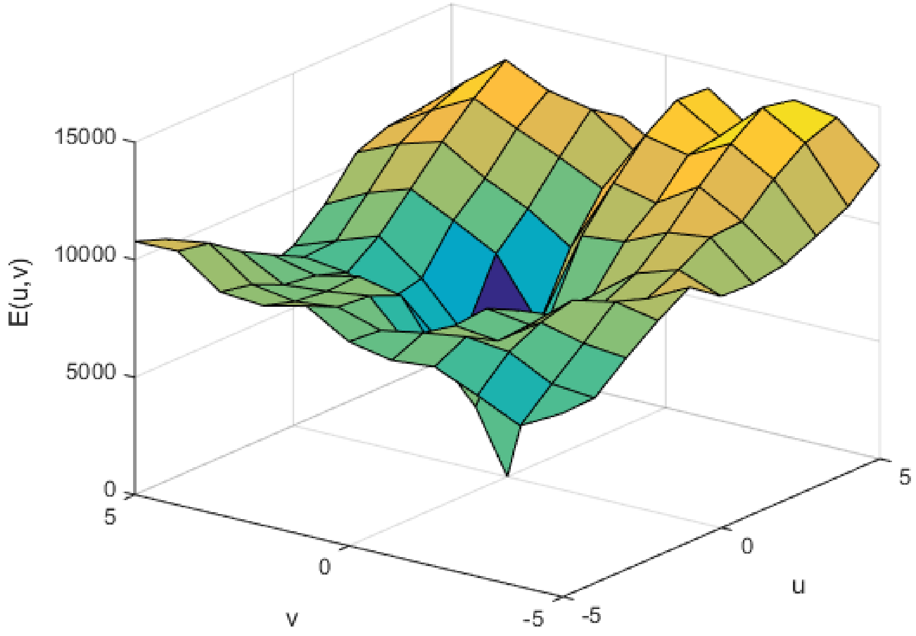
$\lambda_{min} = 1.2$

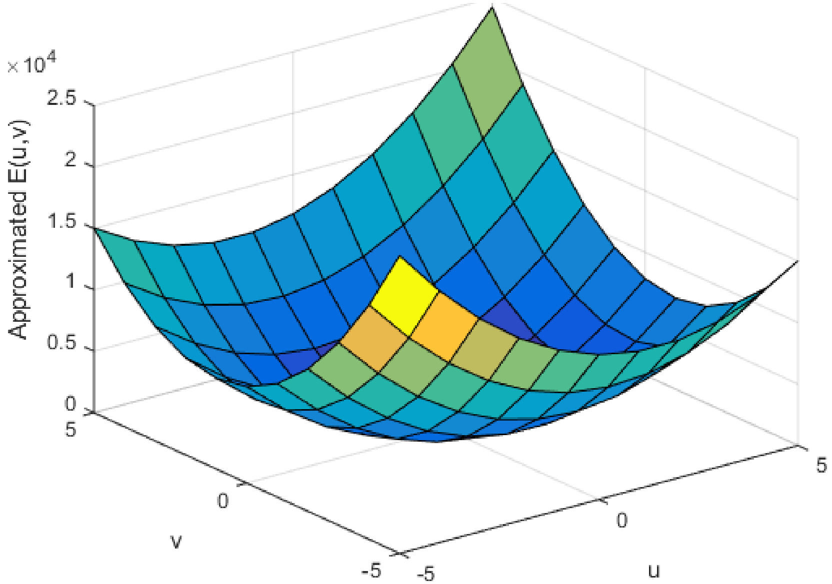




Examples from Holmenkollen

$\lambda_{min} = 272$





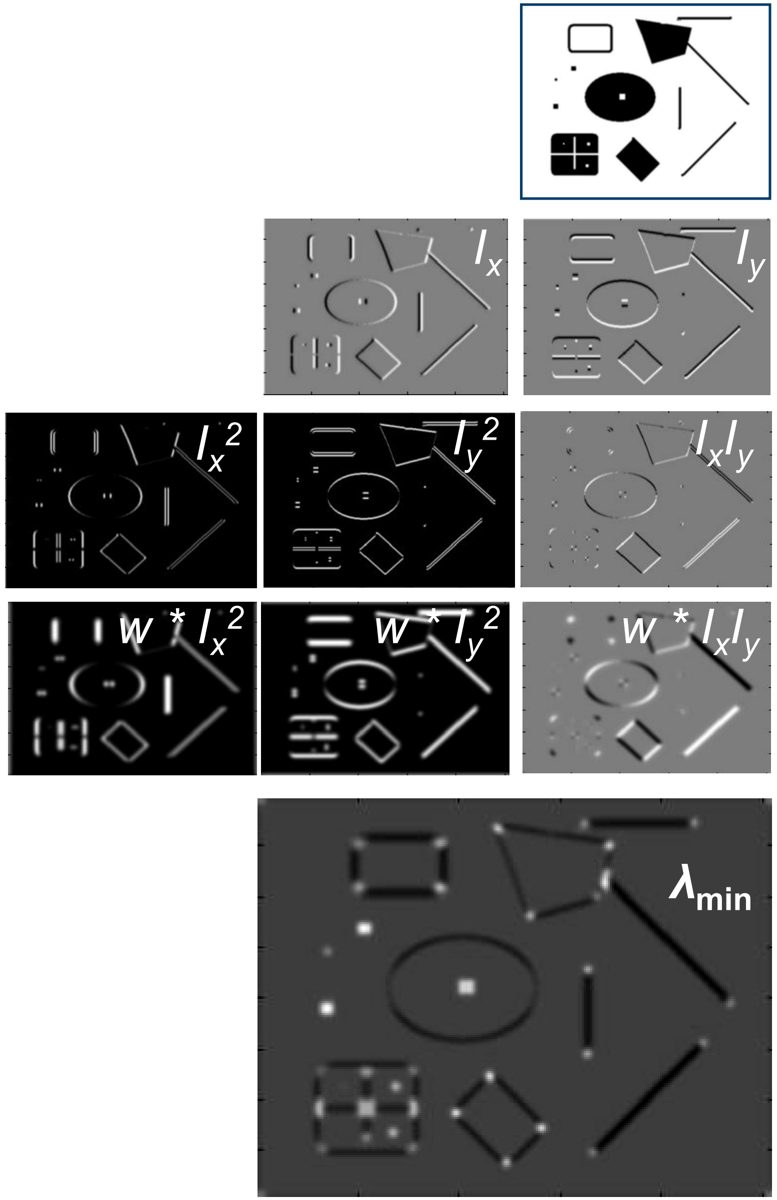
This is more efficient than you think

- Compute the gradient *images*
- Compute the elements in M as three *images* *A*, *B* and *C*

$$M = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} A & B \\ B & C \end{bmatrix}$$

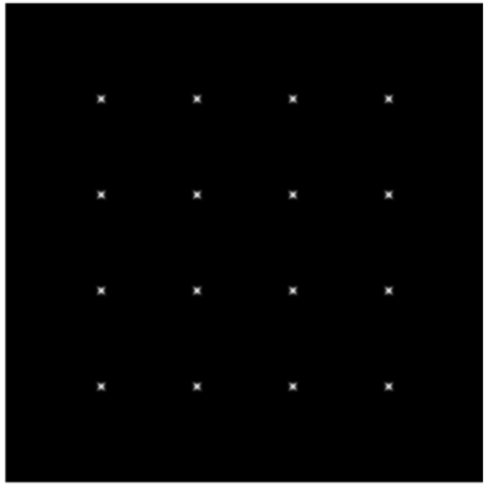
- Compute the *image* of smallest eigenvalues

$$\lambda = \frac{1}{2}\left[(A+C) - \sqrt{4B^2 + \left(A-C\right)^2} \right]$$

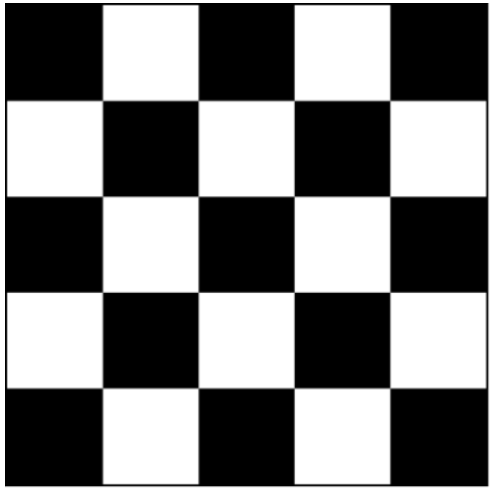


Putting together a *corner detector*

1. Compute the gradient image (using derivatives of Gaussians)
2. Compute the elements of *M* from the gradient image
3. Compute the smallest eigenvalues from the elements of *M*
4. Find points with large response (λ_{min} > threshold)



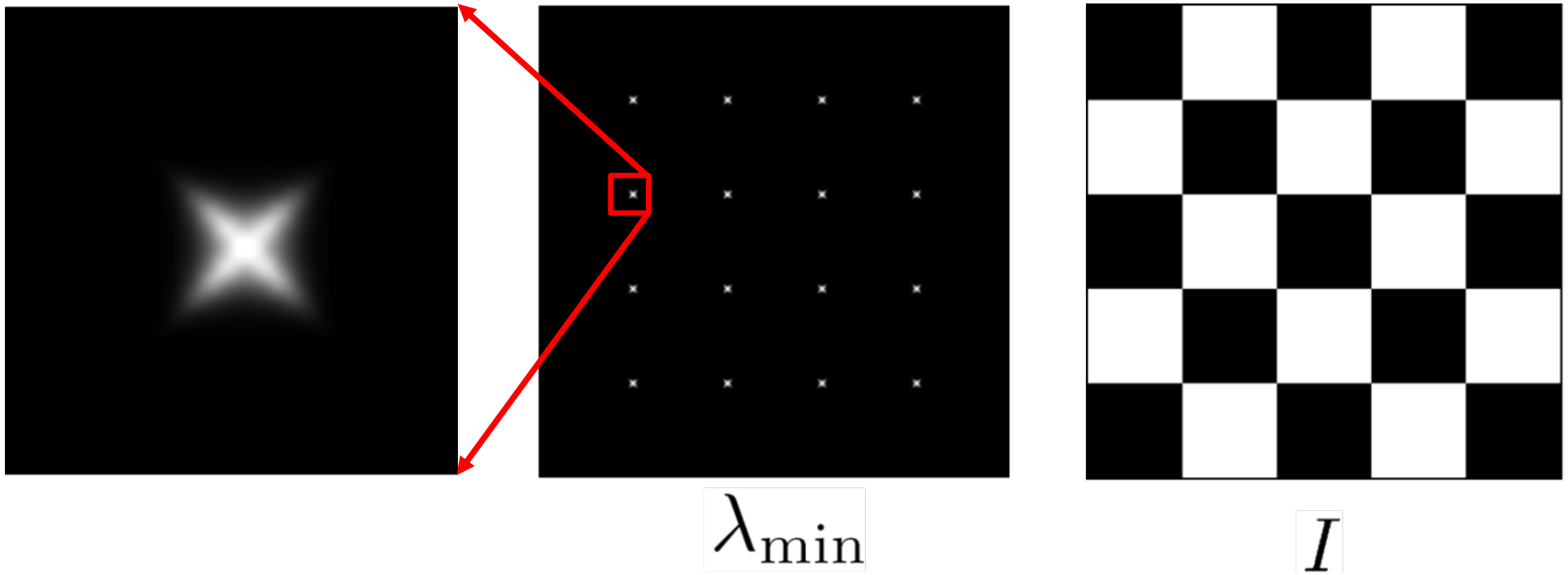
λ_{min}



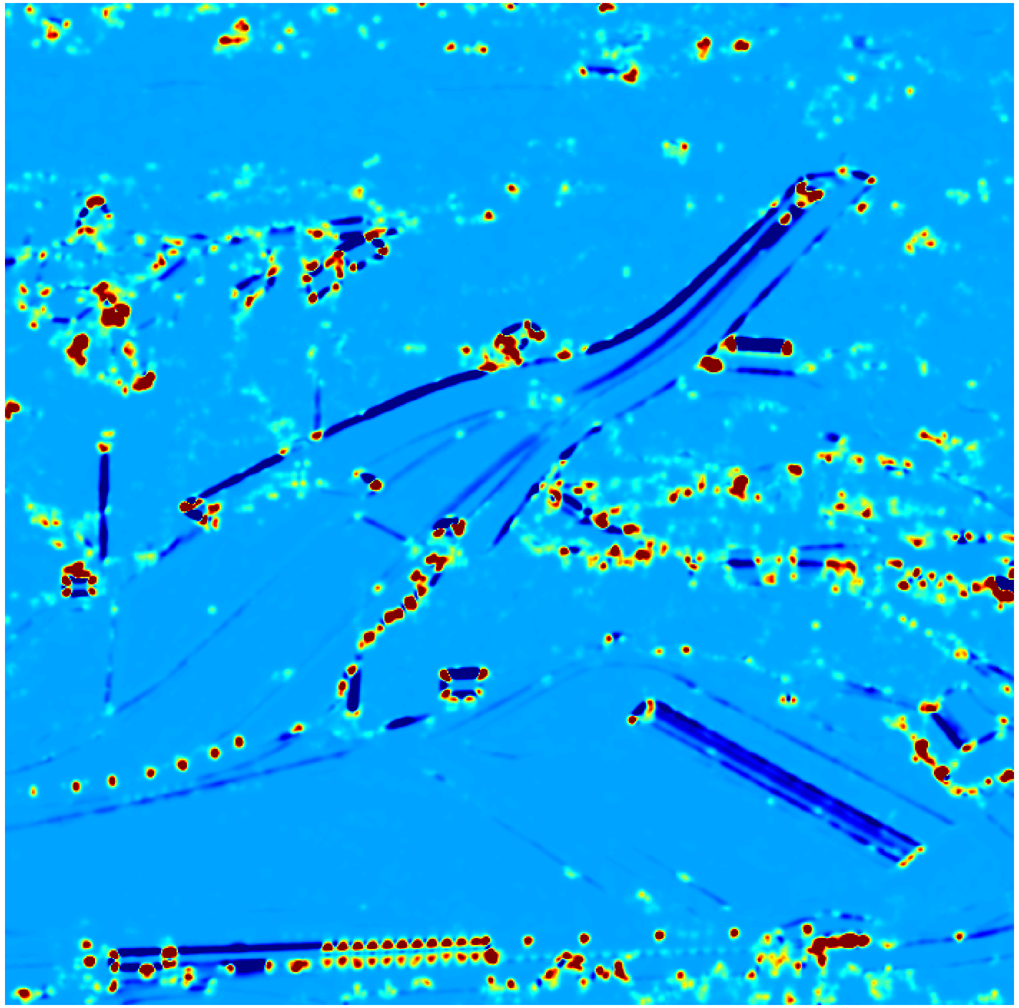
I

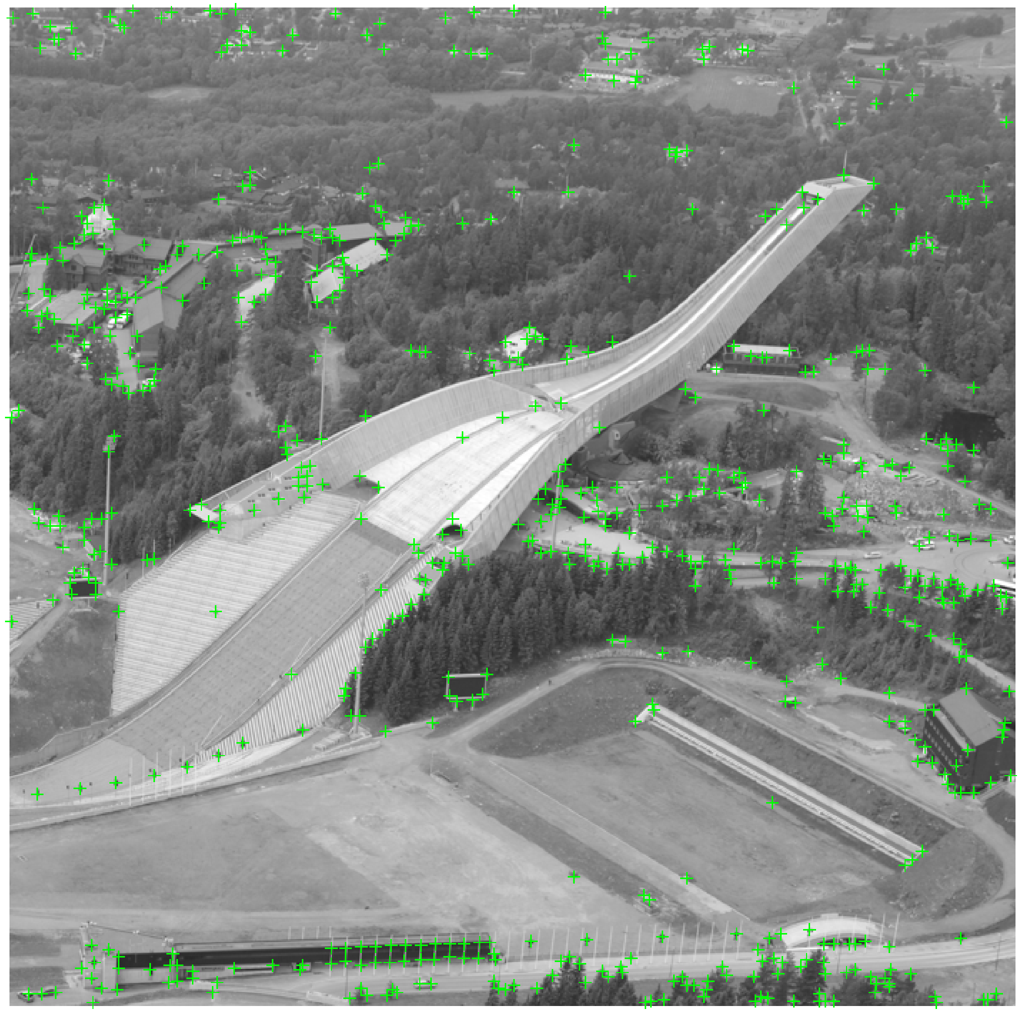
Putting together a *corner detector*

1. Compute the gradient image (using derivatives of Gaussians)
2. Compute the elements of *M* from the gradient image
3. Compute the smallest eigenvalues from the elements of *M*
4. Find points with large response (λ_{min} > threshold)
5. Choose points where λ_{min} is a local maximum as features



Example from Holmenkollen





The Harris operator

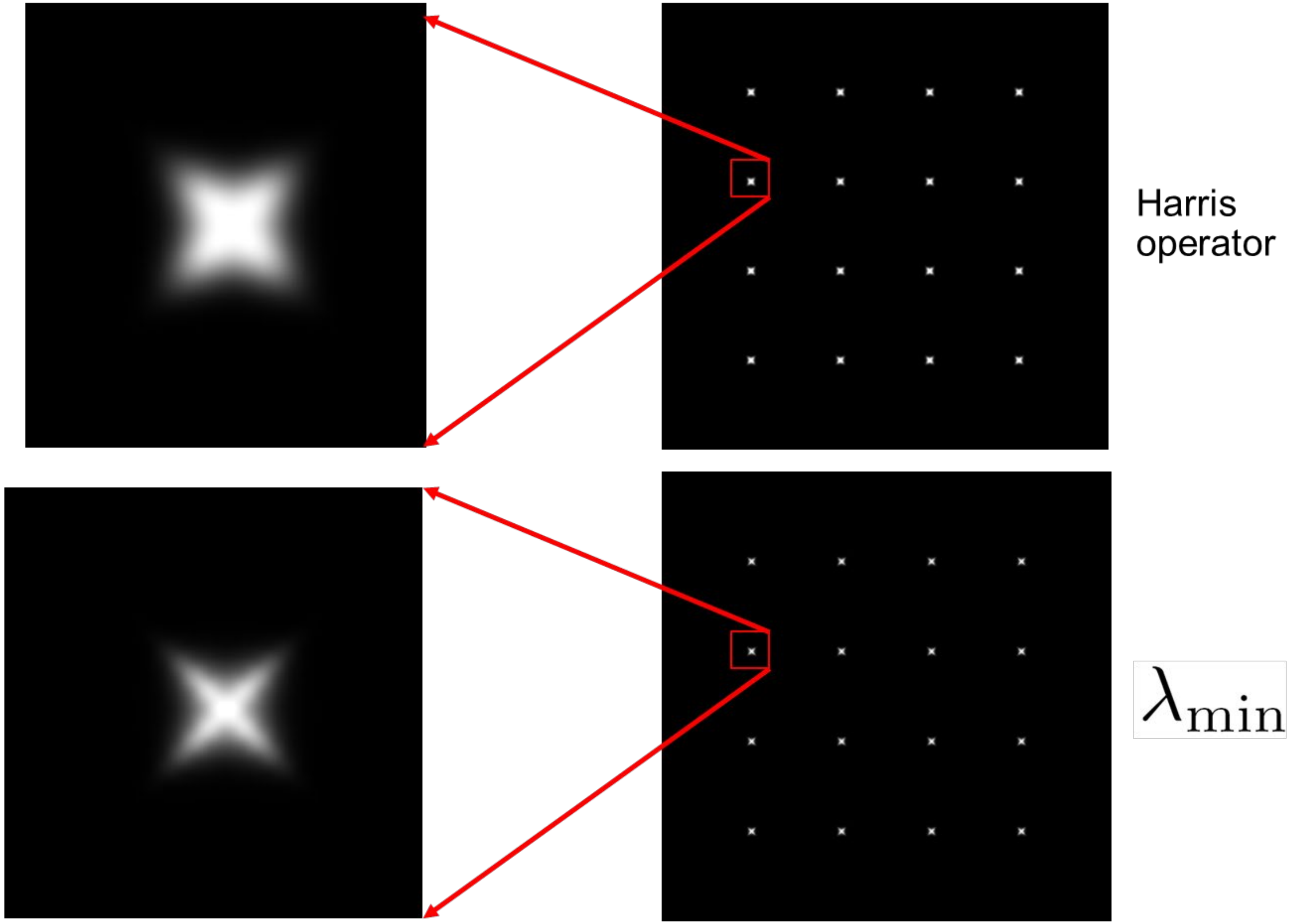
A more efficient alternative to λ_{min}:

$$f = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2 = \det(M) - \alpha \operatorname{trace}(M)^2$$

- $\alpha = 0.06$

- Very similar to λ_{min} but less expensive (no square root)
- Called the “Harris Corner Detector” or “Harris Operator”

The Harris operator



The harmonic mean

A more efficient alternative to $\lambda_{\min}$:

$$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} = \frac{\det(M)}{\mathrm{trace}(M)}$$

- Smoother in the region where $\lambda_1 \approx \lambda_2$

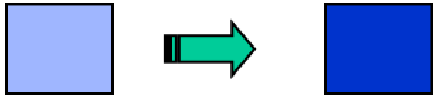
Invariance and covariance

We want corner locations to be *invariant* to photometric transformations and *covariant* to geometric transformations

- **Invariance**: image is transformed and corner locations do not change
- **Covariance**: if we have two transformed versions of the same image, features should be detected in corresponding locations

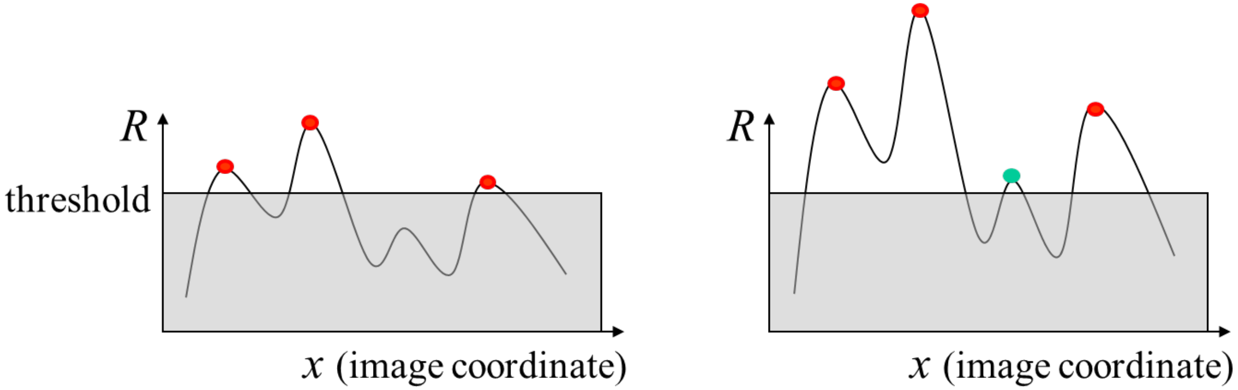
Corner detector properties

Affine intensity change



$$I \rightarrow a\,I + b$$

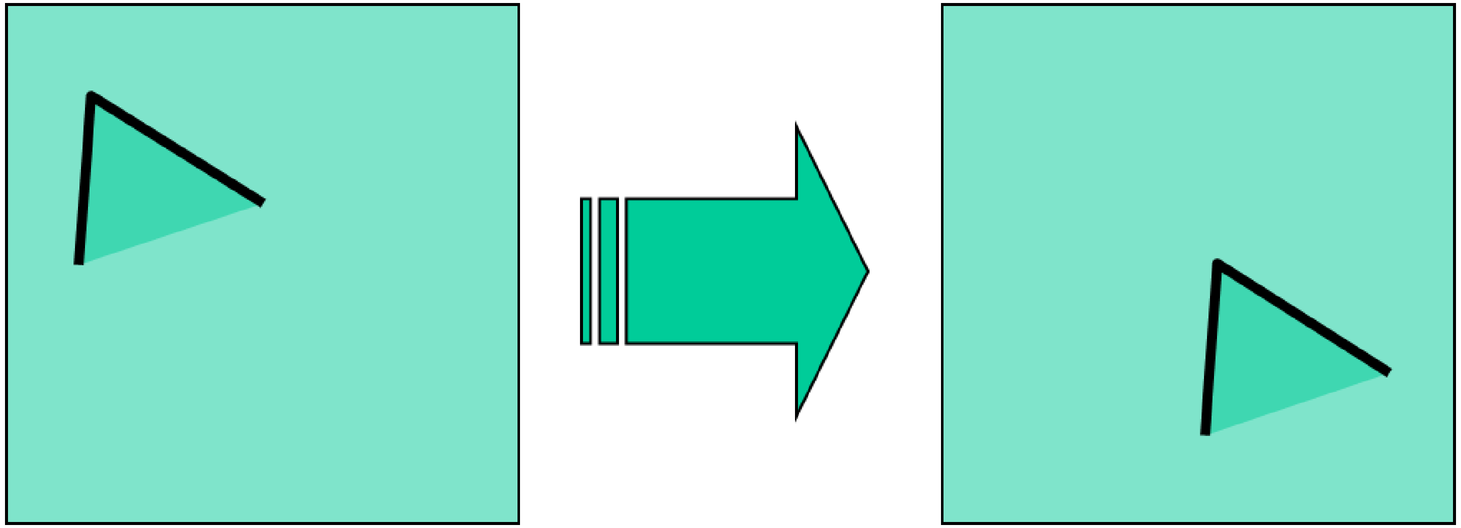
- Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$
- Intensity scaling: $I \rightarrow a\,I$



Partially invariant **to affine intensity change**

Corner detector properties

Image translation

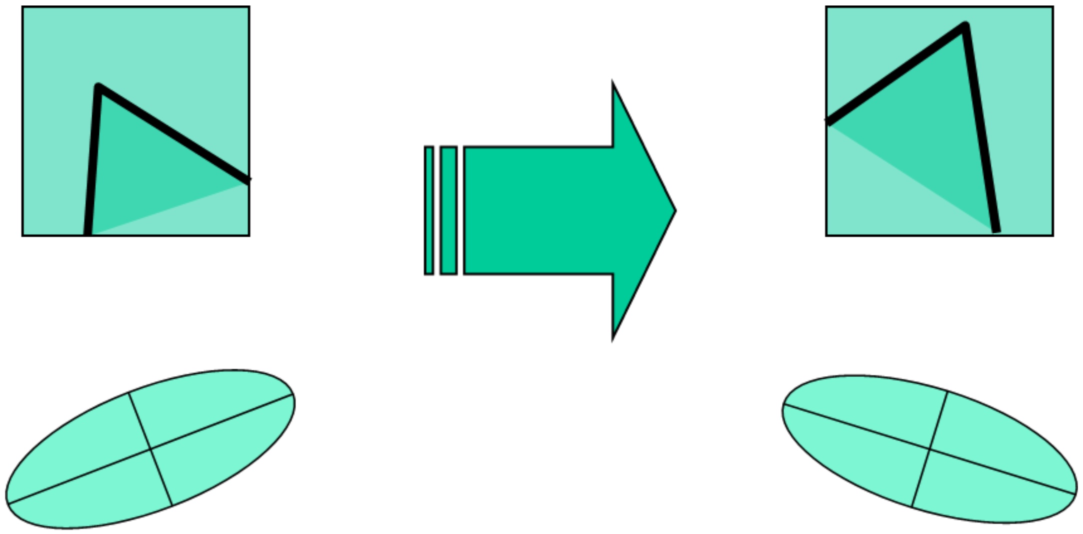


- Derivatives and window function are shift-invariant

Corner location is covariant w.r.t. translation

Corner detector properties

Image rotation

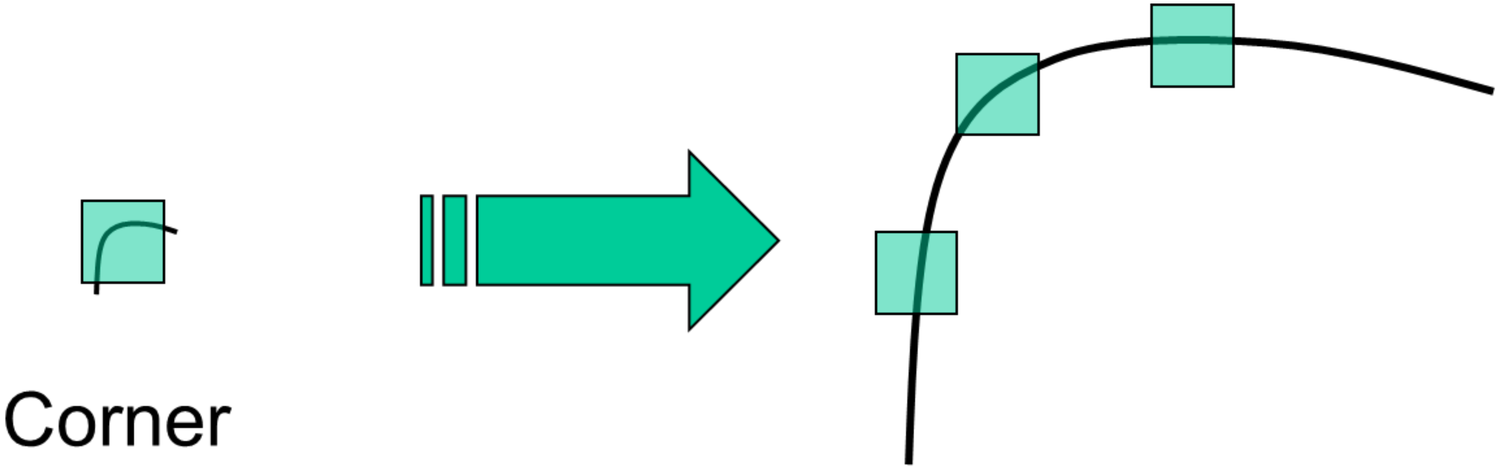


- Ellipse rotates but its shape (i.e. eigenvalues) remains the same

Corner location is covariant w.r.t. rotation

Corner detector properties

Scaling



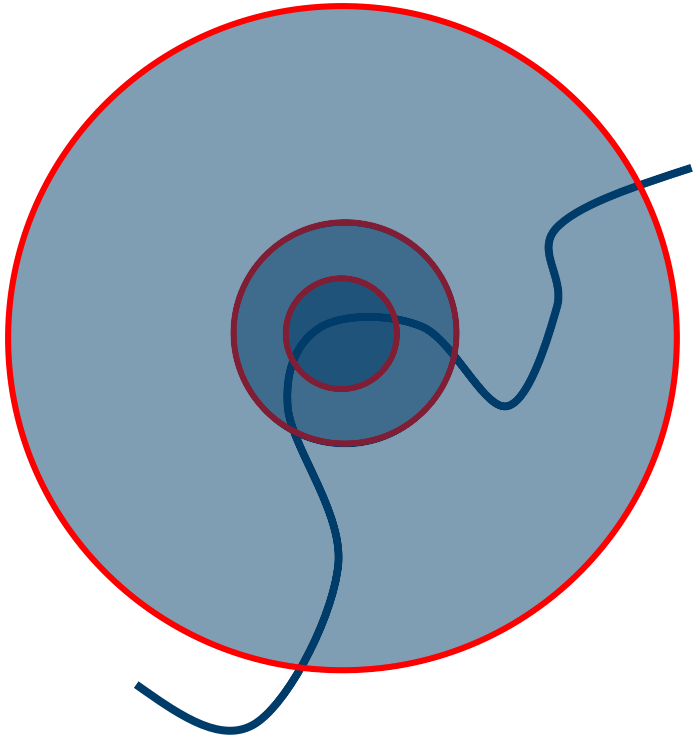
All points will be classified as edges

Corner location is not covariant to scaling!

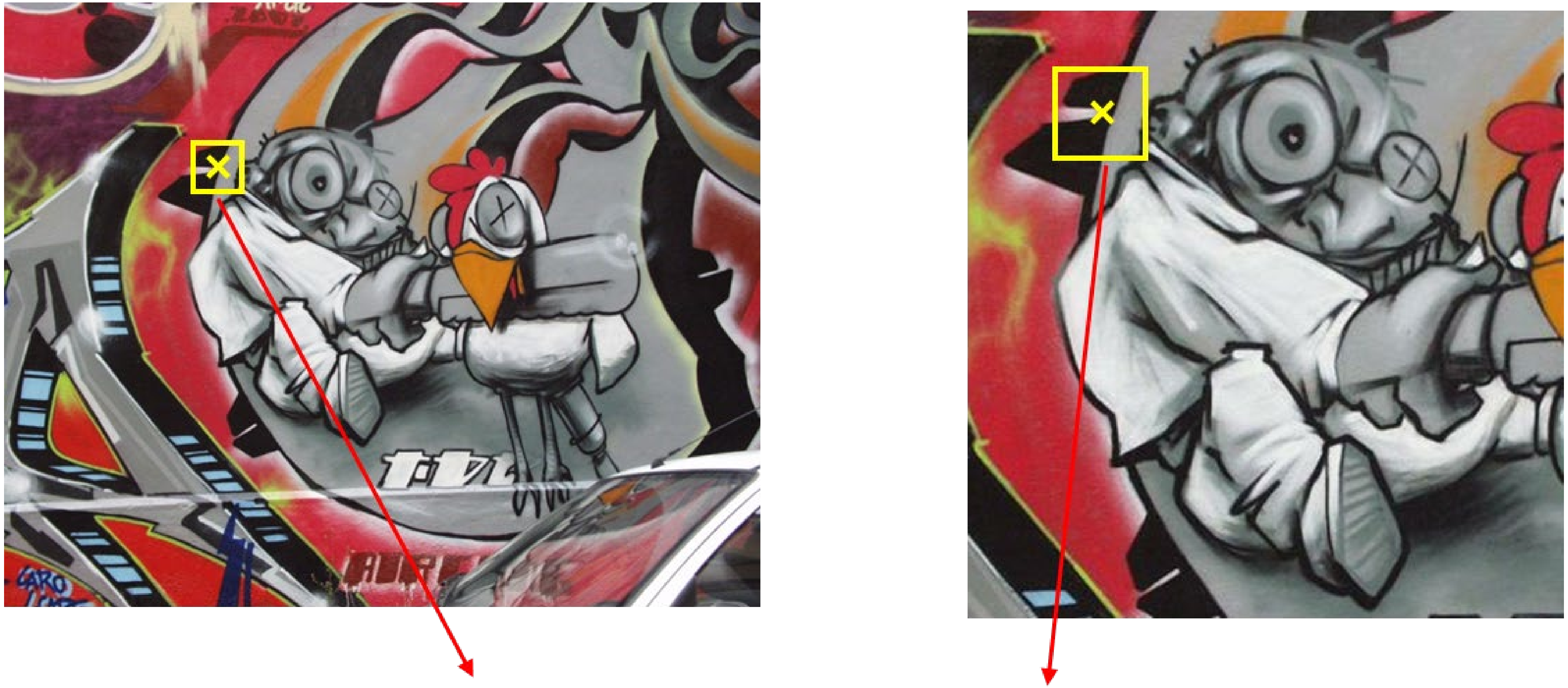
Scale robust corner detection

Find scale that gives local maximum of score *f*

- In both position and scale

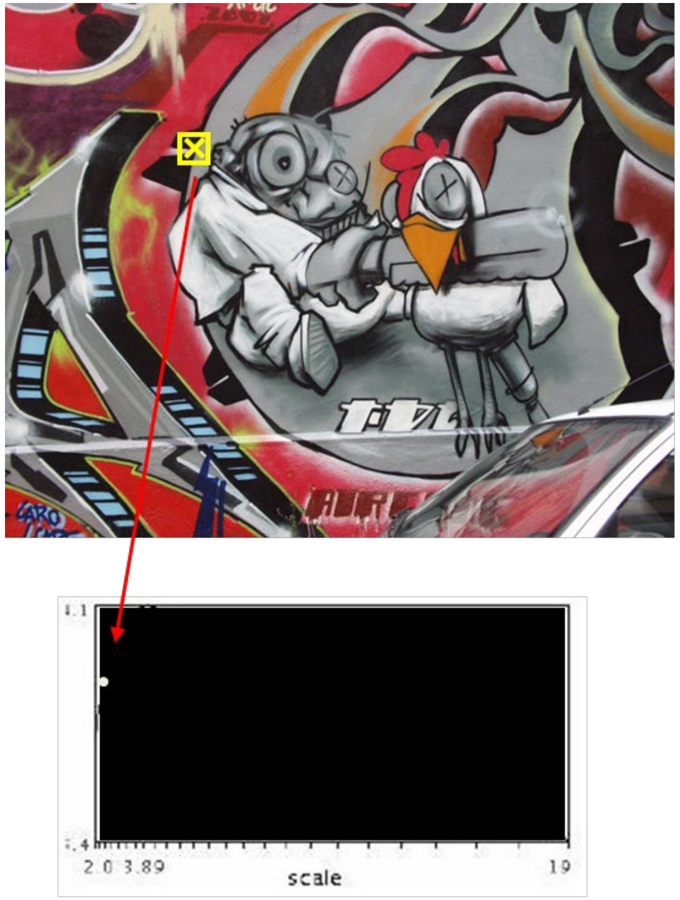


Automatic scale selection



$$f(I_{i_1 \ldots i_m}(x, \sigma)) = f(I_{i_1 \ldots i_m}(x', \sigma'))$$

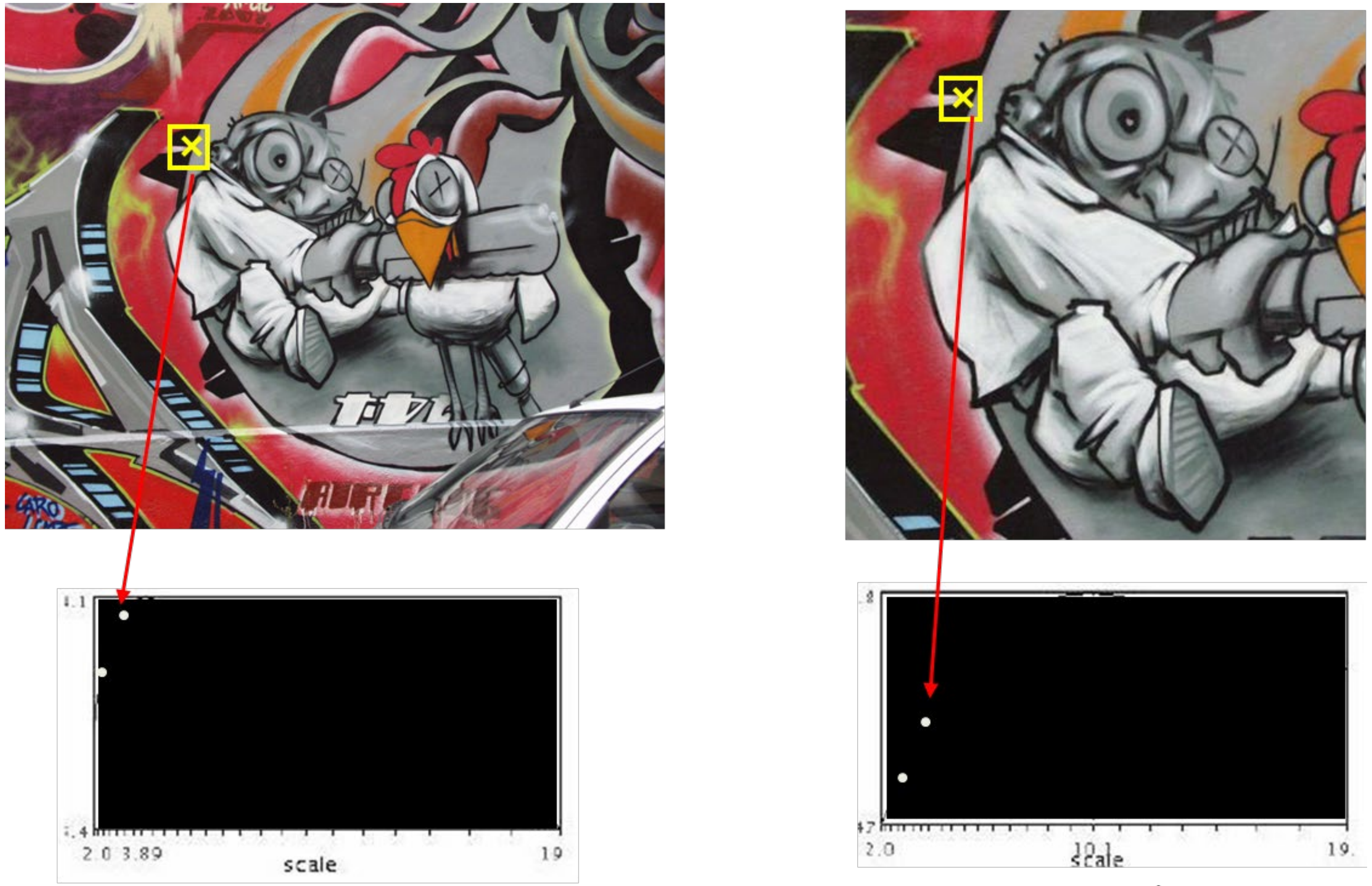
Automatic scale selection



$f(I_{i_1 \dots i_m}(x, \sigma))$

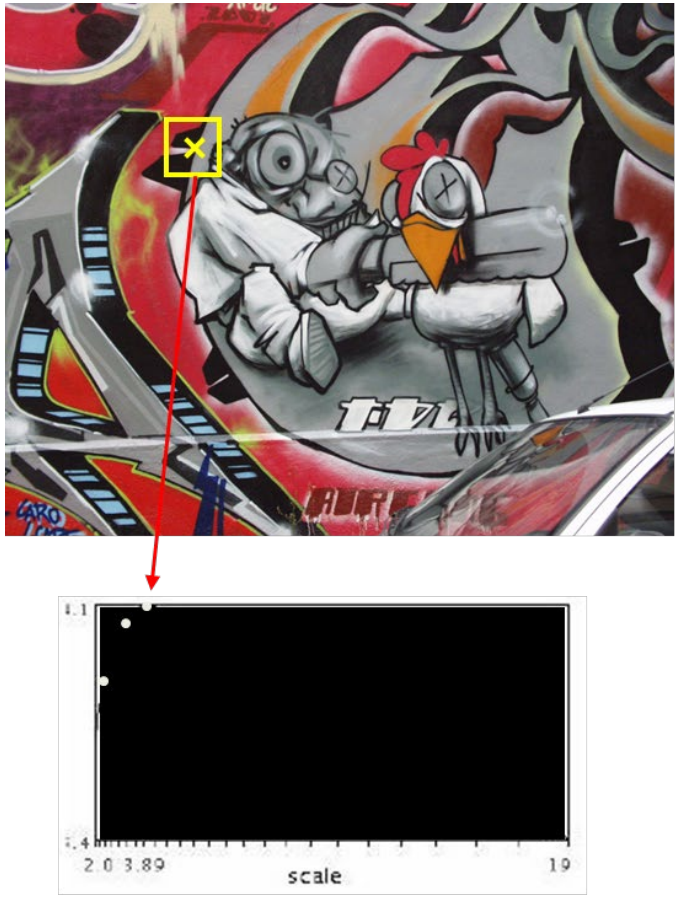
$f(I_{i_1 \dots i_m}(x', \sigma))$

Automatic scale selection



$f(I_{i_1\ldots i_m}(x,\sigma))$ $f(I_{i_1\ldots i_m}(x',\sigma))$

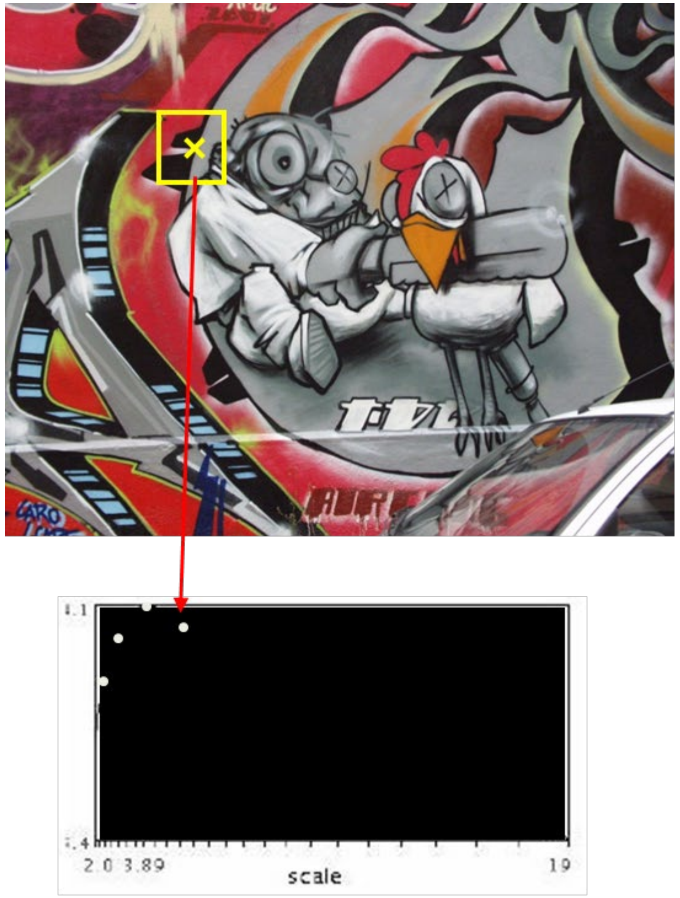
Automatic scale selection



$f(I_{i_1 \ldots i_m}(x, \sigma))$

$f(I_{i_1 \ldots i_m}(x', \sigma))$

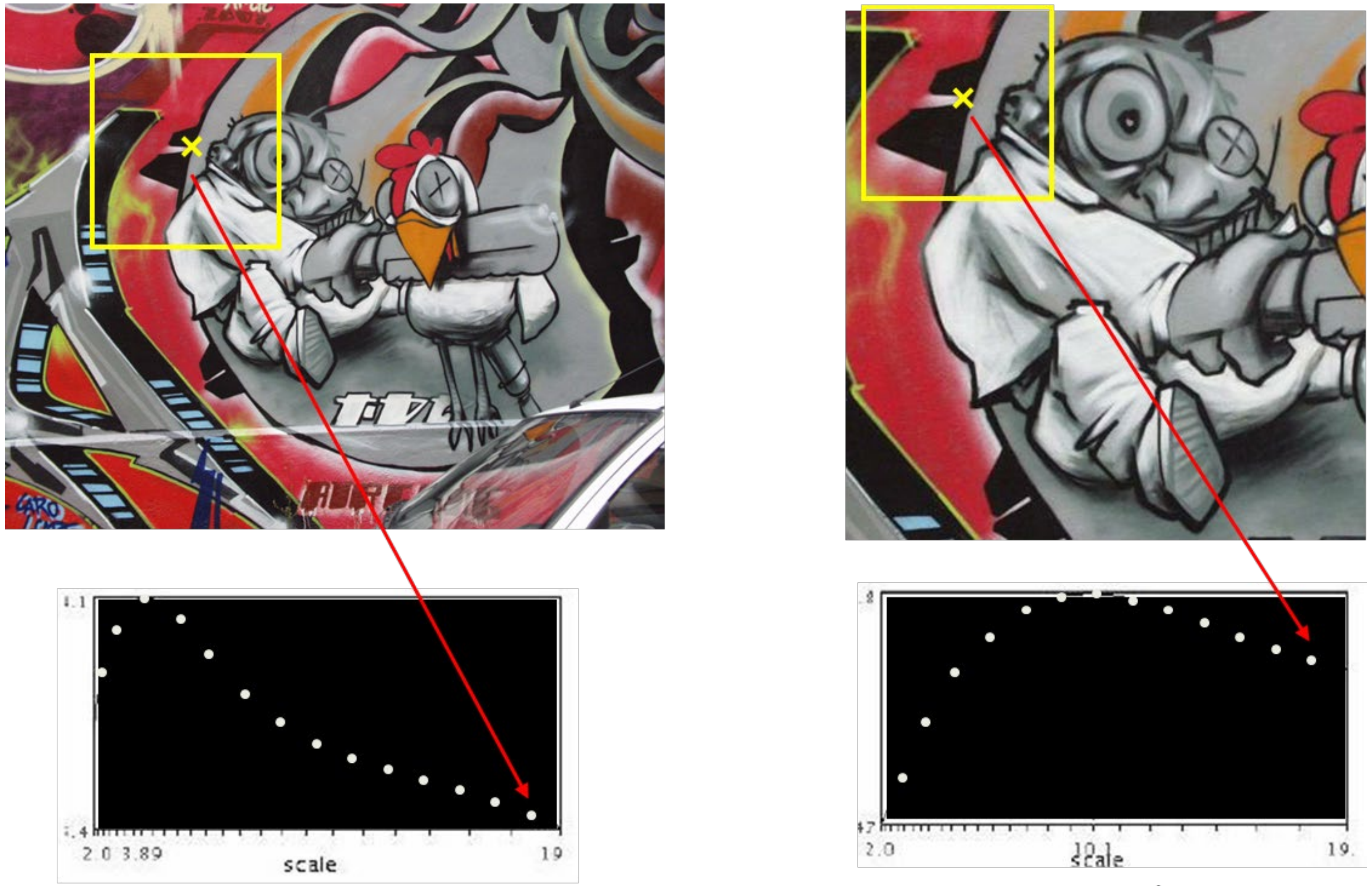
Automatic scale selection



$$f(I_{i_1 \dots i_m}(x, \sigma))$$

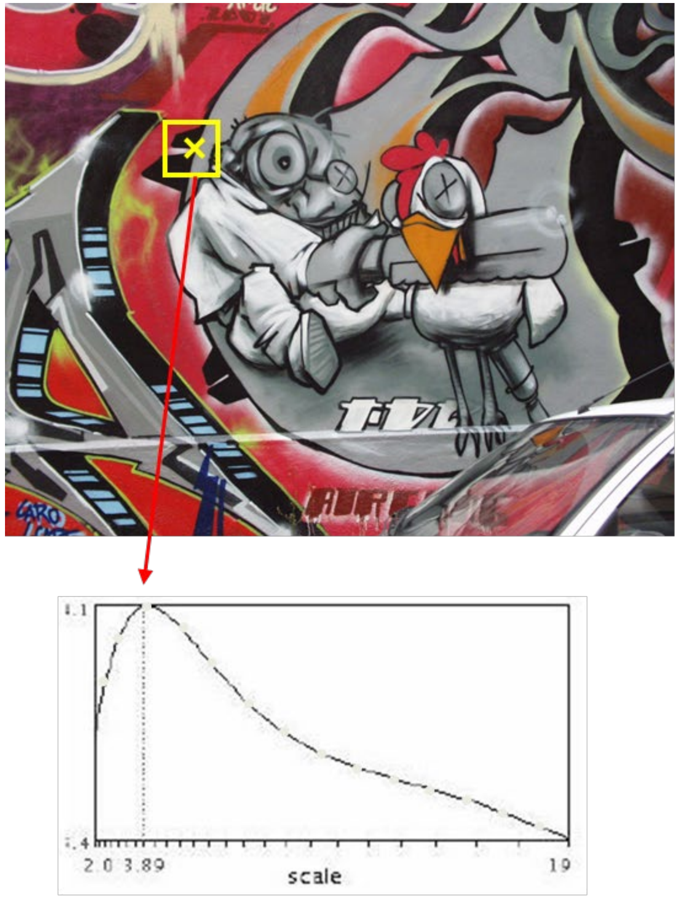
$$f(I_{i_1 \dots i_m}(x', \sigma))$$

Automatic scale selection

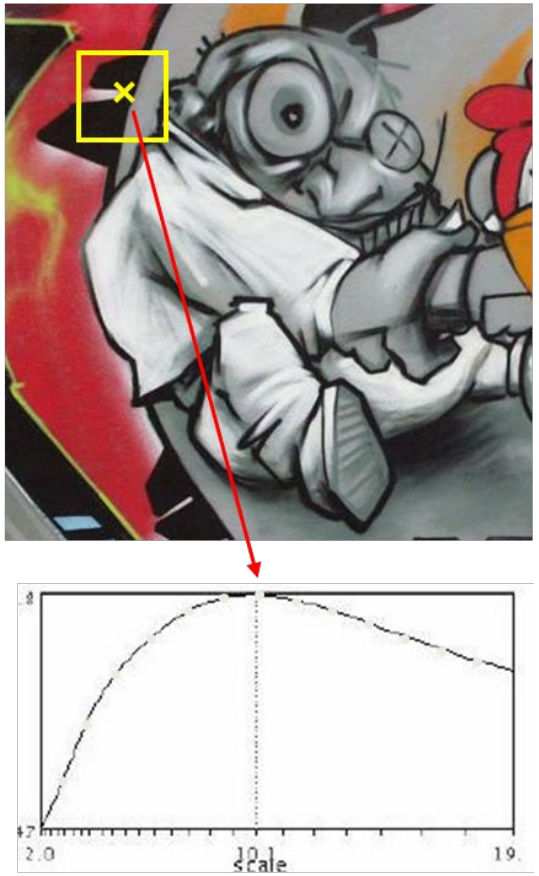


$f(I_{i_1 \ldots i_m}(x,\sigma))$ $f(I_{i_1 \ldots i_m}(x',\sigma))$

Automatic scale selection



$$f(I_{i_1 \ldots i_m}(x, \sigma))$$



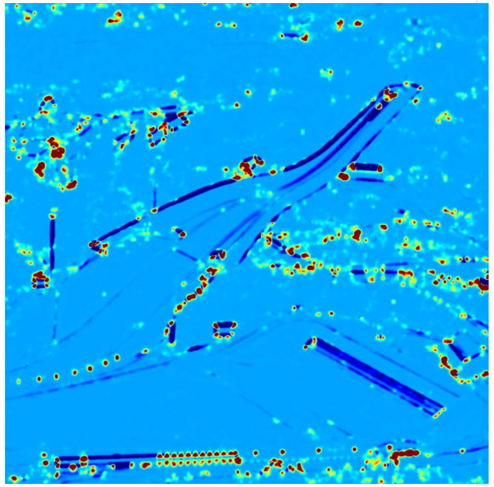
$$f(I_{i_1 \ldots i_m}(x', \sigma'))$$

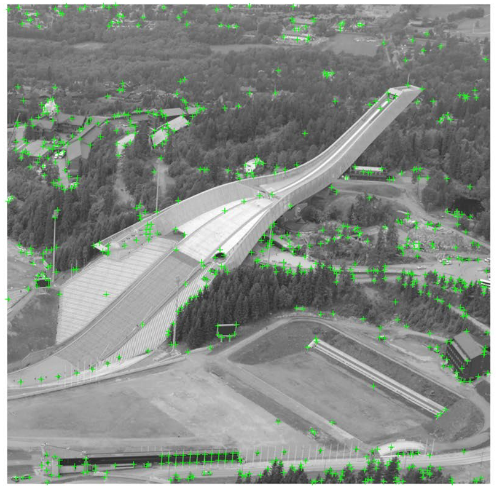
Summary

- From *characteristics of good features* to a *practical corner detector*!
  - Go back and evaluate our resulting detector!

- Corner detector properties

- We will implement this corner detector in the next lab!





- Next lecture:
  - Blob detectors – Distinct in space *and* scale