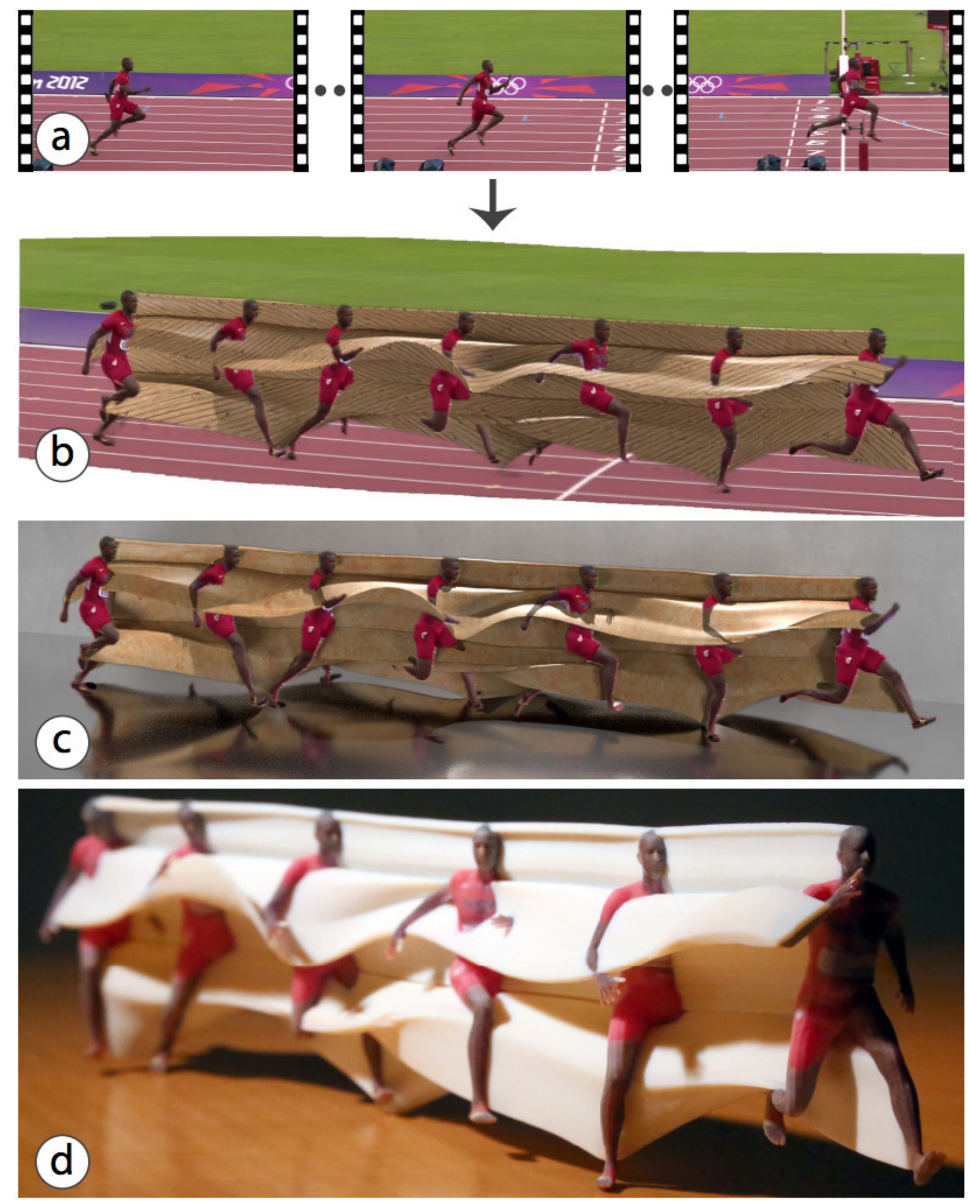
Feature Descriptors

Computer Vision
Fall 2018
Columbia University

Tali Dekel

Tuesday, October 2, 11am, CEPSR 620

http://people.csail.mit.edu/talidekel



Seam Carving

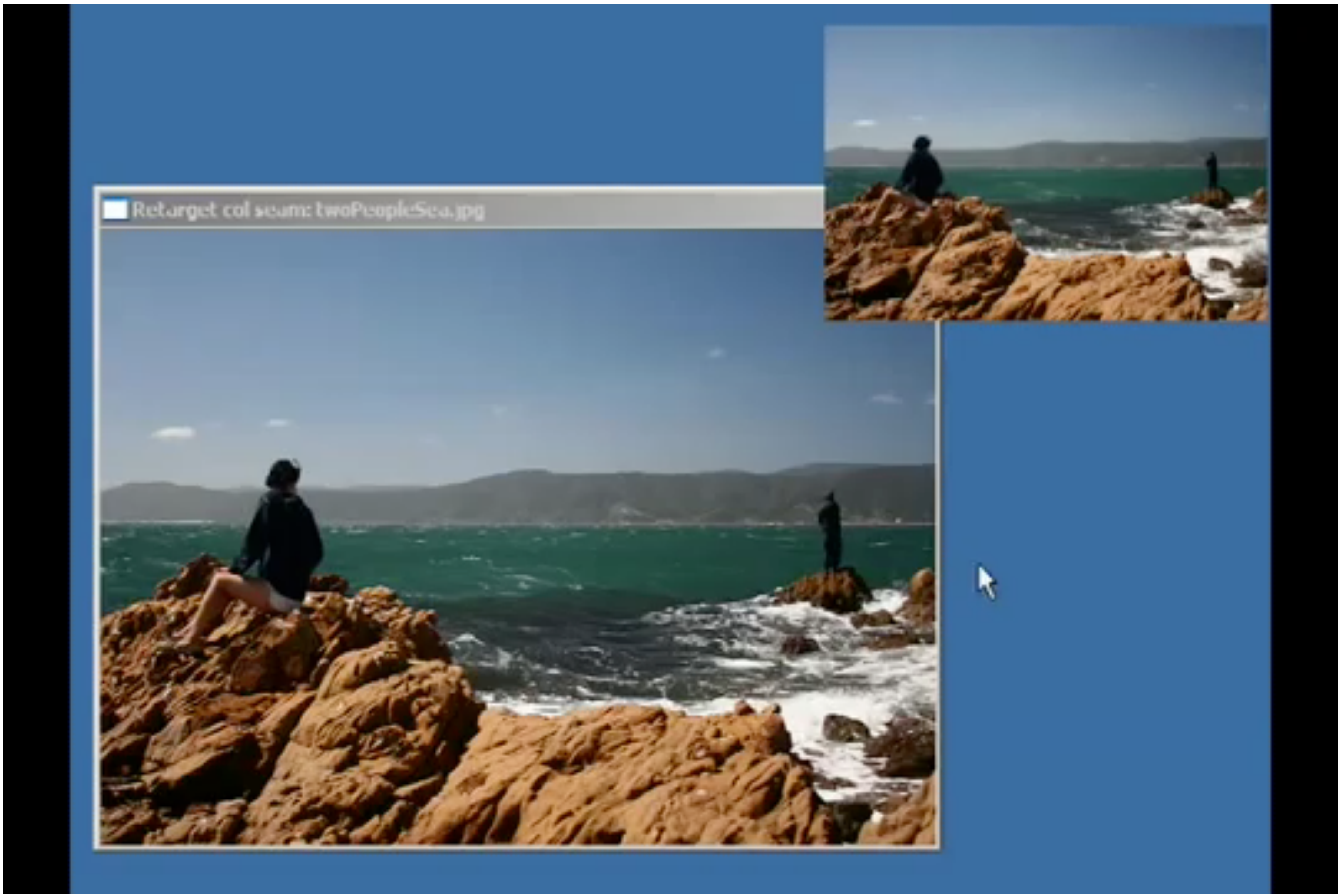
Seam carving: main idea

Content-aware resizing

Traditional resizing

[Shai & Avidan, SIGGRAPH 2007]

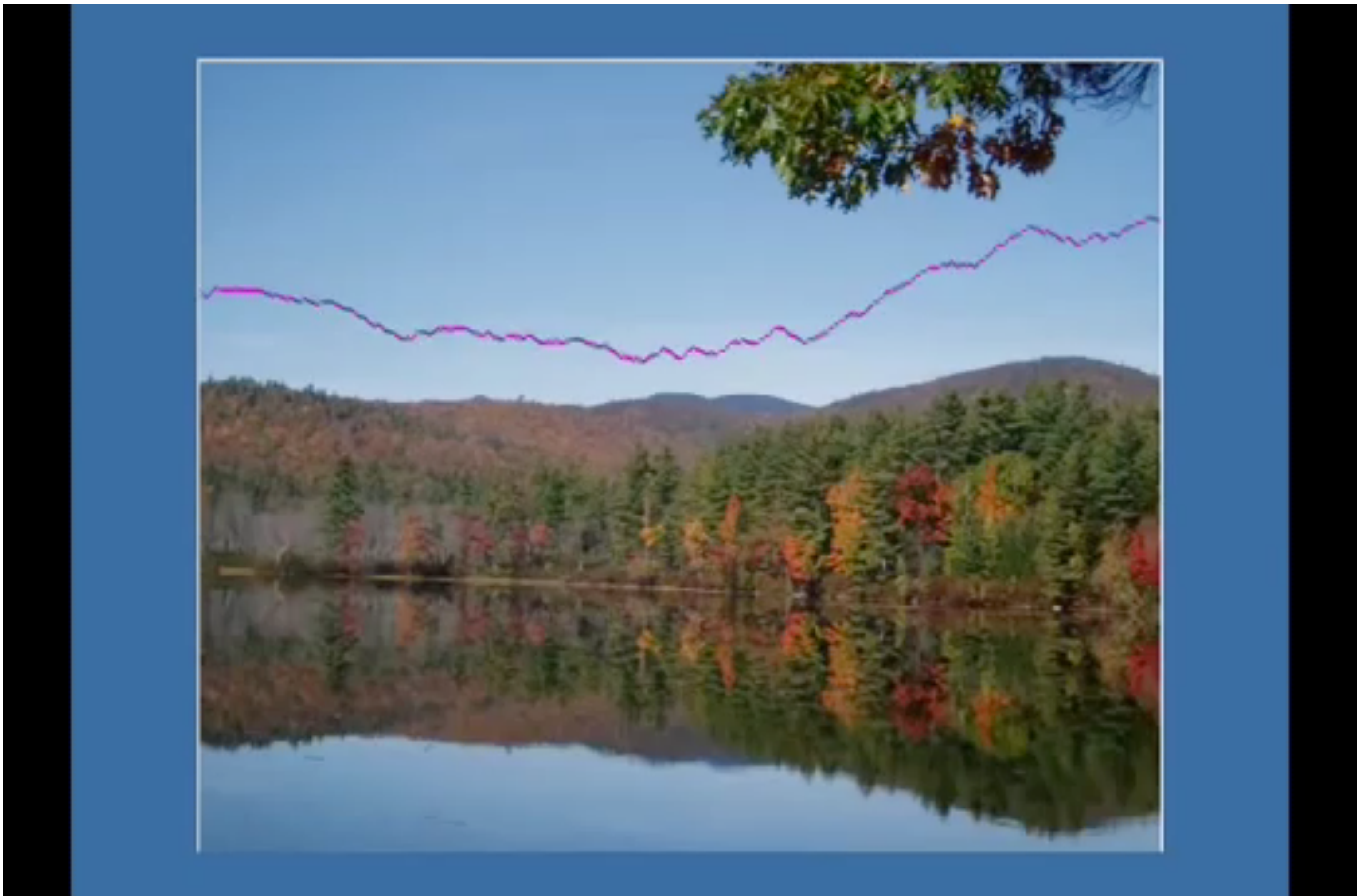
Seam Carving



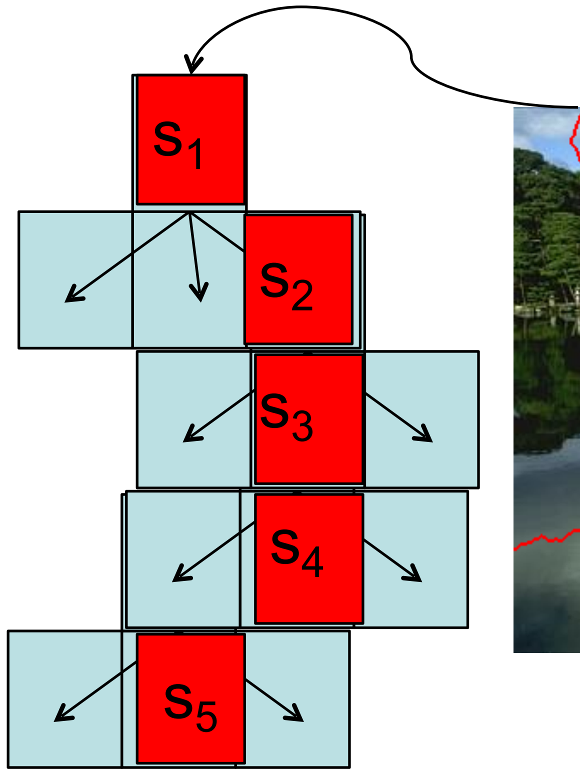
Seam carving: main idea

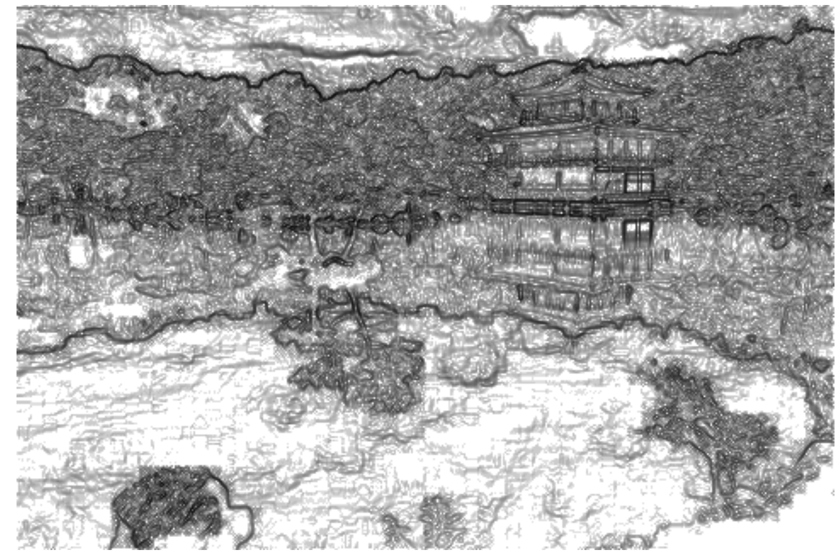
[Shai & Avidan, SIGGRAPH 2007]

Seam Carving



Seam carving: algorithm





$$Energy(f) = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Let a vertical seam **s** consist of *h* positions that form an 8-connected path.

Let the cost of a seam be:

$$Cost(\mathbf{s}) = \sum_{i=1}^{h} Energy(f(s_i))$$

Optimal seam minimizes this cost:

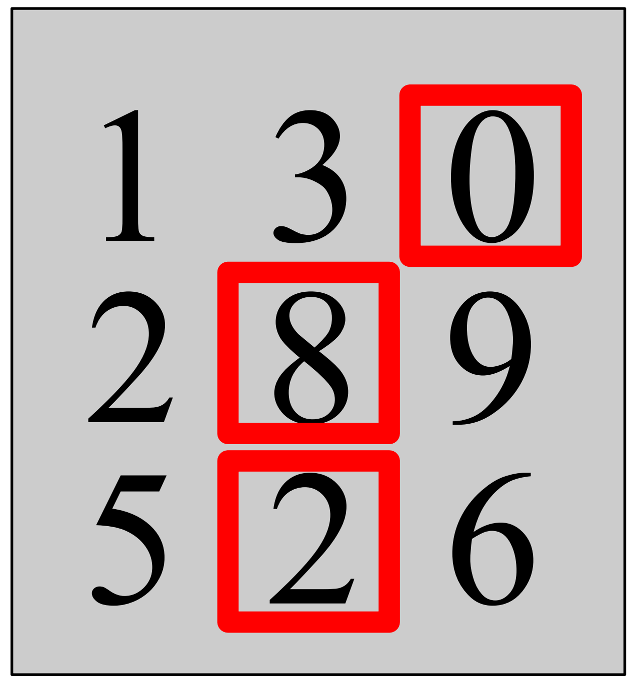
$$\mathbf{s}^* = \min_{\mathbf{s}} Cost(\mathbf{s})$$

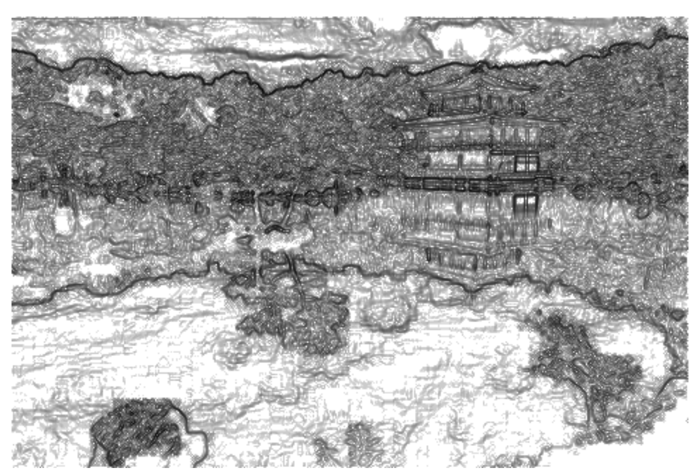
Compute it efficiently with dynamic programming.

Slide credit: Kristen Grauman

How to identify the minimum cost seam?

- First, consider a **greedy** approach:





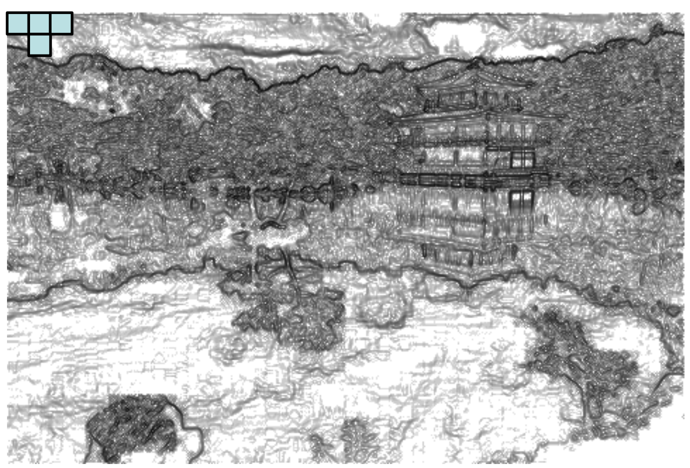
Energy matrix (gradient magnitude)

Slide credit: Kristen Grauman

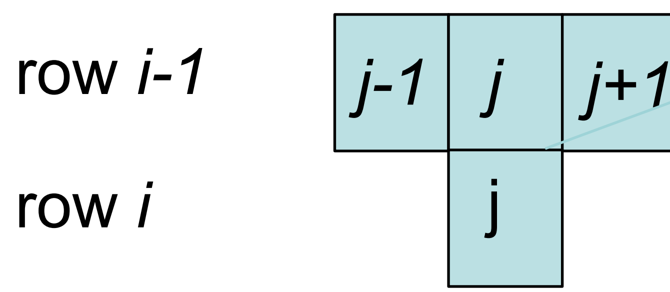
Seam carving: algorithm

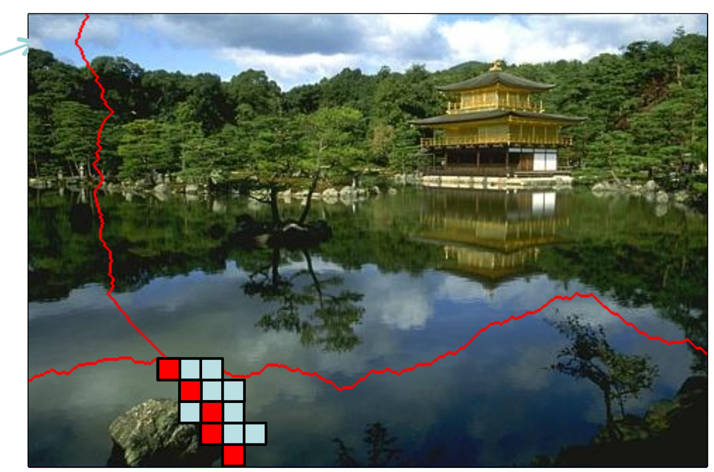
- Compute the cumulative minimum energy for all possible connected seams at each entry *(i,j)*:

$$\mathbf{M}(i,j) = Energy(i,j) + \min\big(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\big)$$



Energy matrix (gradient magnitude)





M matrix: cumulative min energy (for vertical seams)

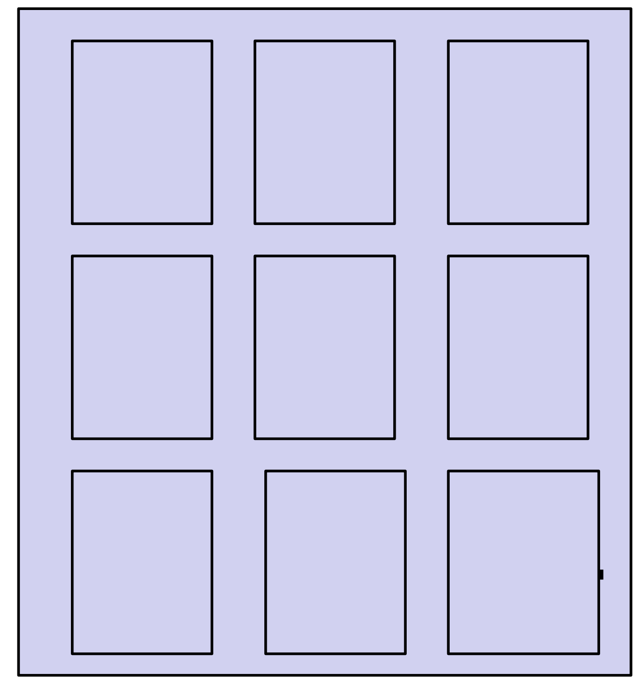
- Then, min value in last row of **M** indicates end of the minimal connected vertical seam.
- Backtrack up from there, selecting min of 3 above in **M**.

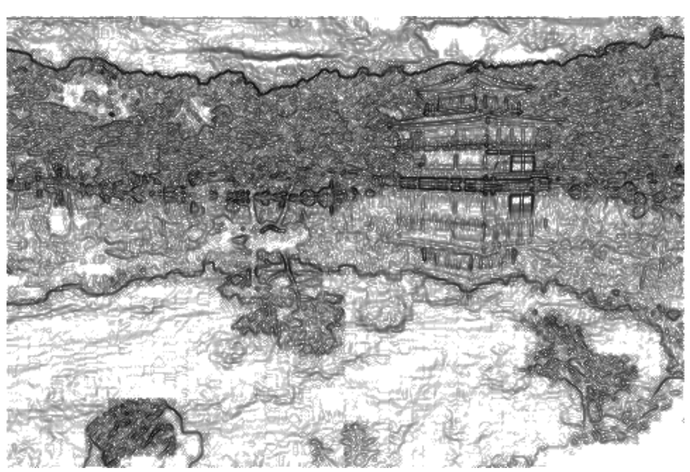
Slide credit: Kristen Grauman

Example

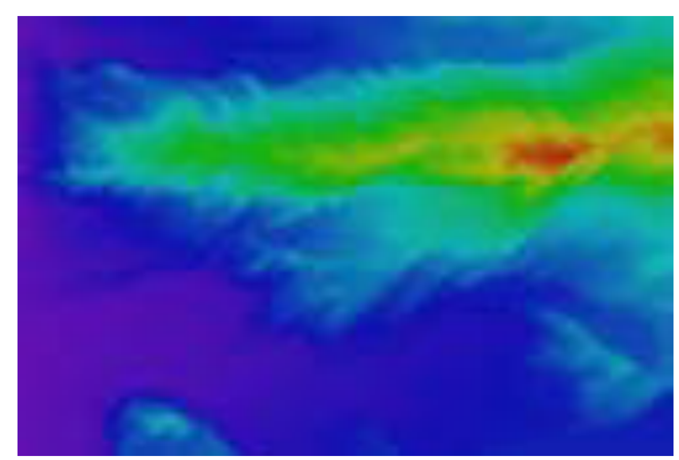
$$\mathbf{M}(i,j) = Energy(i,j) + \min\big(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\big)$$

1	3	0
2	8	9
5	2	6





Energy matrix (gradient magnitude)



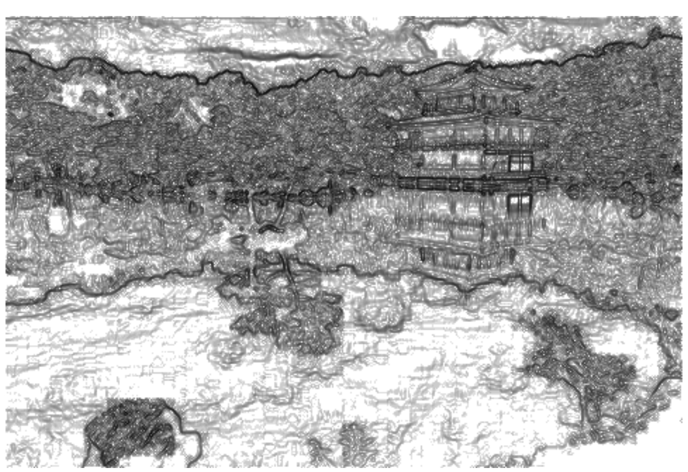
M matrix (for vertical seams)

Slide credit: Kristen Grauman

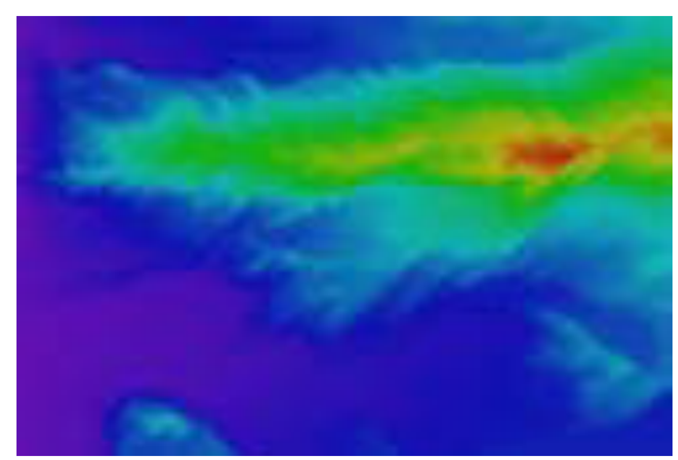
Example

$$\mathbf{M}(i, j) = Energy(i, j) + \min\big(\mathbf{M}(i-1, j-1), \mathbf{M}(i-1, j), \mathbf{M}(i-1, j+1)\big)$$

$$\begin{matrix} \boxed{1} & 3 & 0 \\ \boxed{2} & 8 & 9 \\ 5 & \boxed{2} & 6 \end{matrix} \qquad \begin{matrix} \boxed{1} & 3 & 0 \\ \boxed{3} & 8 & 9 \\ 8 & \boxed{5} & 14 \end{matrix}$$



Energy matrix
(gradient magnitude)



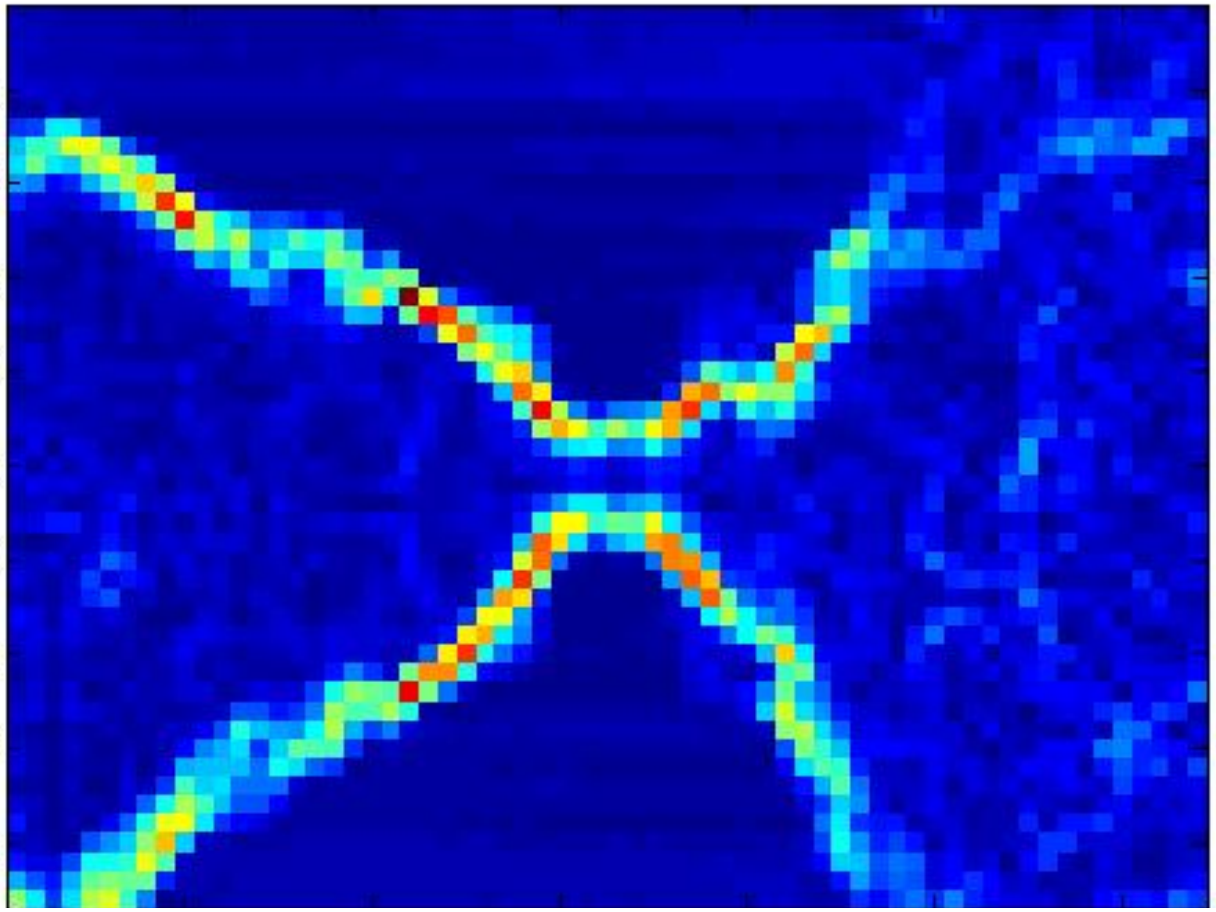
M matrix
(for vertical seams)

Slide credit: Kristen Grauman

Real image example

Original Image

Energy Map



Blue = low energy
Red = high energy

Slide credit: Kristen Grauman

Seam Carving

Why did it fail?

Original

Resized

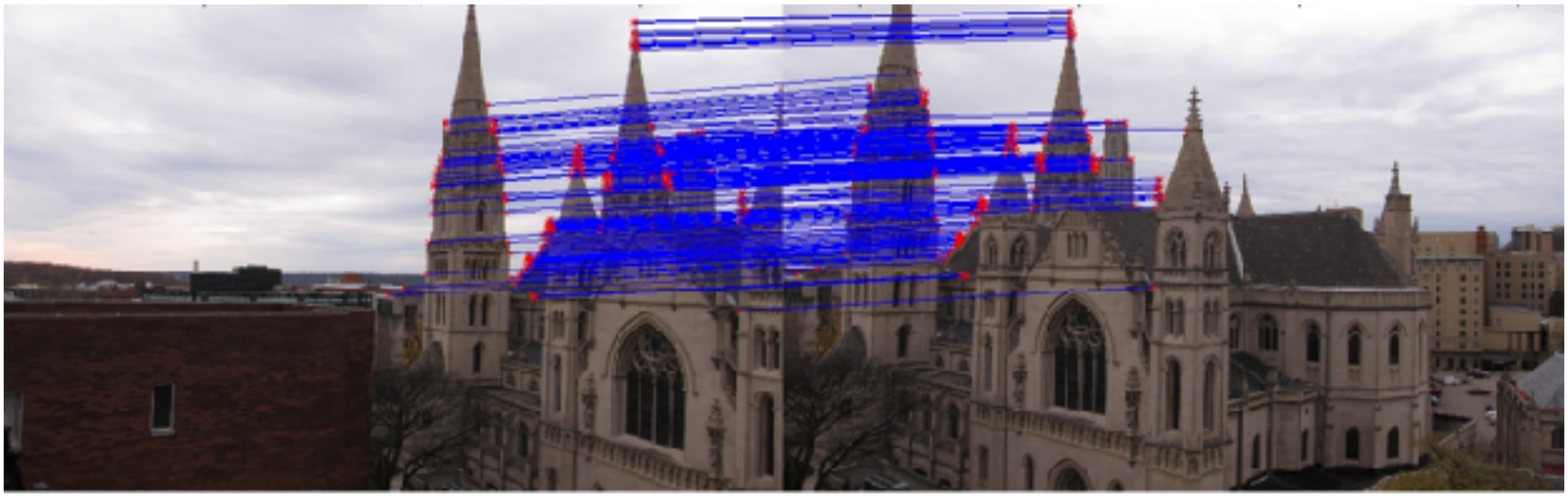
Why did it fail?

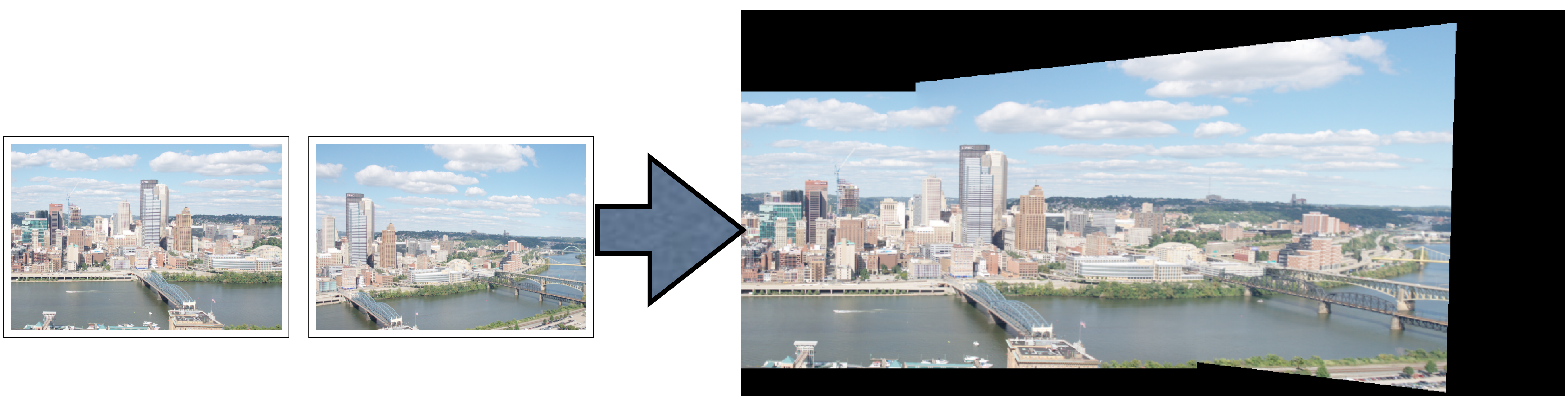
Original

Resized

Feature Descriptors

Core visual understanding task: finding correspondences between images





Source: Deva Ramanan

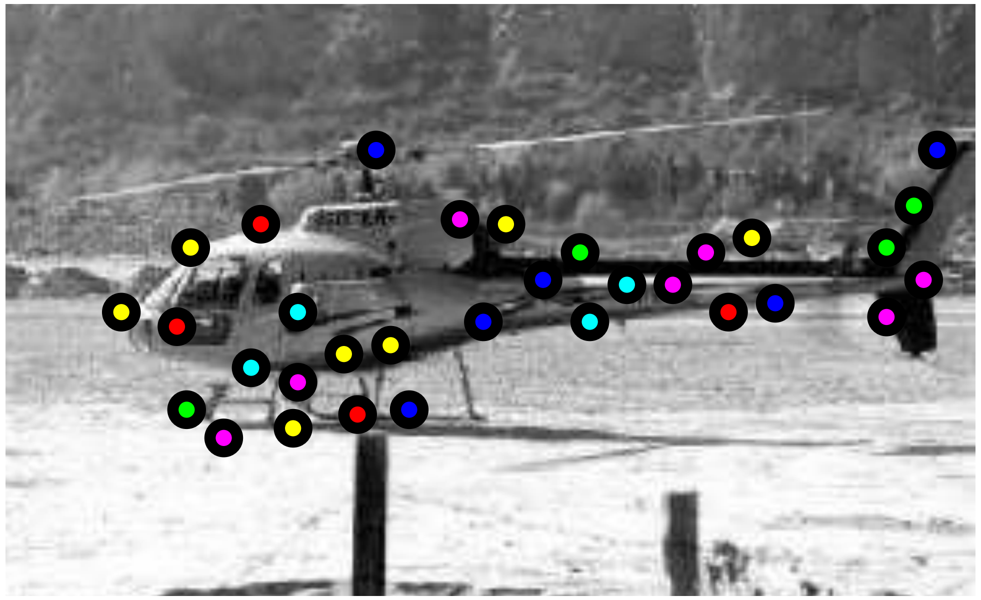
Example: image matching of landmarks

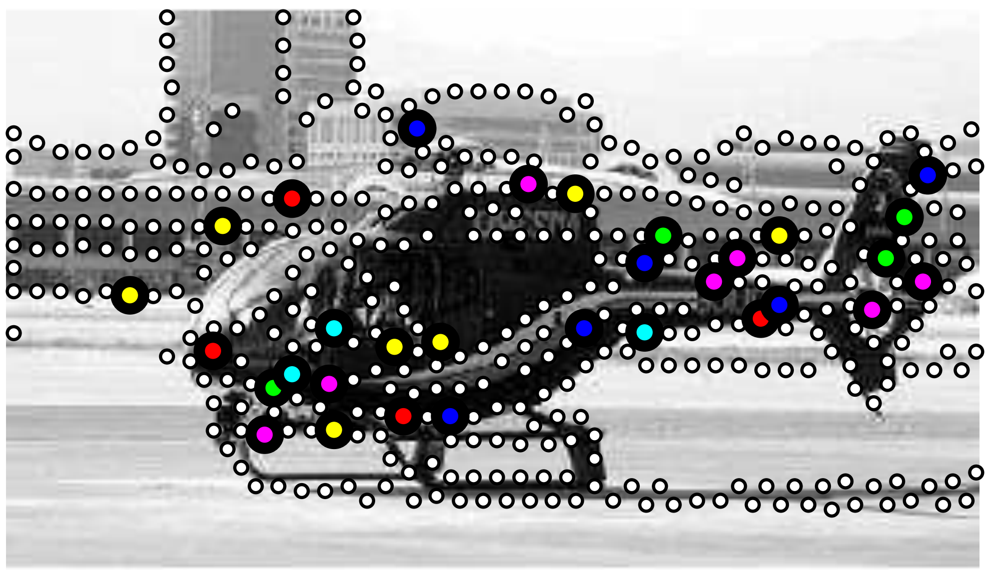
Correspondence + geometry estimation

Source: Deva Ramanan

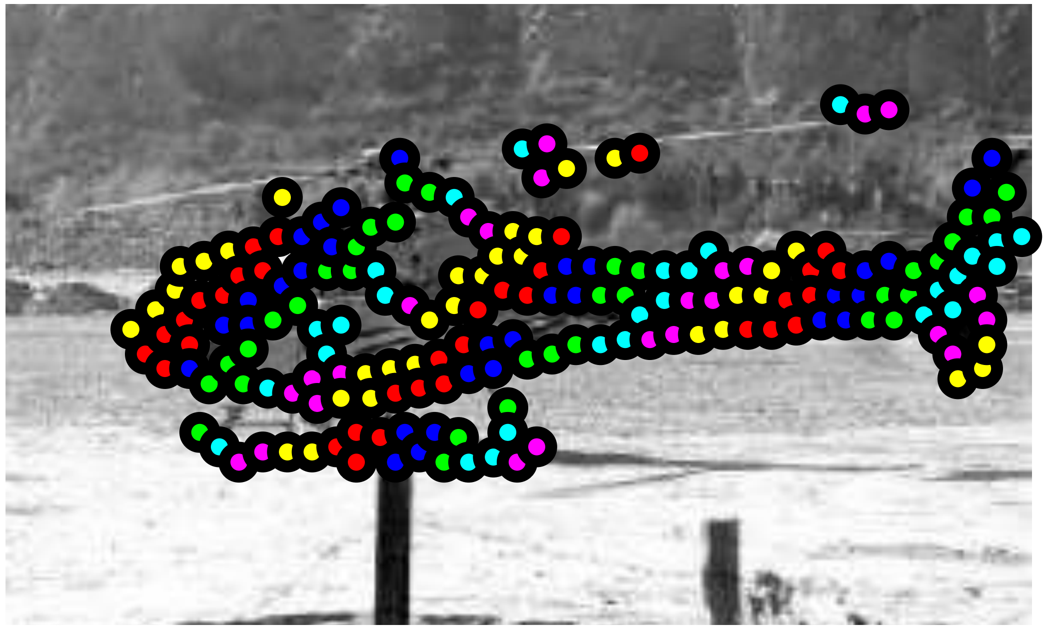
Object recognition by matching

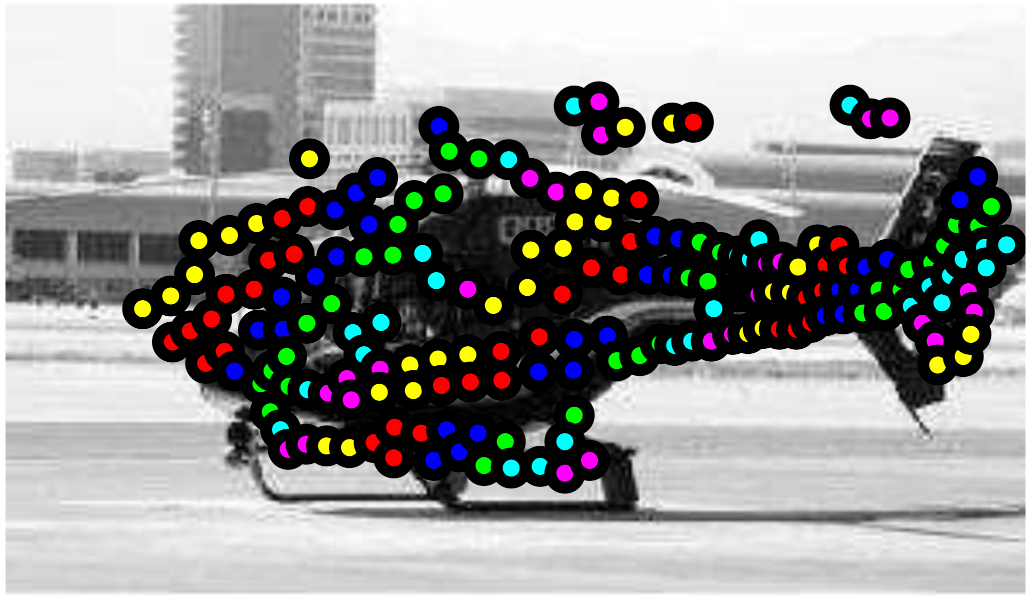
Sparse correspondence





Dense corrrespondence



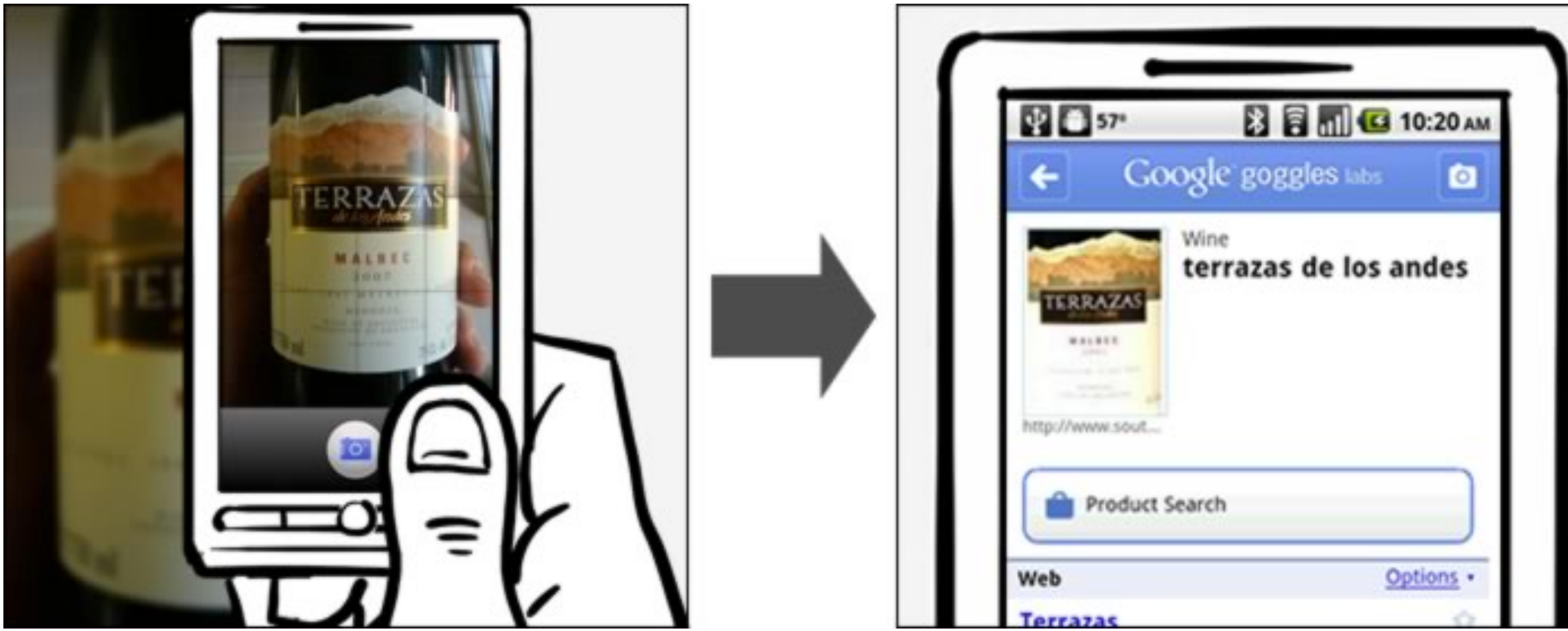


Source: Deva Ramanan

Example: license plate recognition

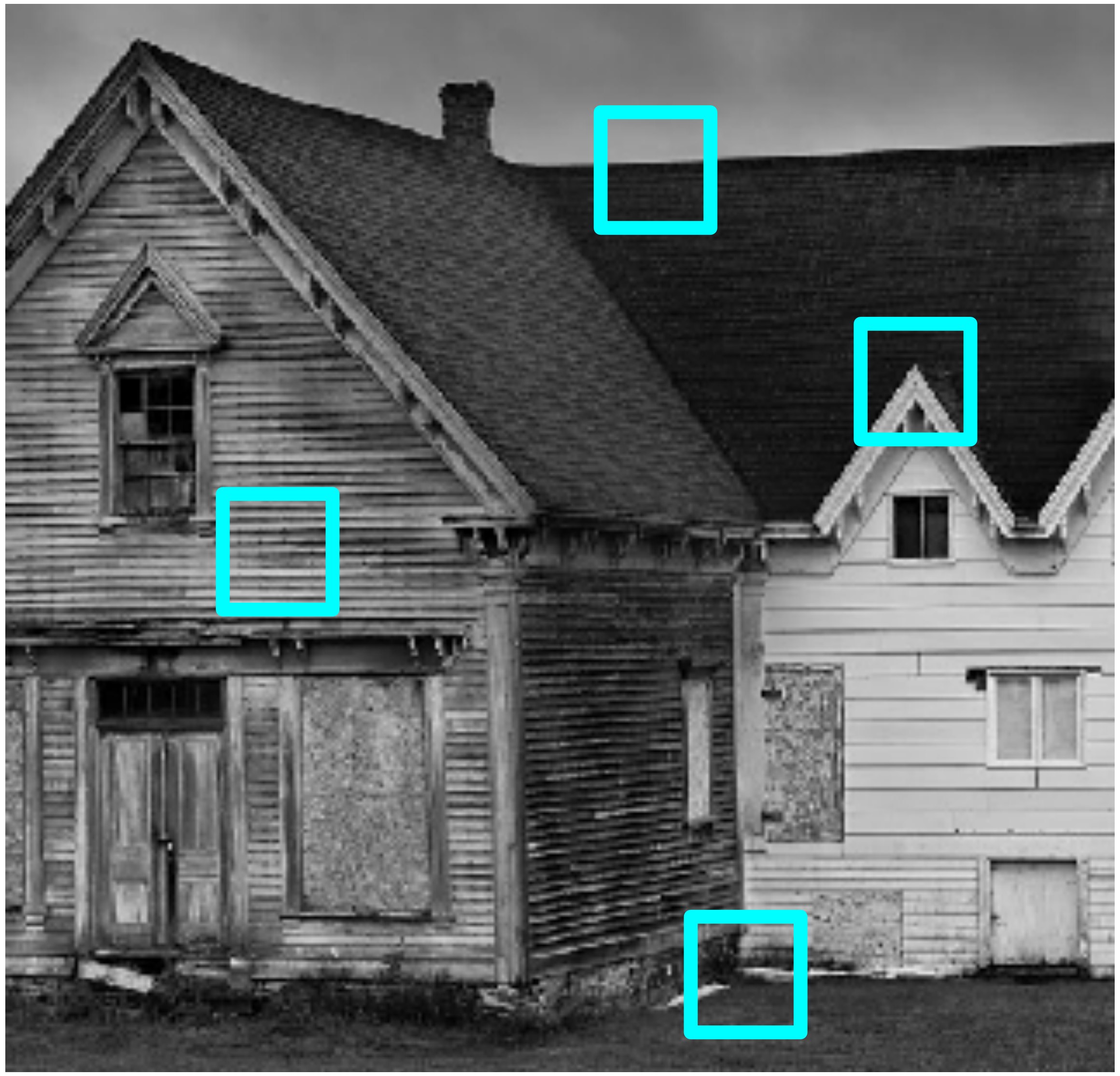
Source: Deva Ramanan

Example: product recognition



Source: Deva Ramanan

Motivation

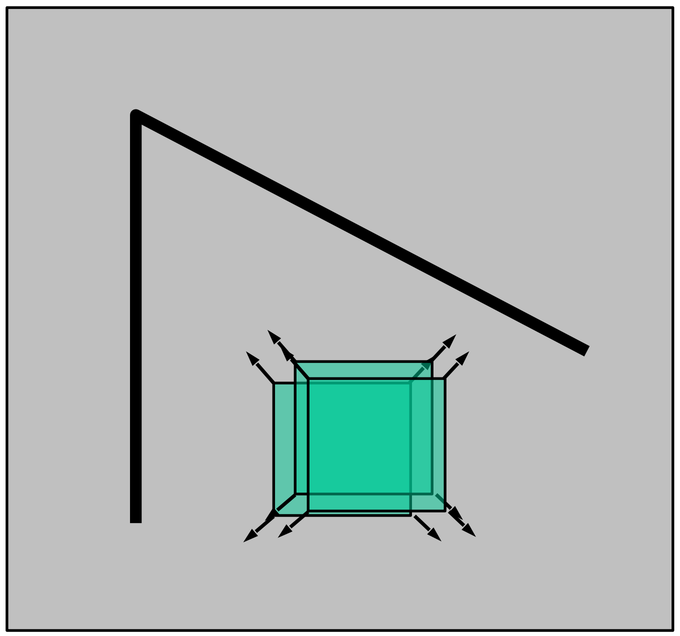


Which of these patches are easier to match?

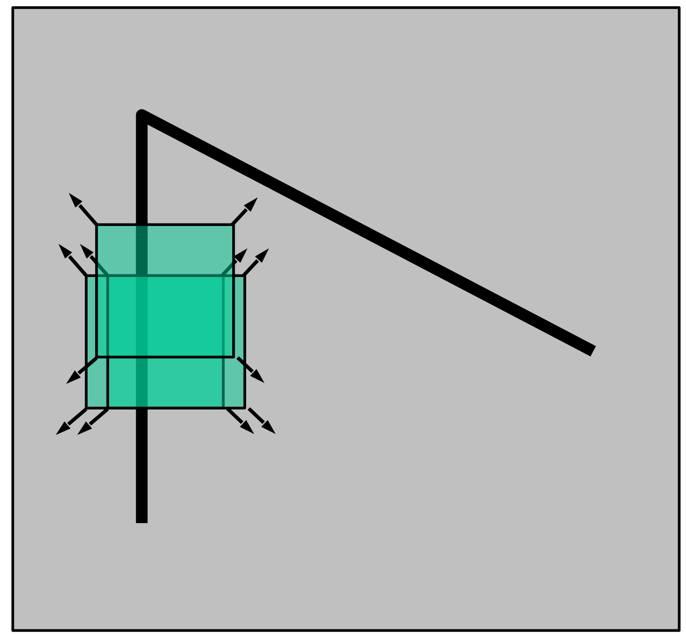
Why? How can we mathematically operationalize this?

Source: Deva Ramanan

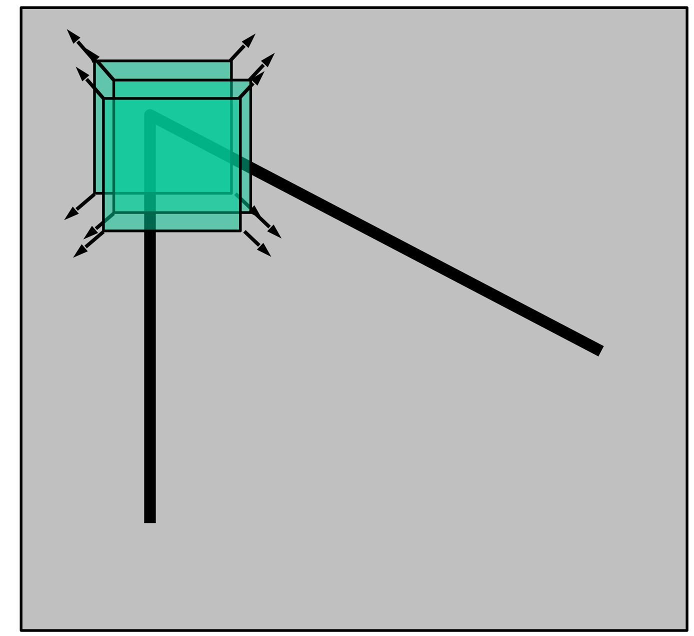
Corner Detector: Basic Idea



"flat" region:
no change in any direction



"edge":
no change along the edge direction



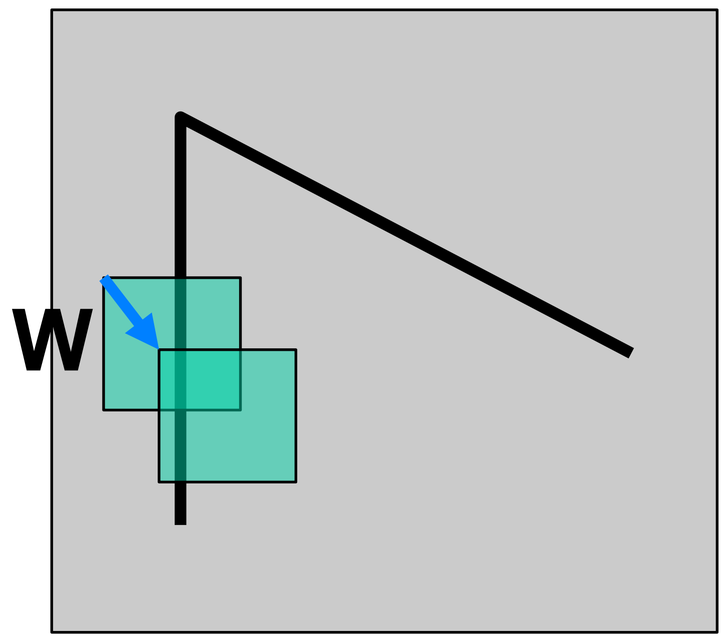
"corner":
significant change in all directions

Defn: points are "matchable" if small shifts always produce a large SSD error

Source: Deva Ramanan

The math

Defn: points are “matchable” if small shifts always produce a large SSD error

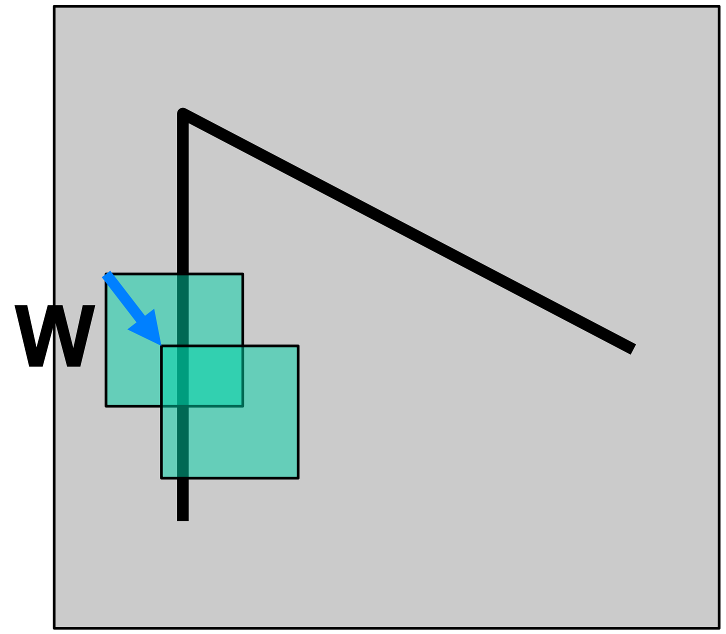


$$E_{x_0,y_0}(u,v) = \sum_{(x,y)\in W(x_0,y_0)} [I(x+u,y+v) - I(x,y)]^2$$

Source: Deva Ramanan

The math

Defn: points are “matchable” if small shifts always produce a large SSD error



$$\text{cornerness}(x_0, y_0) = \min_{u,v} E_{x_0,y_0}(u, v)$$

where

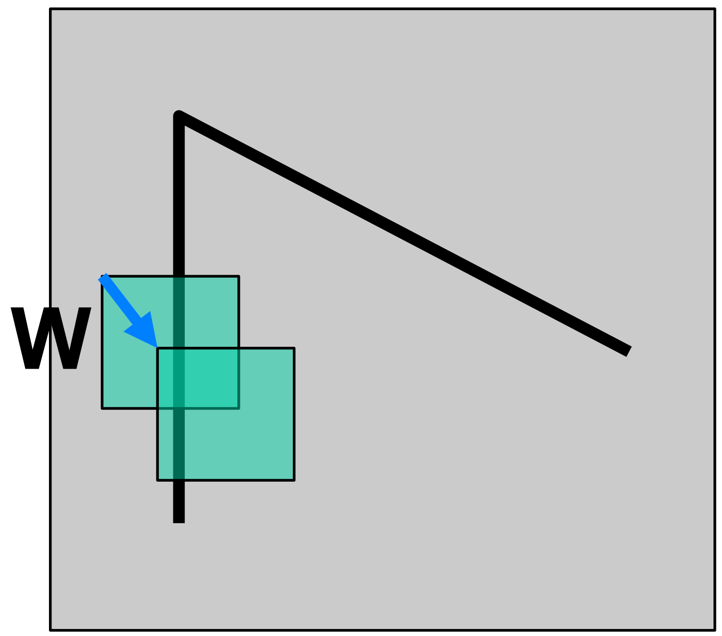
$$E_{x_0,y_0}(u, v) = \sum_{(x,y)\in W(x_0,y_0)} [I(x + u, y + v) - I(x, y)]^2$$

Why can’t this be right?

Source: Deva Ramanan

The math

Defn: points are “matchable” if small shifts always produce a large SSD error



$$\text{cornerness}(x_0, y_0) = \min_{u^2 + v^2 = 1} E_{x_0,y_0}(u, v)$$

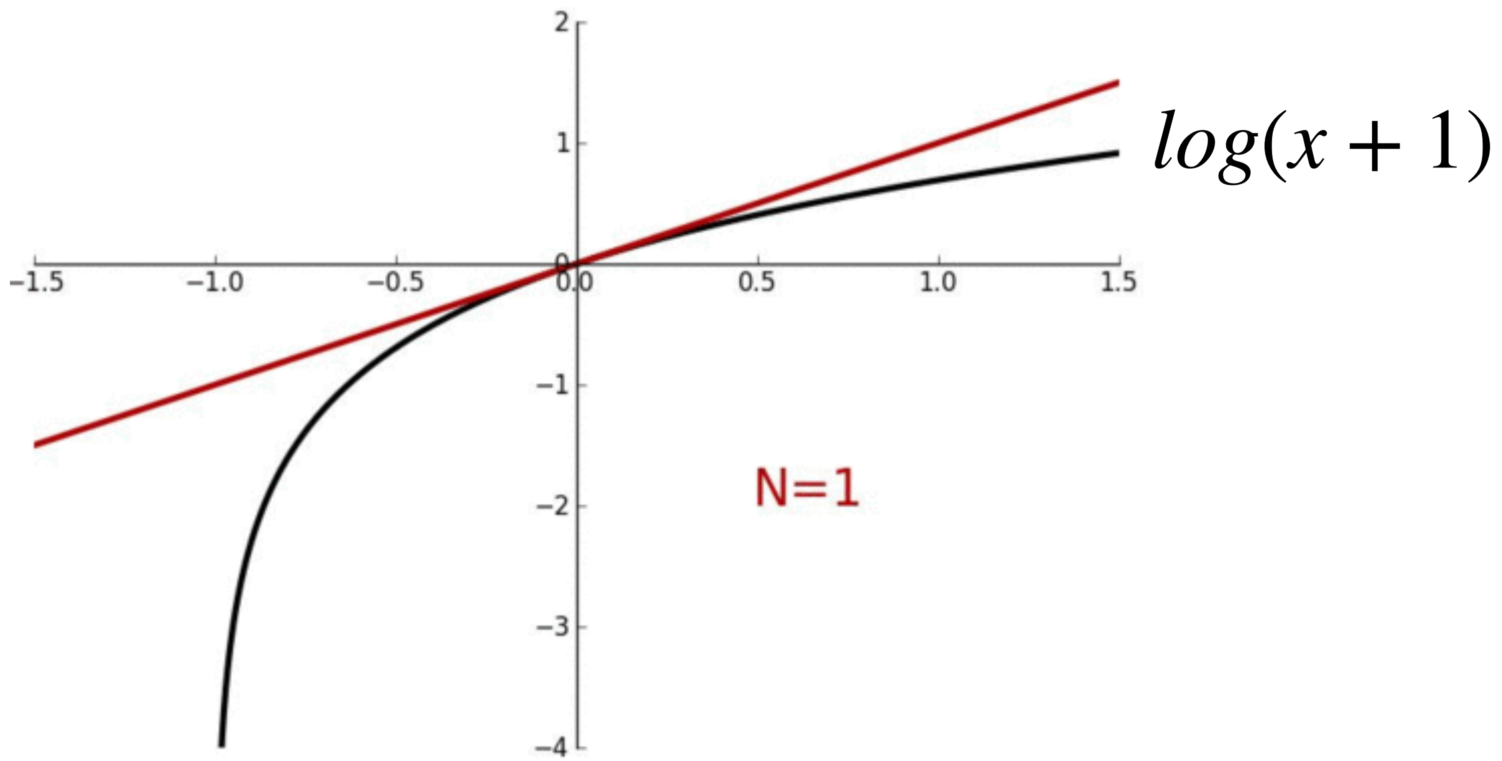
where

$$E_{x_0,y_0}(u, v) = \sum_{(x,y)\in W(x_0,y_0)} [I(x + u, y + v) - I(x, y)]^2$$

Source: Deva Ramanan

Background: taylor series expansion

$$f(x+u) = f(x) + \frac{\partial f(x)}{\partial x} u + \frac{1}{2} \frac{\partial f(x)}{\partial xx} u^2 + \text{Higher Order Terms}$$



Why are low-order expansions reasonable?
Underyling smoothness of real-world signals

Source: Deva Ramanan

Multivariate taylor series

$$I(x+u, y+v) = I(x,y) + \underbrace{\begin{bmatrix} \frac{\partial I(x,y)}{\partial x} & \frac{\partial I(x,y)}{\partial y} \end{bmatrix}}_{\text{gradient}} \begin{bmatrix} u \\ v \end{bmatrix} +$$

$$\frac{1}{2} \begin{bmatrix} u & v \end{bmatrix} \underbrace{\begin{bmatrix} \frac{\partial I(x,y)}{\partial xx} & \frac{\partial I(x,y)}{\partial xy} \\ \frac{\partial I(x,y)}{\partial xy} & \frac{\partial I(x,y)}{\partial yy} \end{bmatrix}}_{\text{Hessian}} \begin{bmatrix} u \\ v \end{bmatrix} + \text{Higher Order Terms}$$

$$I(x+u, y+v) \approx \mathbf{I} + \mathbf{I}_x u + \mathbf{I}_y v$$

where

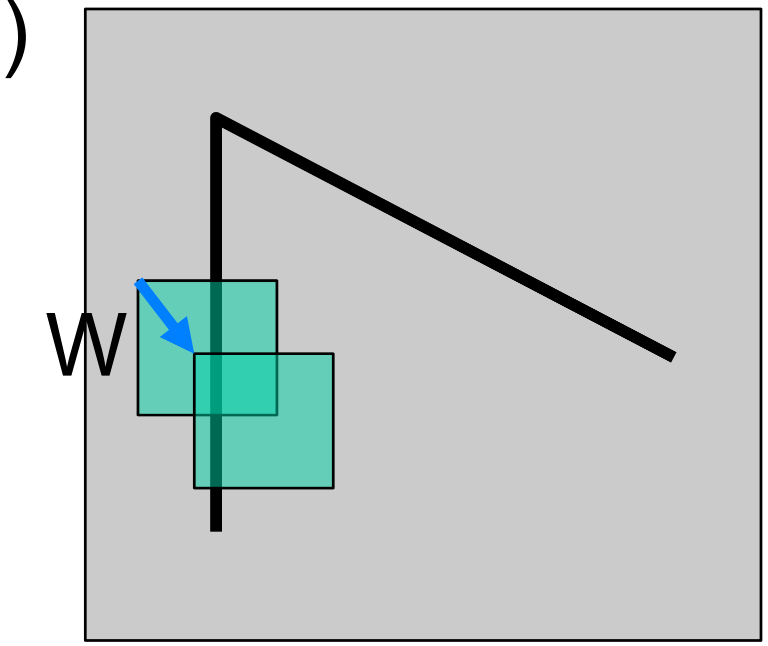
$$\mathbf{I}_x = \frac{\partial I(x,y)}{\partial x}$$

Source: Deva Ramanan

Feature detection: the math

Consider shifting the window W by (u,v)

- how do the pixels in W change?
- compare each pixel before and after by summing up the squared differences
- this defines an "error" of E(u,v):



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

$$\approx \sum_{(x,y)\in W} [\mathbf{I} + \mathbf{I}_x u + \mathbf{I}_y v - \mathbf{I}]^2$$

$$= \sum_{(x,y)\in W} [\mathbf{I}_x^2 u^2 + \mathbf{I}_y^2 v^2 + 2\mathbf{I}_x \mathbf{I}_y uv]$$

$$= \begin{bmatrix} u & v \end{bmatrix} A \begin{bmatrix} u \\ v \end{bmatrix}, \quad A = \sum_{(x,y)\in W} \begin{bmatrix} \mathbf{I}_x^2 & \mathbf{I}_x \mathbf{I}_y \\ \mathbf{I}_y \mathbf{I}_x & \mathbf{I}_y^2 \end{bmatrix}$$

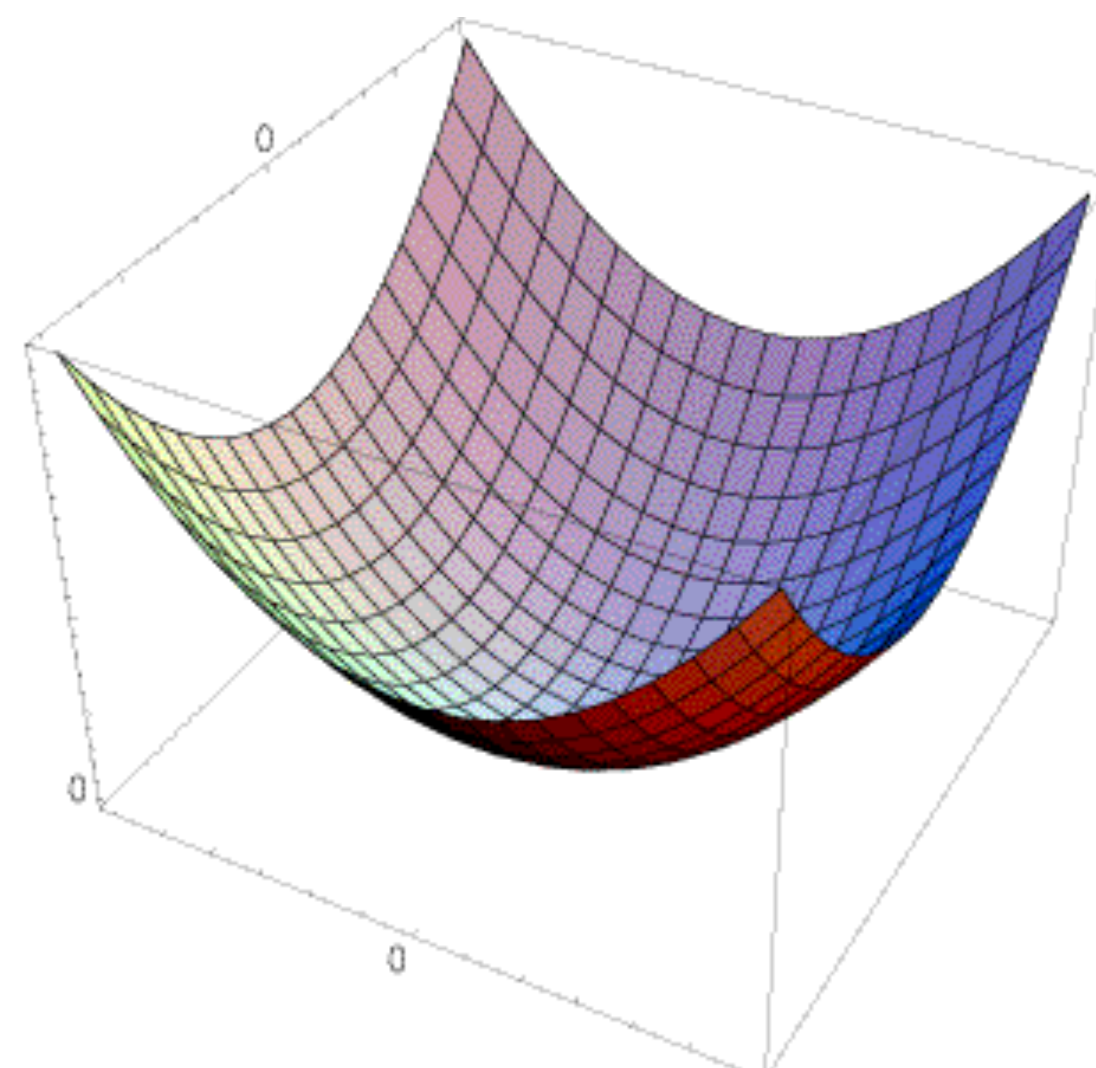
Source: Deva Ramanan

Interpreting the second moment matrix

The surface *E*(*u*,*v*) is locally approximated by a quadratic form. Let's try to understand its shape.

$$E(u,v) \approx [u \;\; v] \;\; A \begin{bmatrix} u \\ v \end{bmatrix}$$

$$A = \sum_{(x,y)\in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

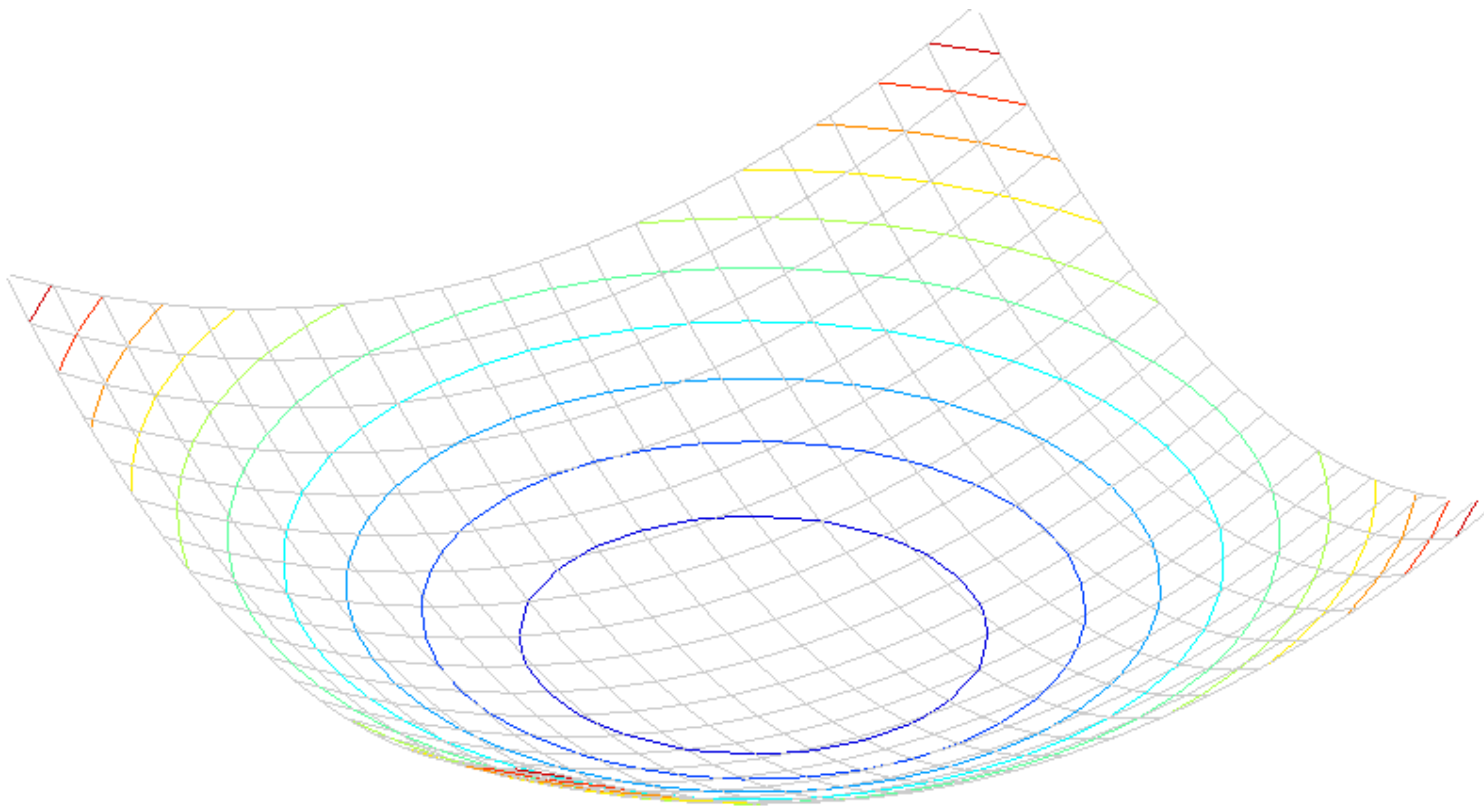


James Hays

Interpreting the second moment matrix

Consider a horizontal "slice" of $E(u, v)$: $[u \ v] \ A \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$

This is the equation of an ellipse.



James Hays

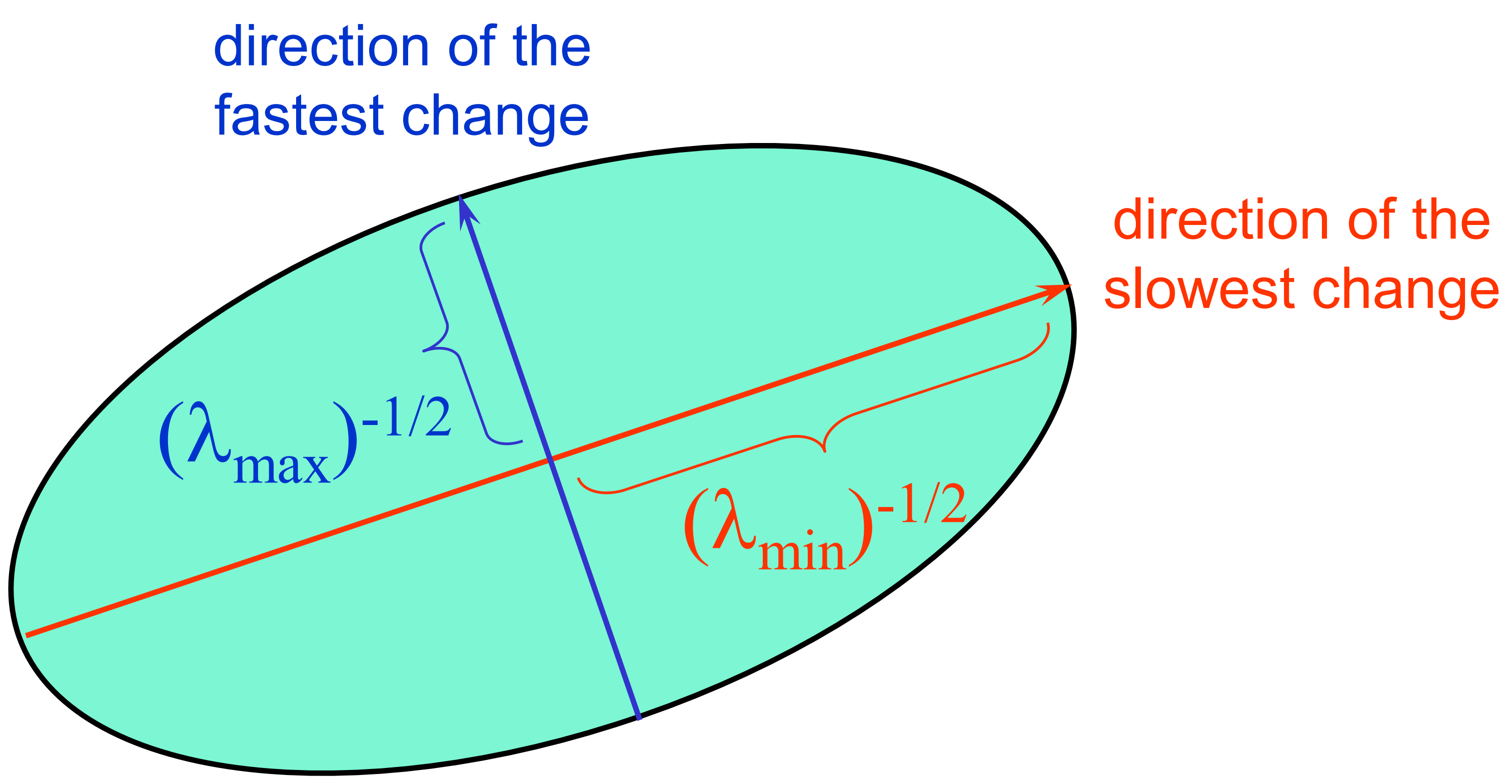
Interpreting the second moment matrix

Consider a horizontal "slice" of $E(u, v)$: $[u \ v] \ A \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$

This is the equation of an ellipse.

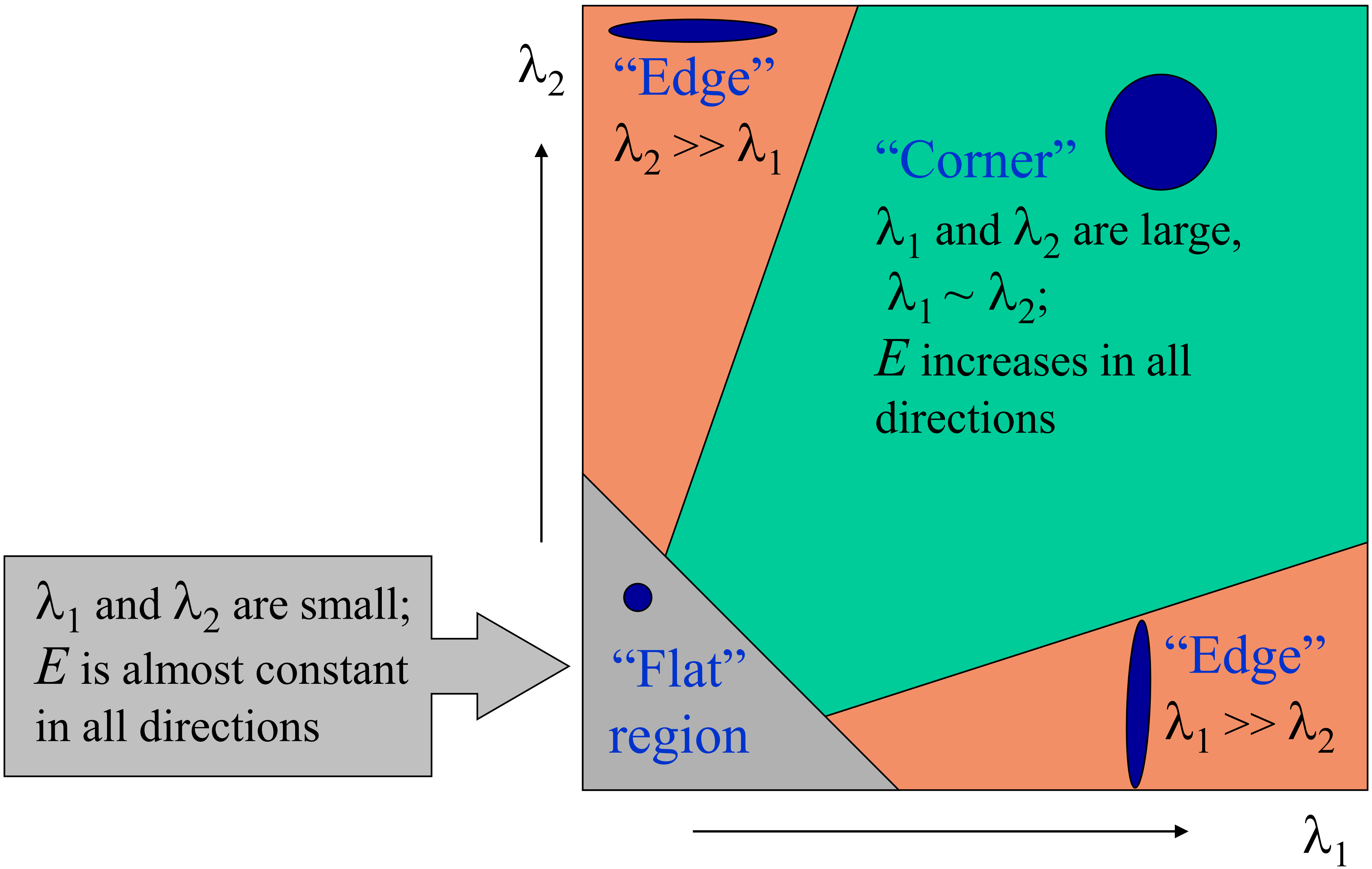
Diagonalization of A: $$A = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

The axis lengths of the ellipse are determined by the eigenvalues, and the orientation is determined by a rotation matrix R.



James Hays

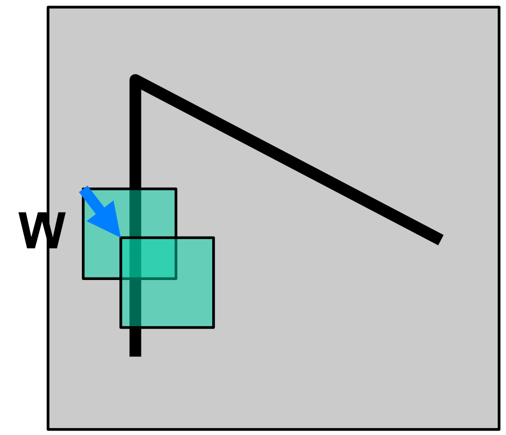
Classification of image points using eigenvalues of *M*



Source: Deva Ramanan

Back to corner(ness)

Defn: points are "matchable" if small shifts always produce a large SSD error



$$\text{Corner}(x_0, y_0) = \min_{u^2+v^2=1} E(u, v)$$

where

$$E(u,v) = \begin{bmatrix} u & v \end{bmatrix} A \begin{bmatrix} u \\ v \end{bmatrix}, \quad A = \sum_{(x,y)\in W(x_0,y_0)} \begin{bmatrix} \mathbf{I}_x^2 & \mathbf{I}_x\mathbf{I}_y \\ \mathbf{I}_y\mathbf{I}_x & \mathbf{I}_y^2 \end{bmatrix}$$

Solution is given by minimum eigenvalue

Implies (xo,yo) is a good corner if minimum eigenvalue is large

(or alternatively, if *both* eigenvalues of 'A' are large)

Source: Deva Ramanan

Efficient computation

Computing eigenvalues (and eigenvectors) is expensive

Turns out that it's easy to compute their sum (trace) and product (determinant)

- $\mathrm{Det}(A) = \lambda_{\min}\lambda_{\max}$
- $\mathrm{Trace}(A) = \lambda_{\min}+\lambda_{\max}$ (trace = sum of diagonal entries)

$$R = 4\frac{Det(A)}{Trace(A)^2}$$

(is proportional to the ratio of eigvenvalues and is 1 if they are equal)

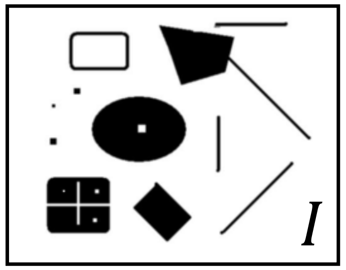
$$R = Det(A) - \alpha Trace(A)^2$$

(also favors large eigenvalues)

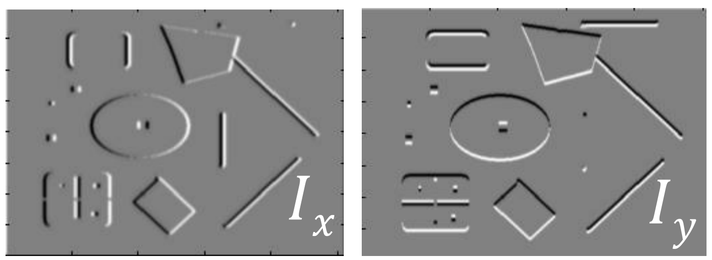
Source: Deva Ramanan

James Hays

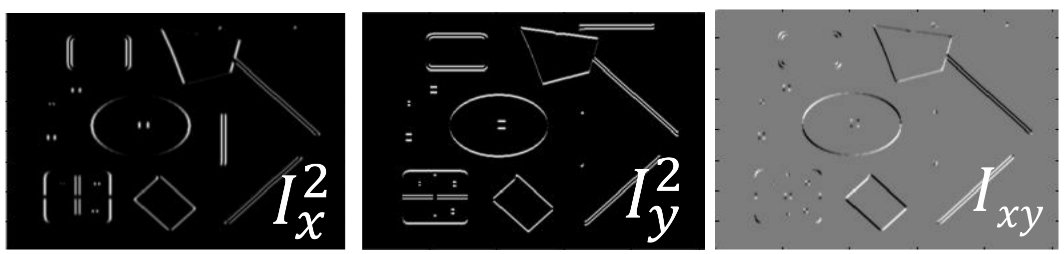
Harris Corner Detector [Harris88]



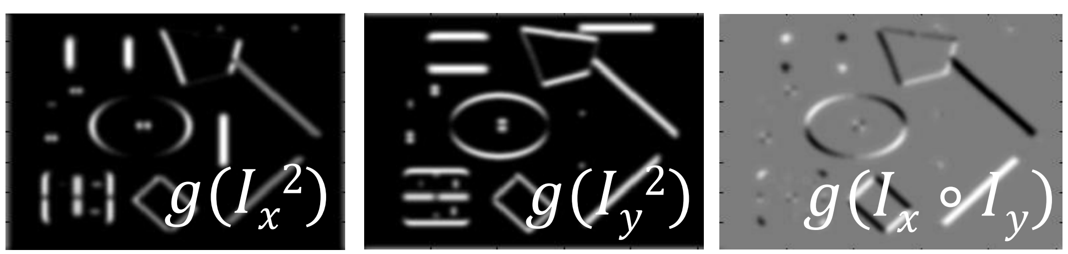
0. Input image
   We want to compute M at each pixel.



1. Compute image derivatives (optionally, blur first).



2. Compute M components
   as squares of derivatives.



3. Gaussian filter *g()* with width σ

4. Compute cornerness

   $$C = \det(M) - \alpha \operatorname{trace}(M)^2$$

5. Threshold on C to pick high cornerness

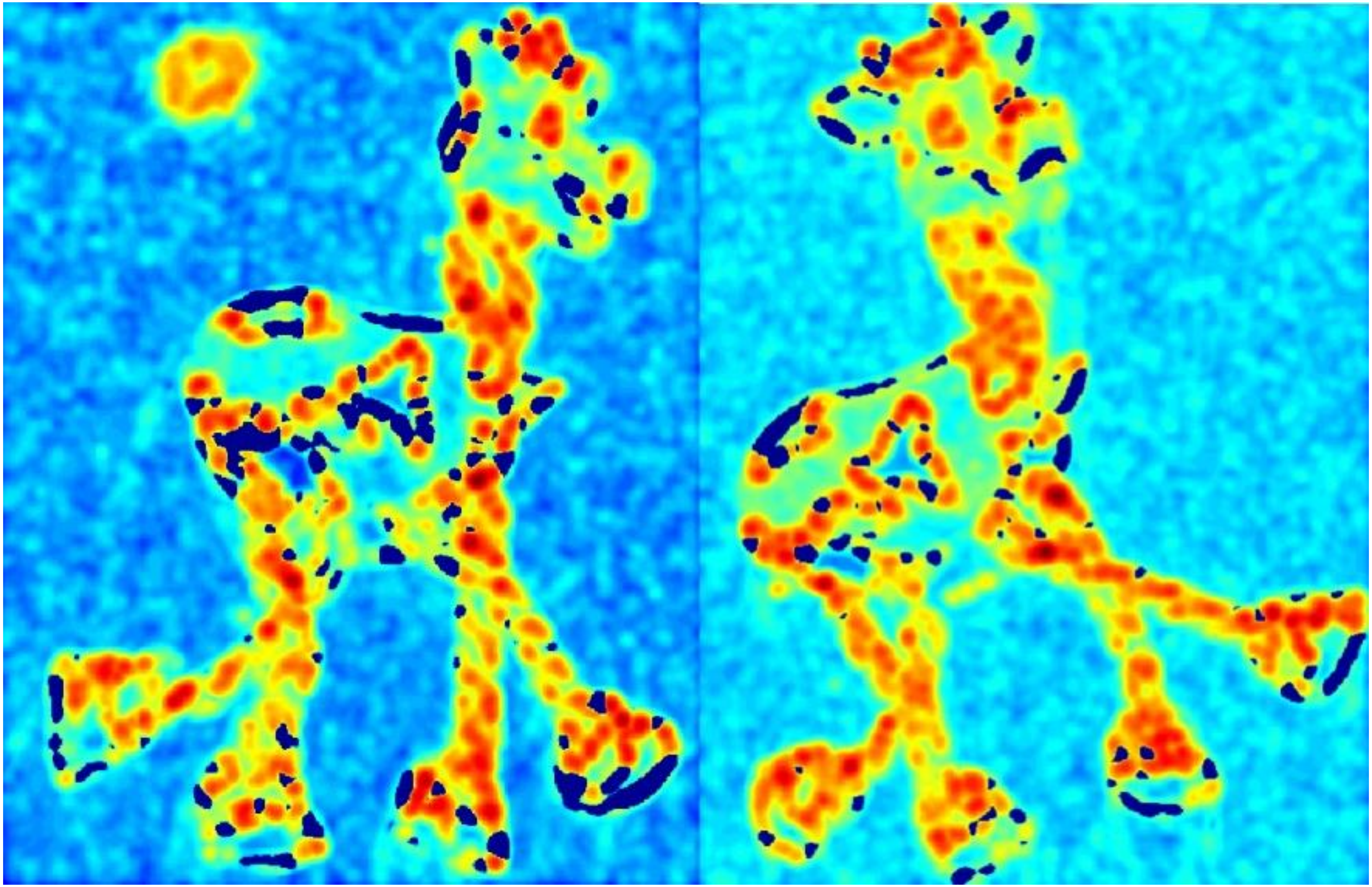
6. Non-maxima suppression to pick peaks.

Harris Detector: Steps

Source: Deva Ramanan

Harris Detector: Steps

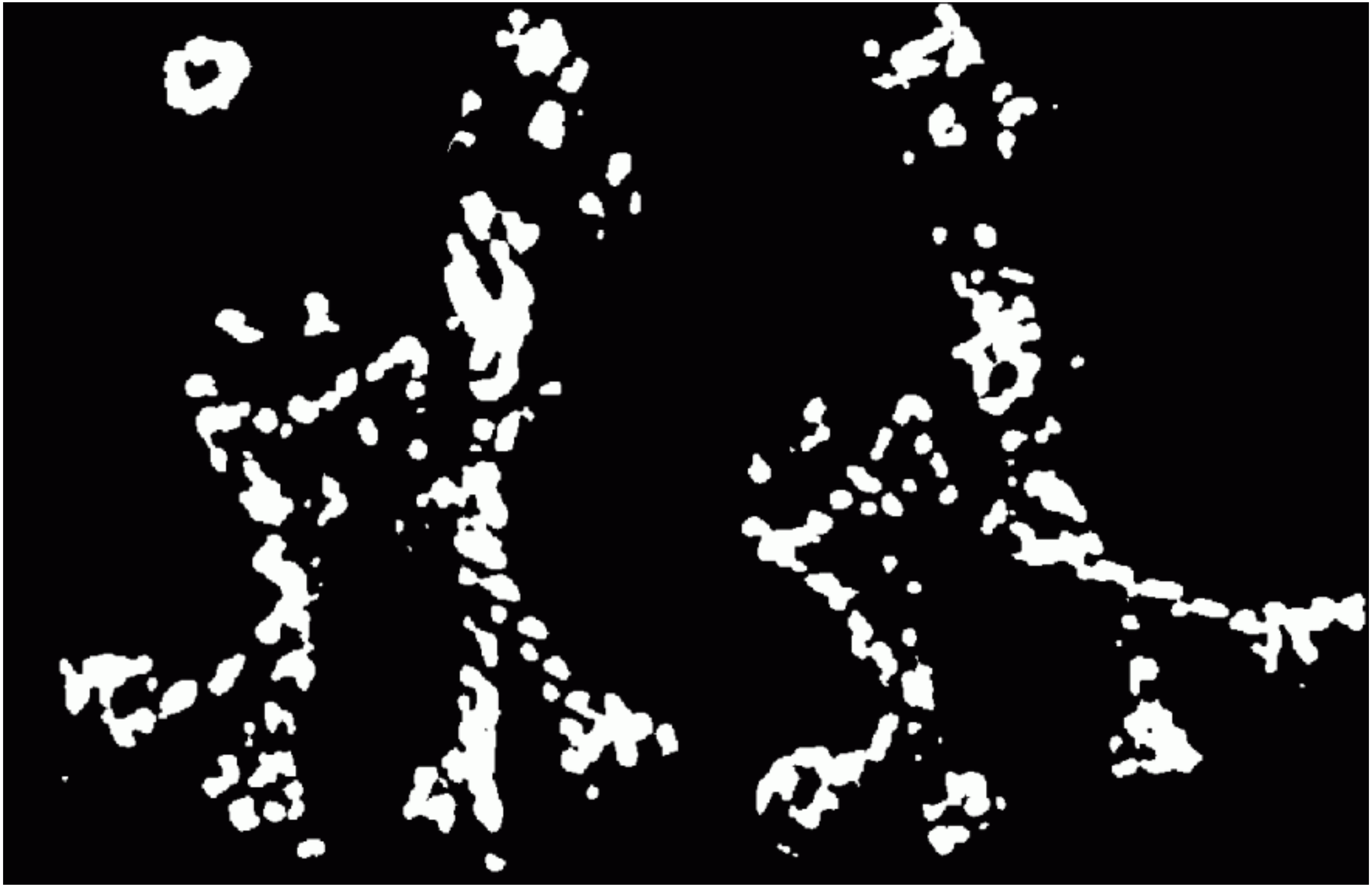
Compute corner response C



Source: Deva Ramanan

Harris Detector: Steps

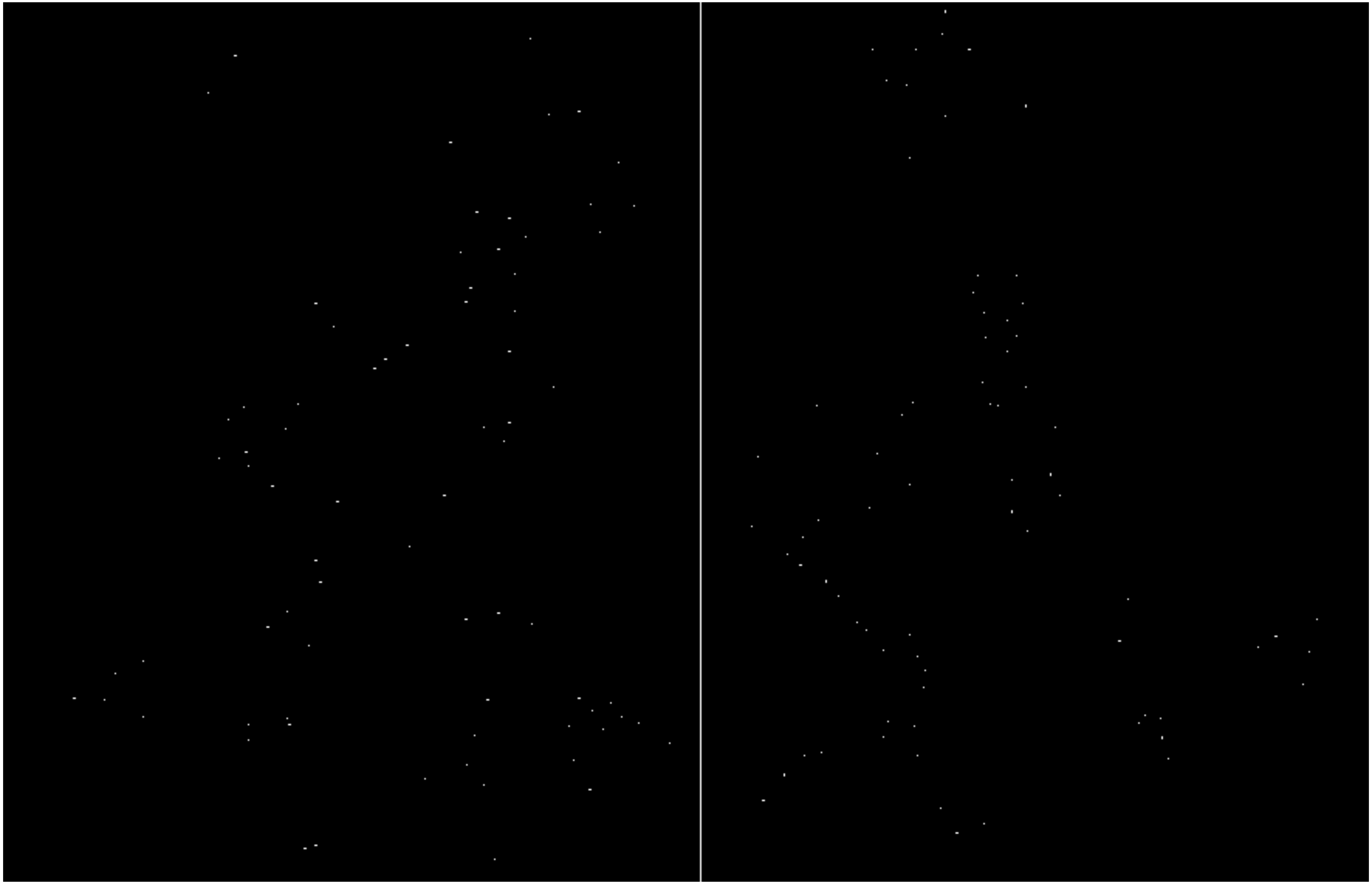
Find points with large corner response: C > threshold



Source: Deva Ramanan

Harris Detector: Steps

Take only the points of local maxima of C



Source: Deva Ramanan

Harris Detector: Steps

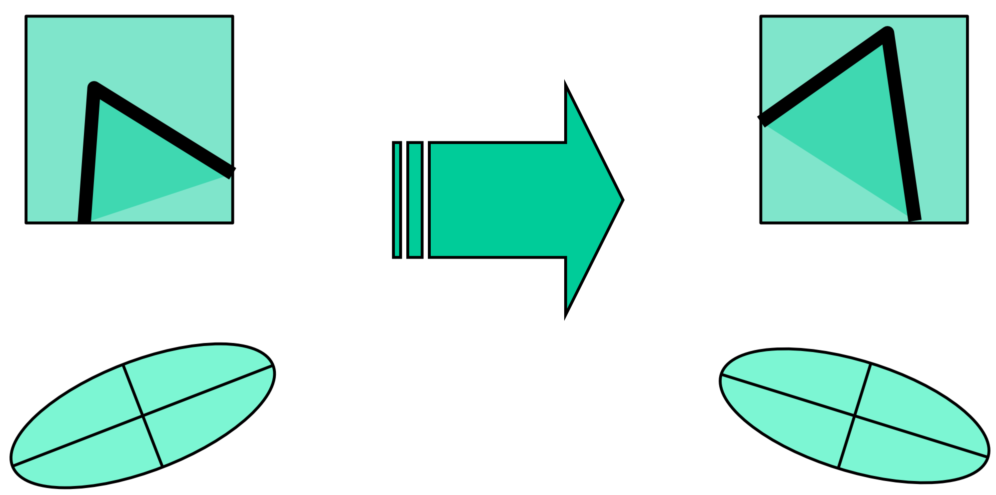
Source: Deva Ramanan

Scale and rotation invariance

Will interest point detector still fire on rotated & scaled images?

Source: Deva Ramanan

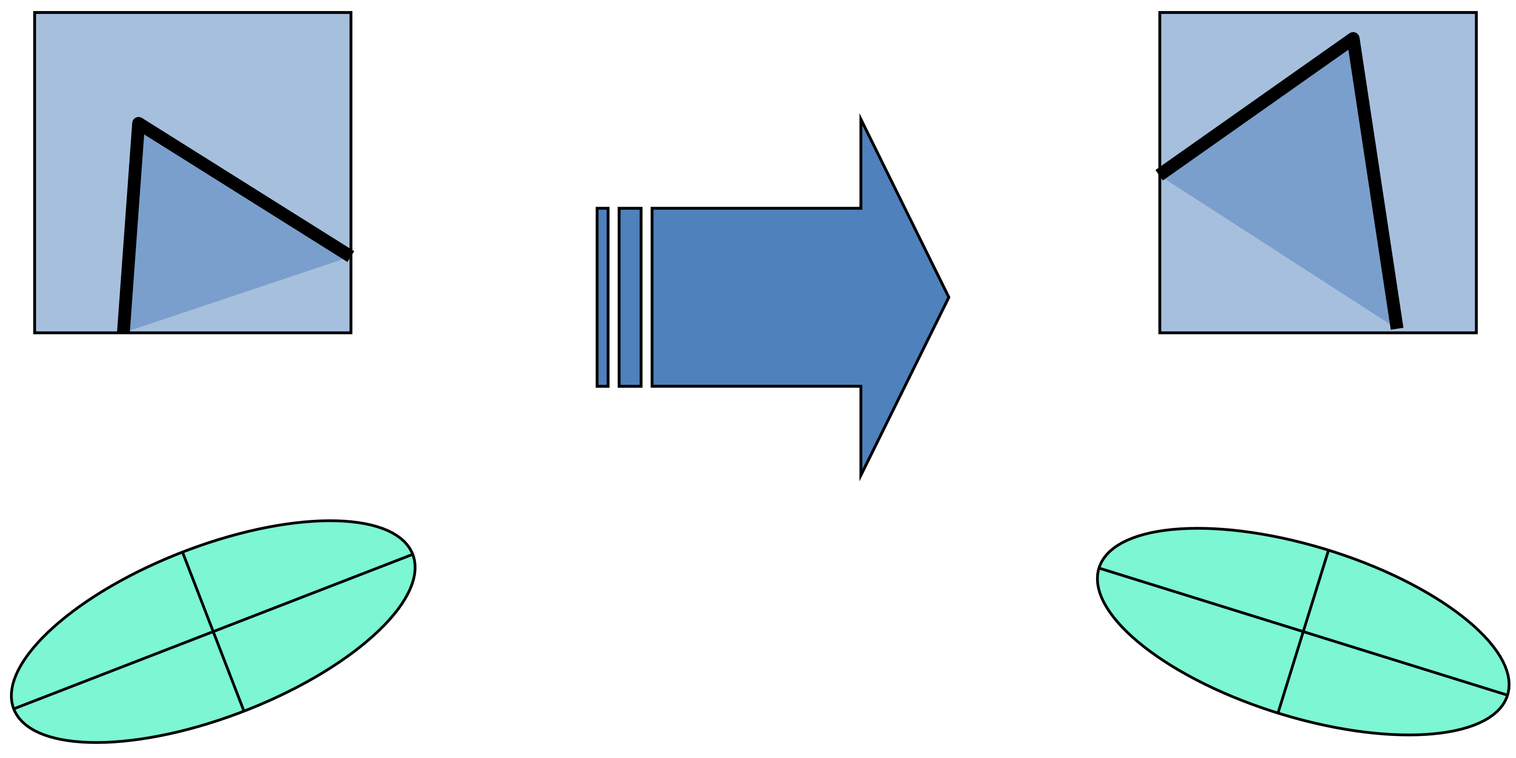
Rotation invariance (?)



Are eigenvector stable under rotations?
Are eigenvalues stable under rotations?

Source: Deva Ramanan

Image rotation

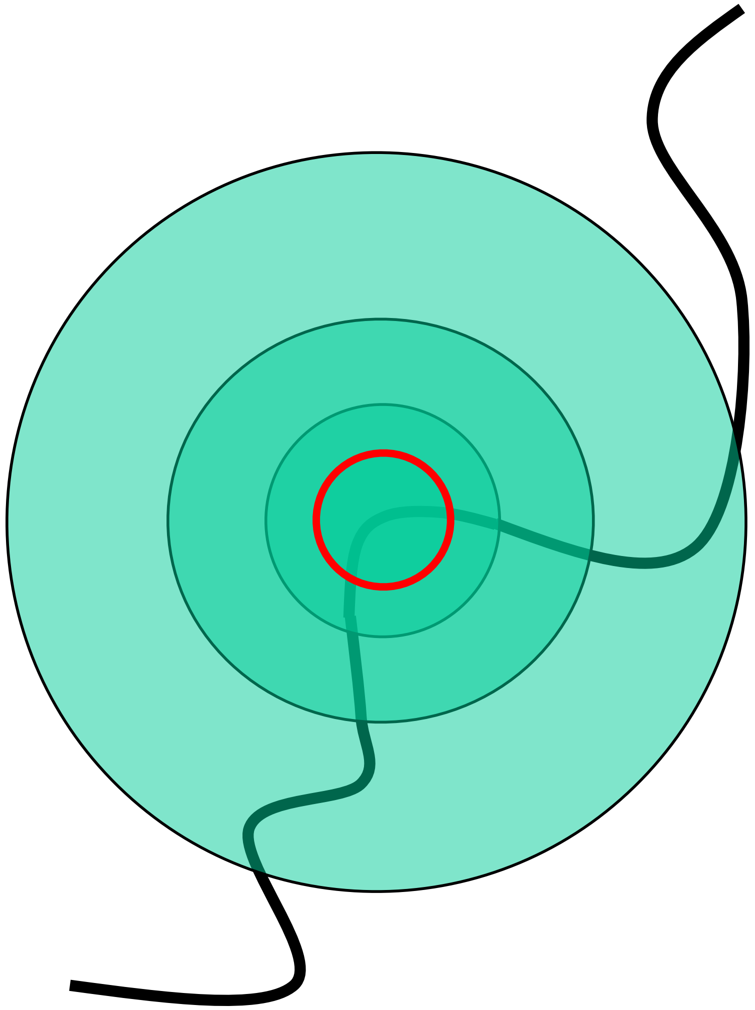


Second moment ellipse rotates but its shape (i.e., eigenvalues) remains the same.

Corner location is covariant w.r.t. rotation

James Hays

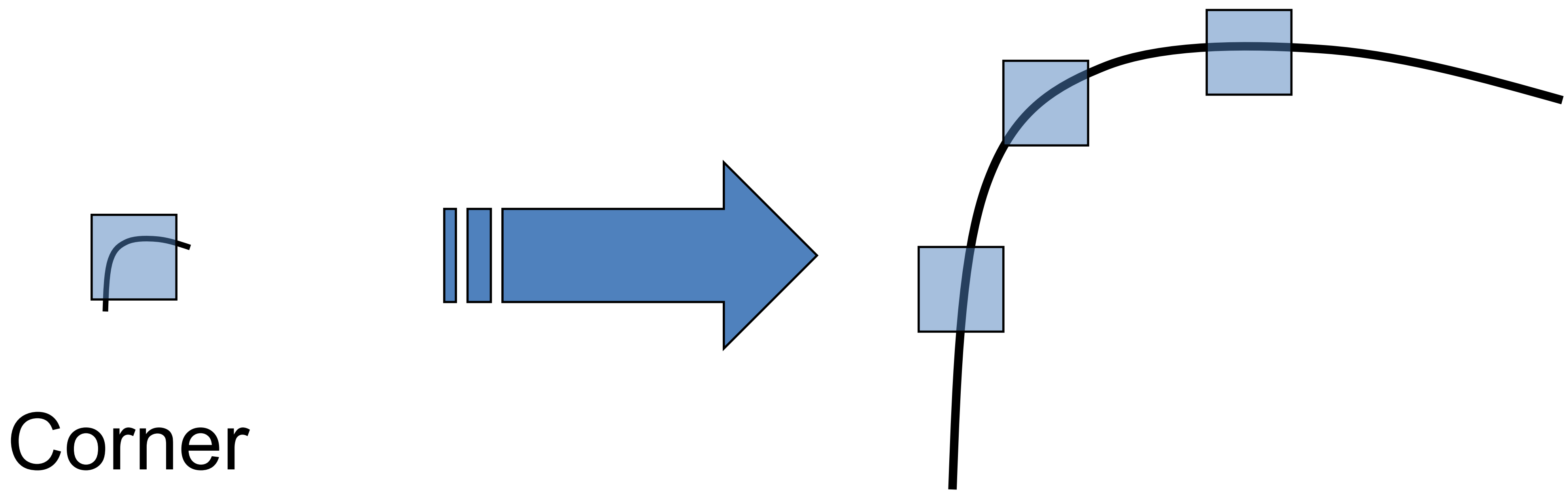
Scale invariance?



Are eigenvector stable under scalings?
Are eigenvalues stable under scalings?

Source: Deva Ramanan

Scaling

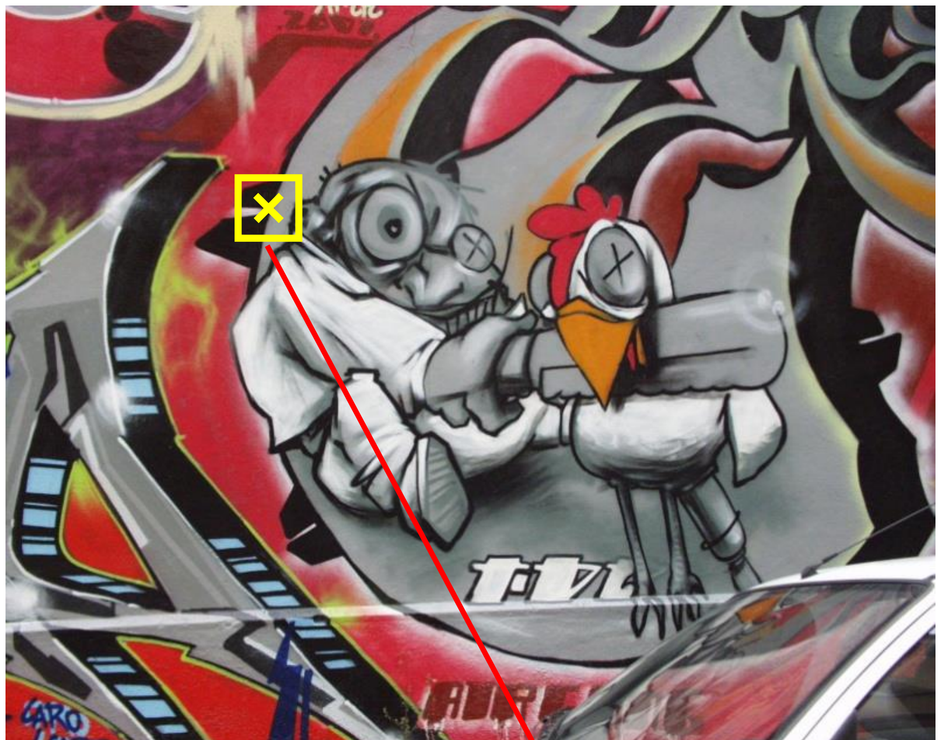


All points will be classified as edges

Corner location is not covariant to scaling!

James Hays

Automatic Scale Selection



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

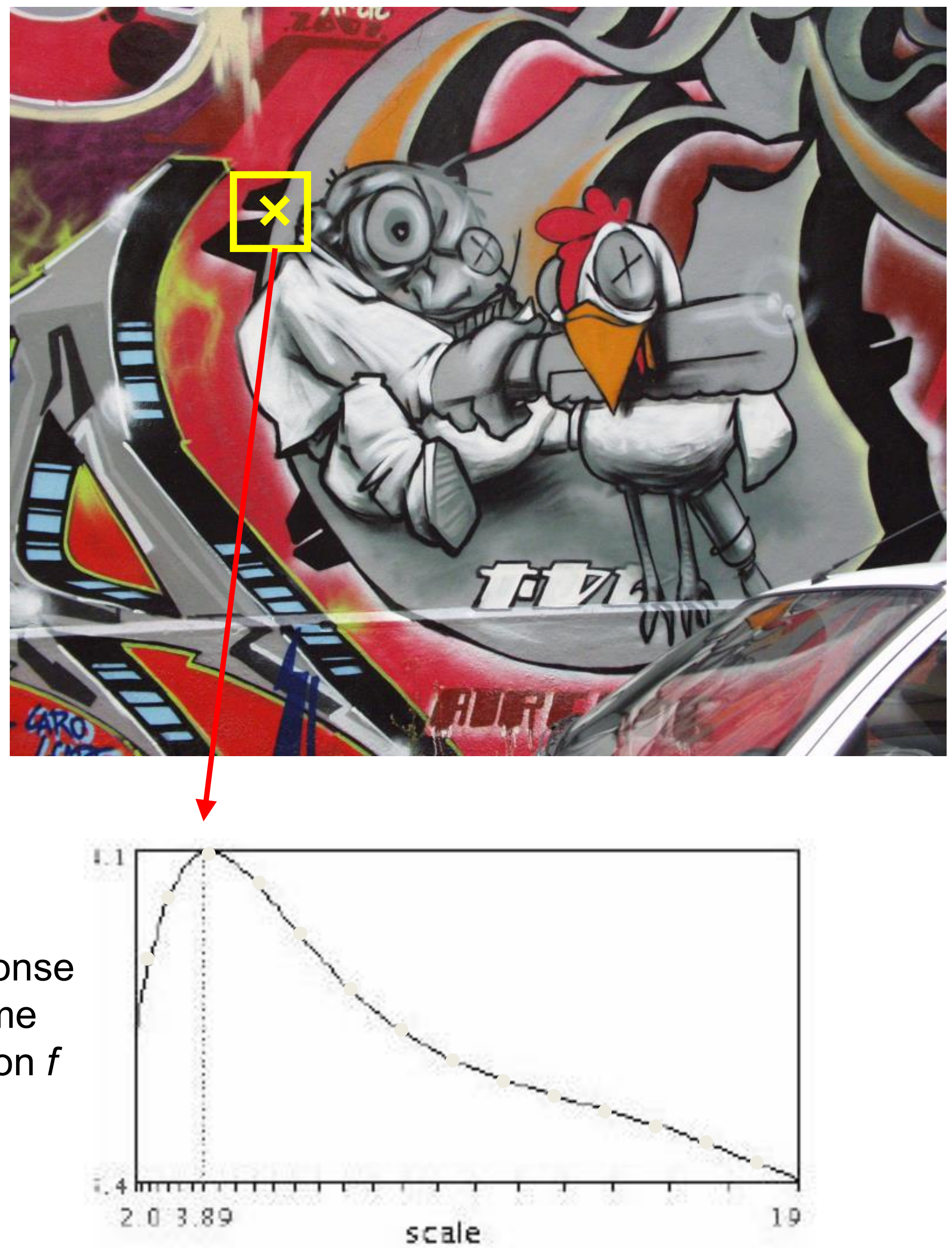
How to find patch sizes at which *f* response is equal?

What is a good *f* ?

K. Grauman, B. Leibe

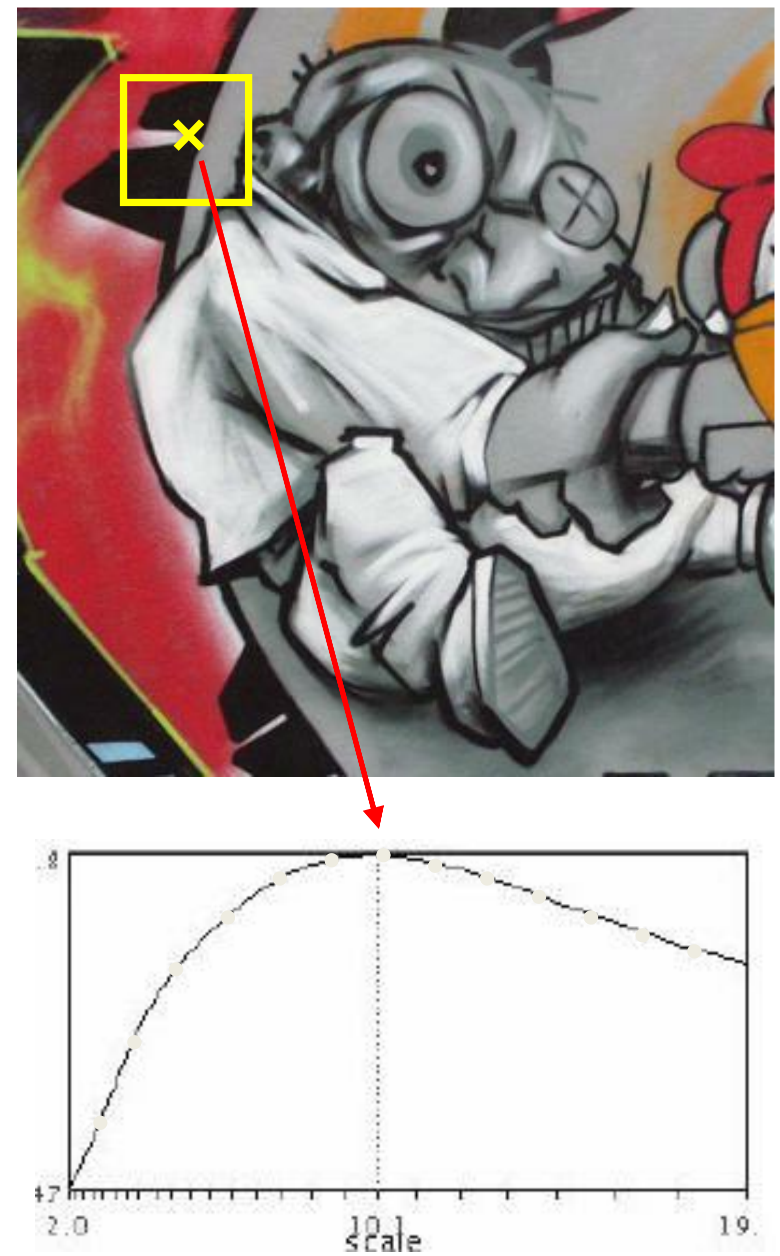
Automatic Scale Selection

- Function responses for increasing scale (scale signature)



Response of some function *f*

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

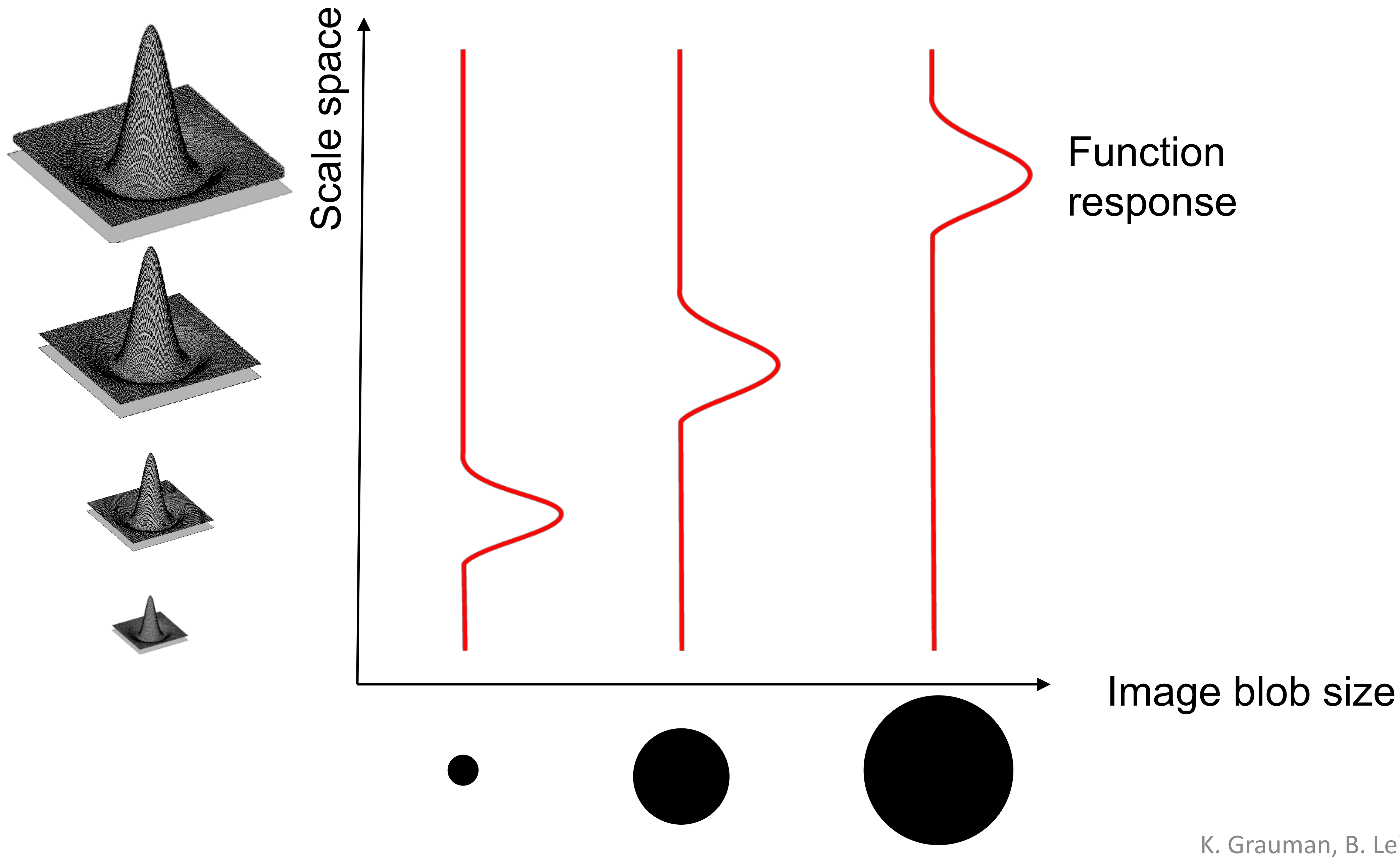


$$f(I_{i_1 \ldots i_m}(x', \sigma'))$$

K. Grauman, B. Leibe

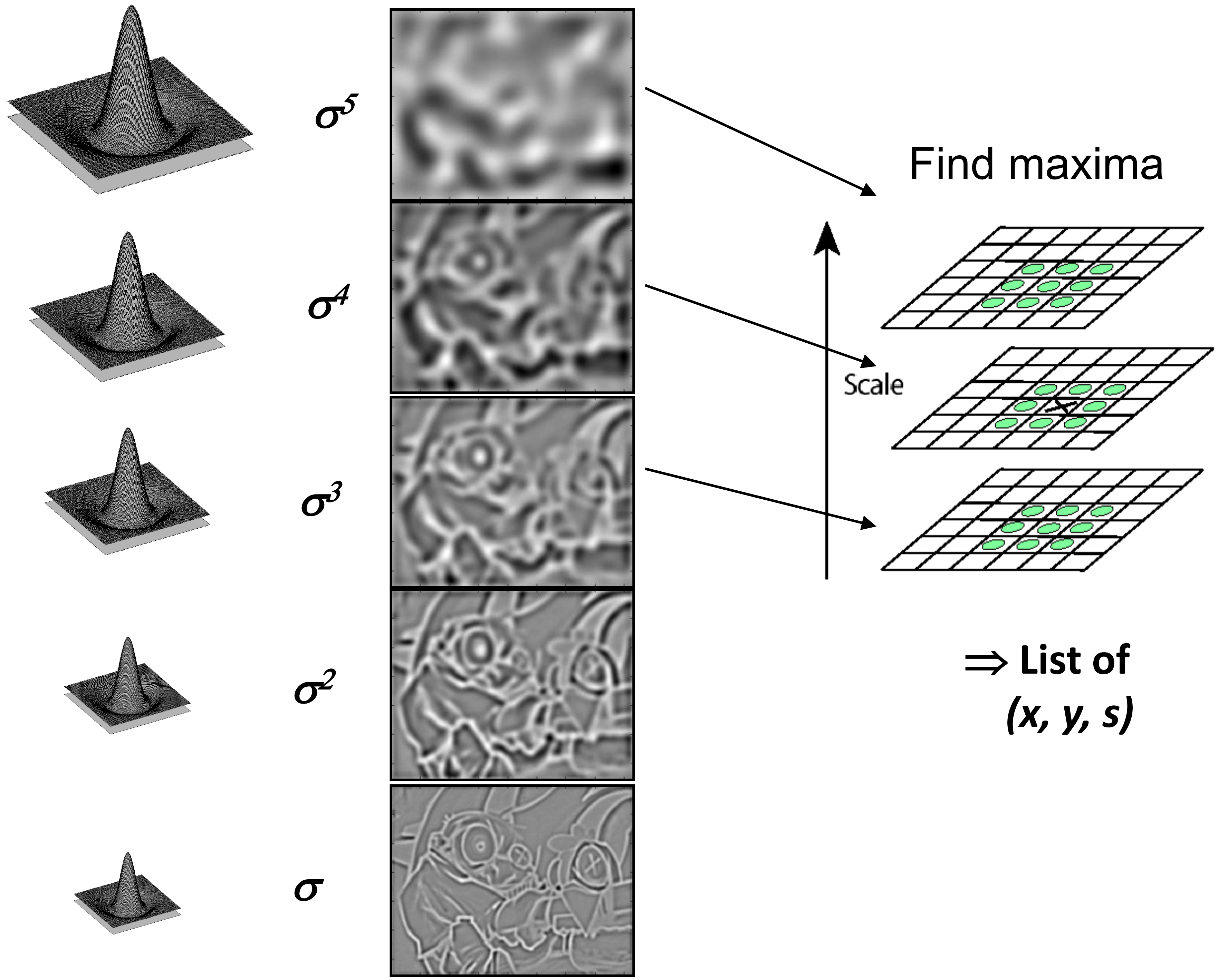
What Is A Useful Signature Function *f* ?

- "Blob" detector is common for corners
  - - Laplacian (2nd derivative) of Gaussian (LoG)



K. Grauman, B. Leibe

Find local maxima in position-scale space



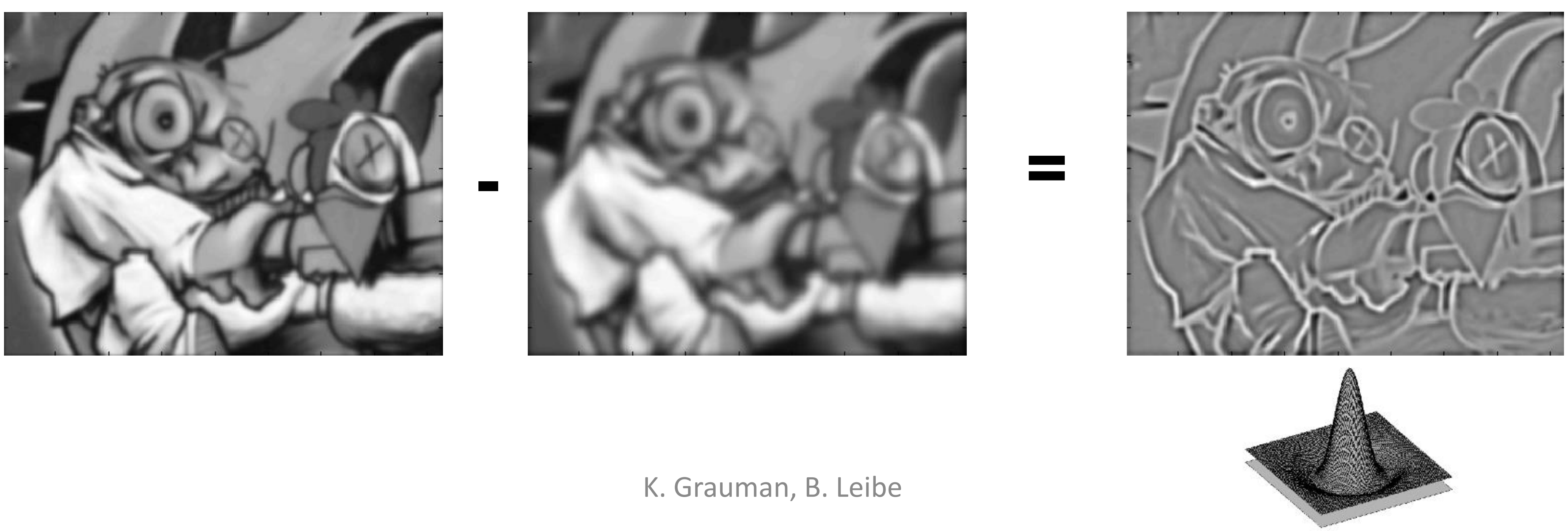
$\Rightarrow$ **List of**
(x, y, s)

K. Grauman, B. Leibe

Alternative approach

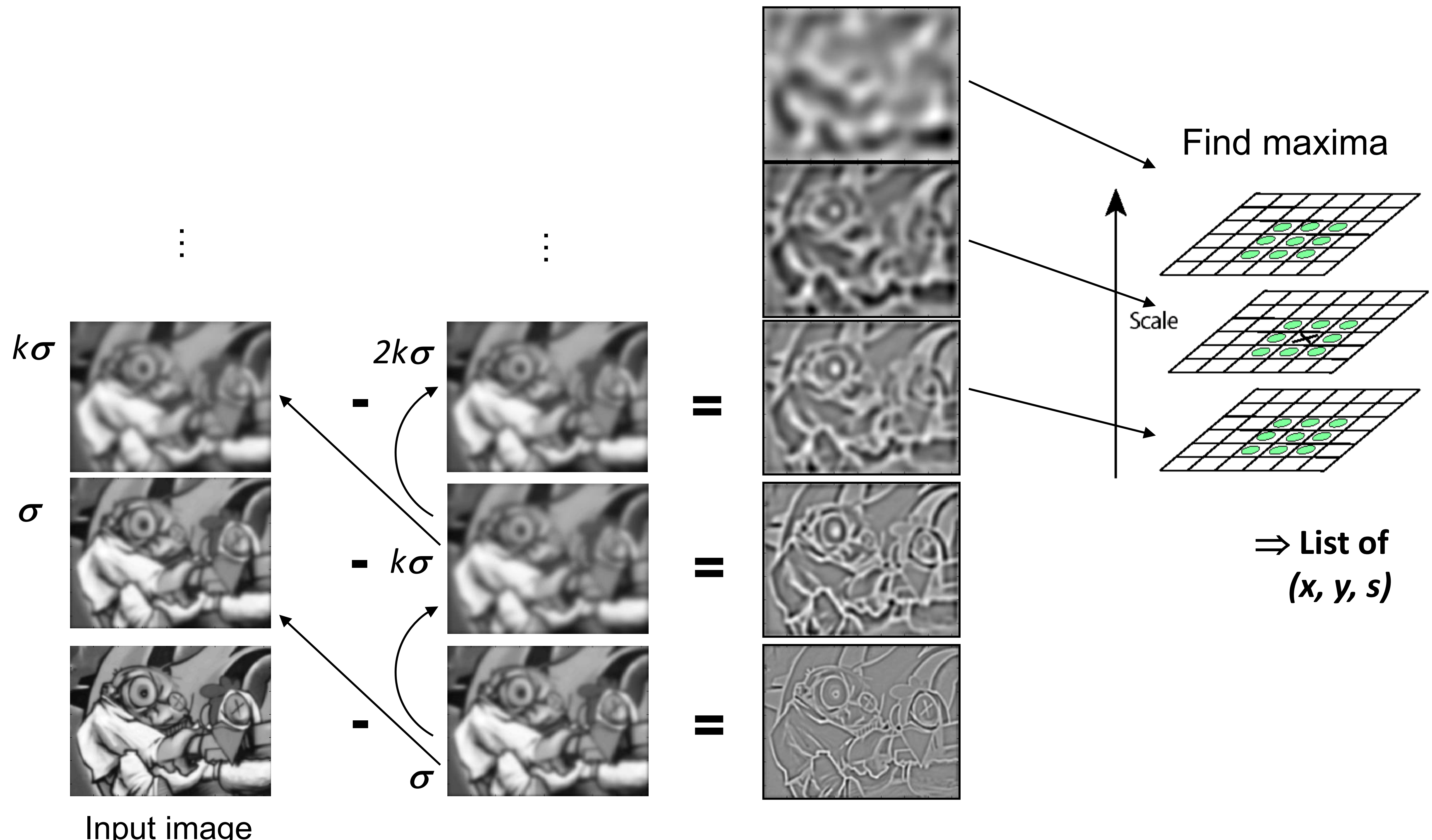
Approximate LoG with Difference-of-Gaussian (DoG).

1. Blur image with σ Gaussian kernel
2. Blur image with *k*σ Gaussian kernel
3. Subtract 2. from 1.



K. Grauman, B. Leibe

Find local maxima in position-scale space of DoG

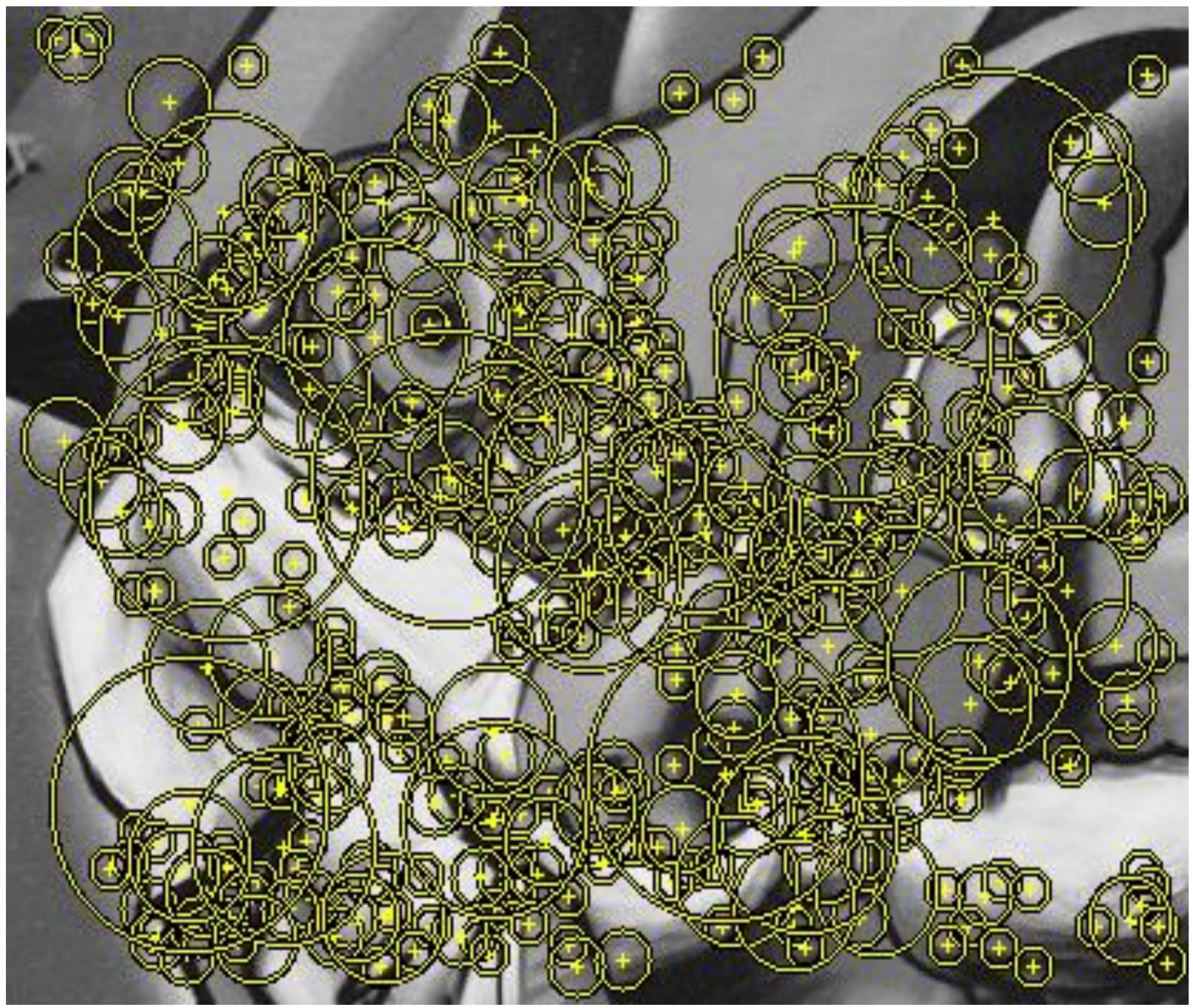


$\Rightarrow$ **List of *(x, y, s)***

K. Grauman, B. Leibe

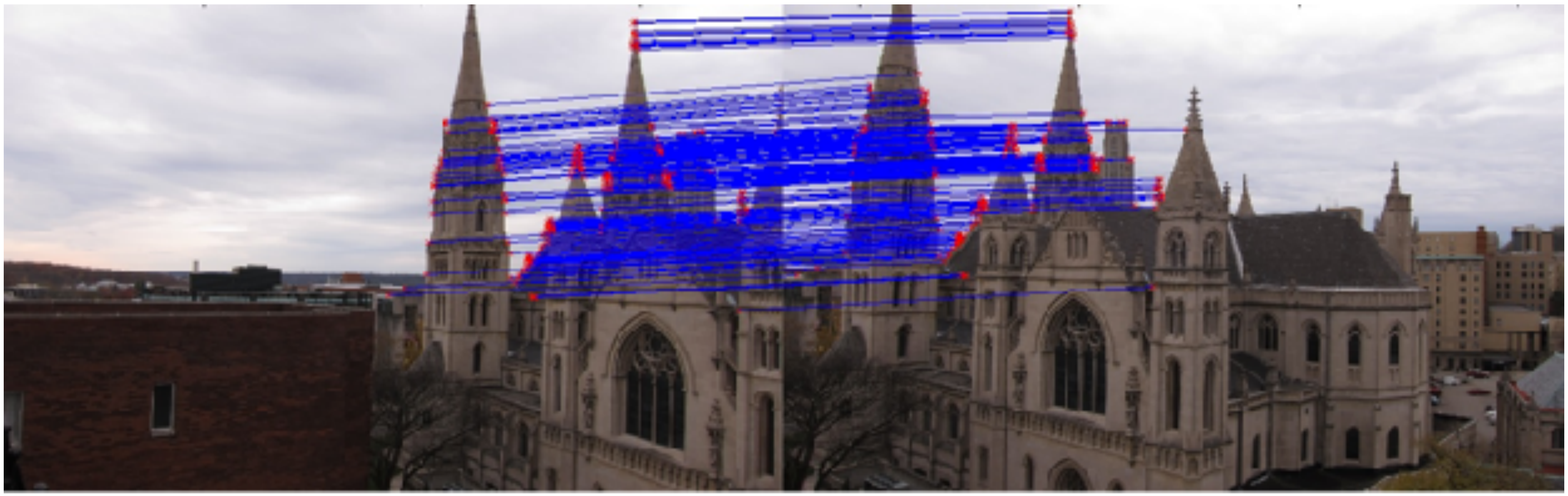
Results: Difference-of-Gaussian

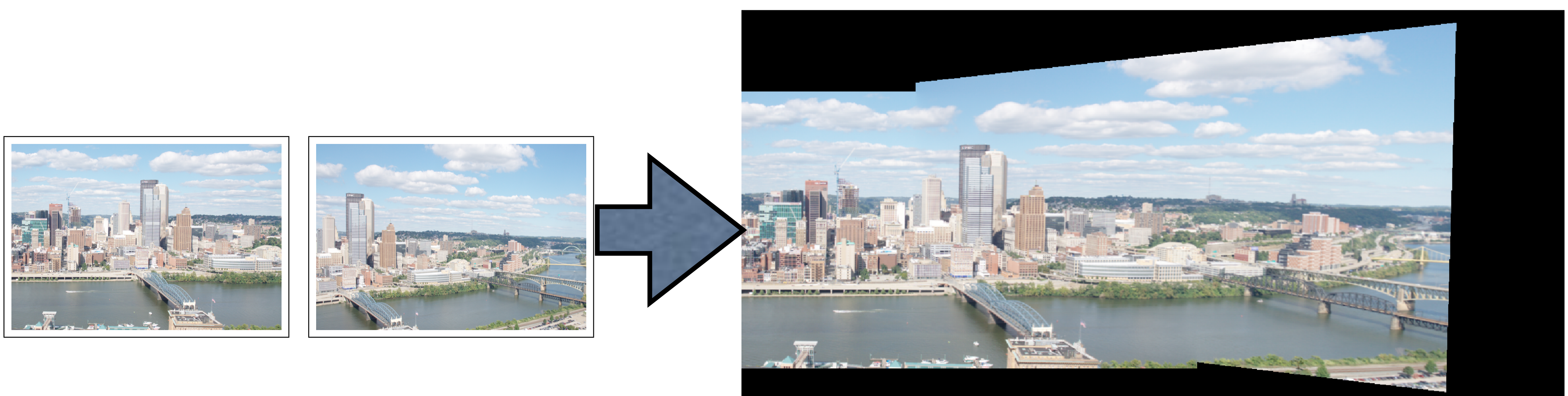
- Larger circles = larger scale
- Descriptors with maximal scale response



K. Grauman, B. Leibe

Core visual understanding task: finding correspondences between images





Source: Deva Ramanan

SIFT

Scale Invariant Feature Transform

Distinctive Image Features from Scale-Invariant Keypoints

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

January 5, 2004

IJCV 04

48,547 citations!

SIFT

Scale Invariant Feature Transform

**Distinctive Image Features
from Scale-Invariant Keypoints**

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

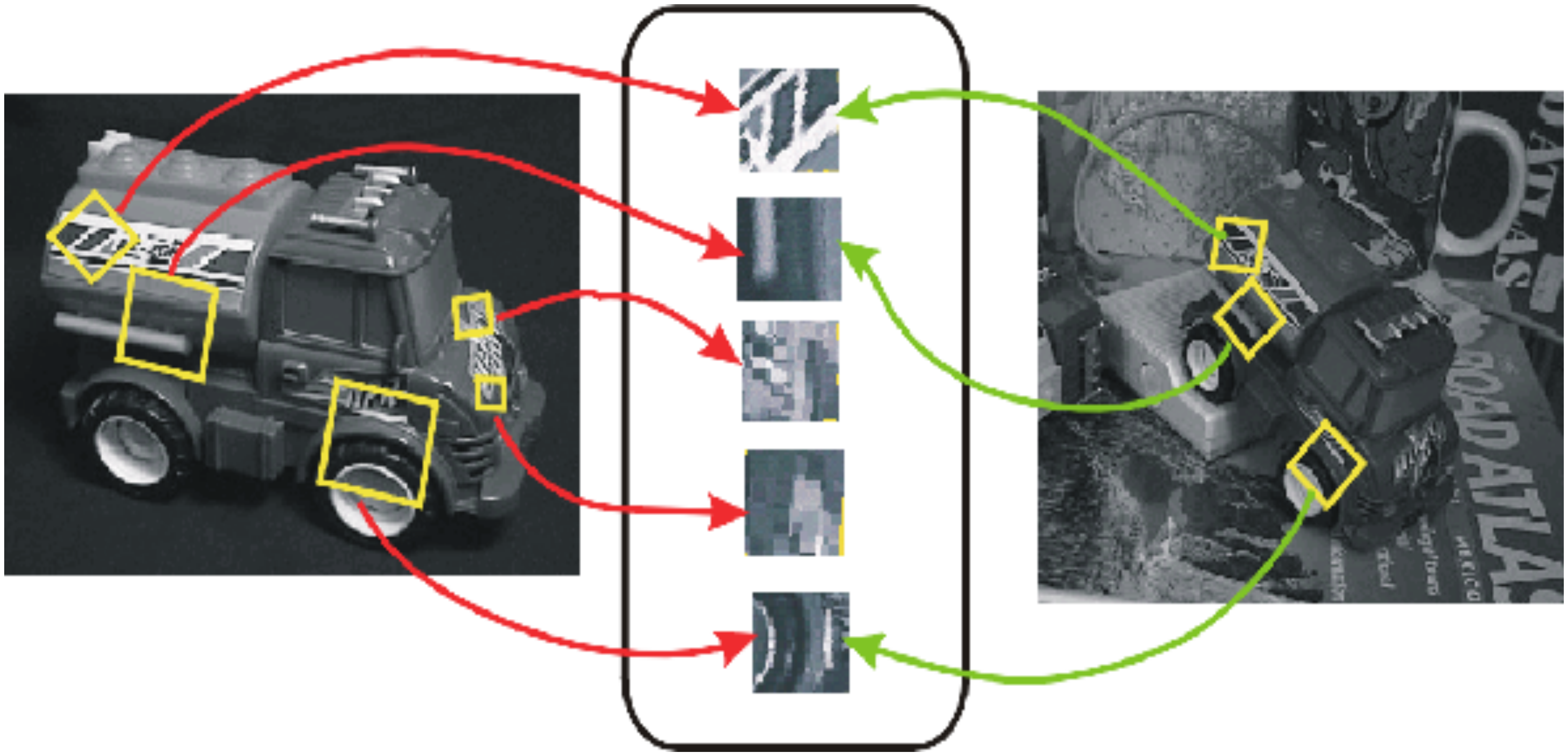
January 5, 2004

IJCV 04

48,563 citations!

~~**48,547 citations!**~~

Coordinate frames

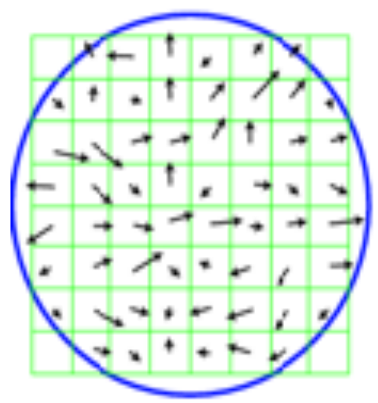


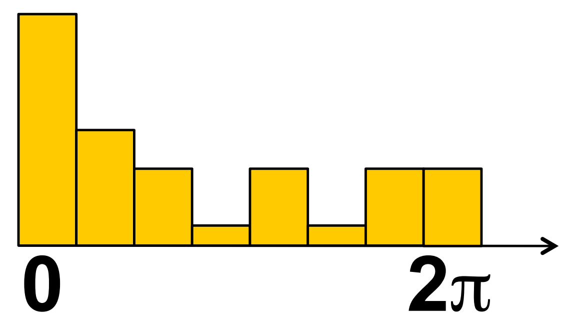
Represent each patch in a canonical scale and orientation (or general *affine* coordinate frame)

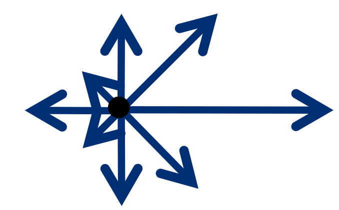
Source: Deva Ramanan

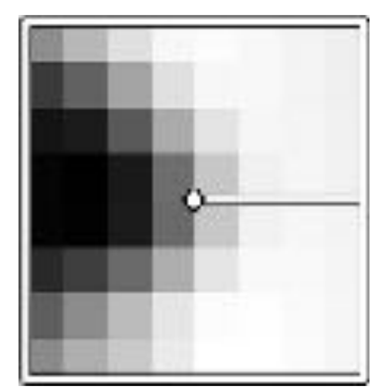
Find dominant orientation

Compute gradients for all pixels in patch. Histogram (bin) gradients by orientation



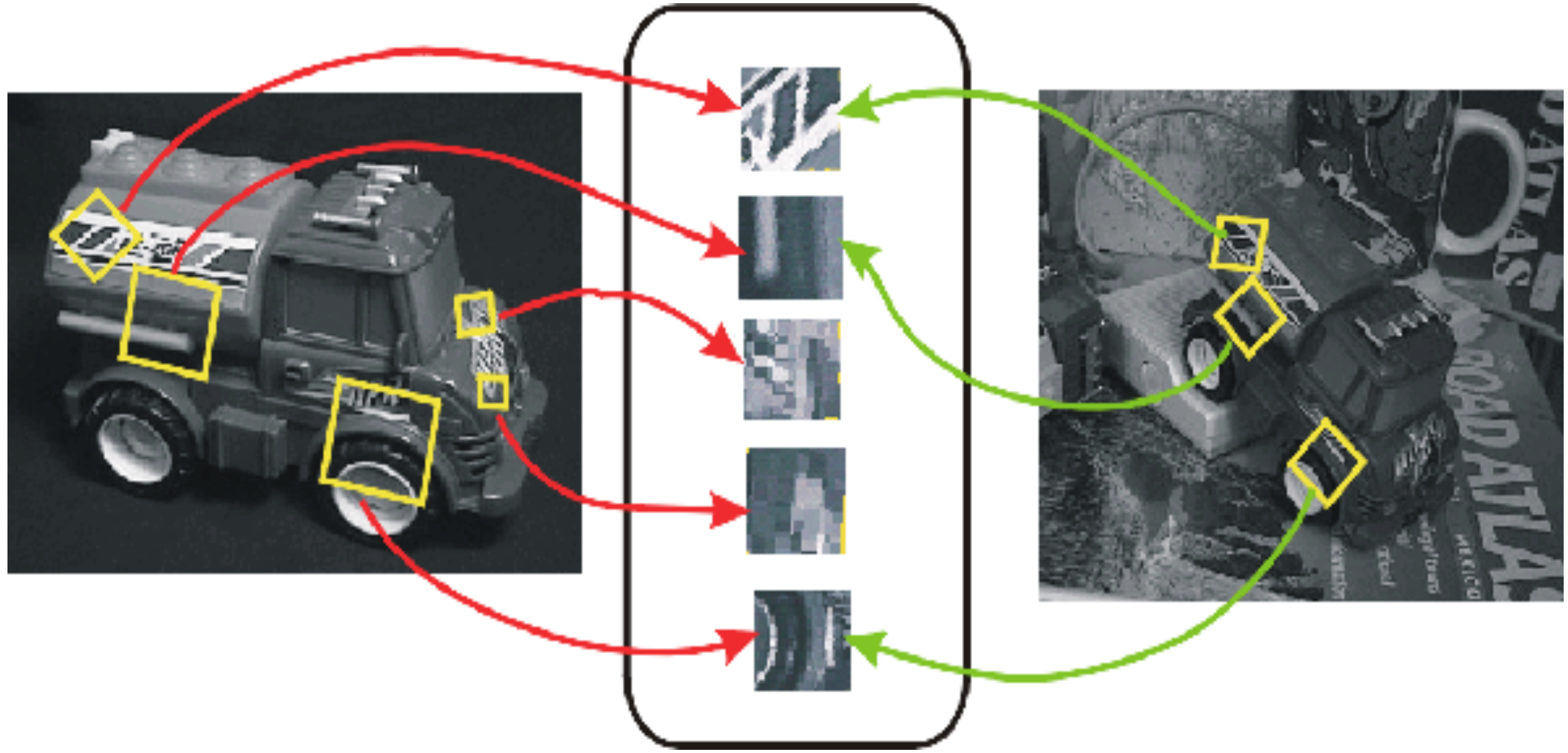






Source: Deva Ramanan

Appearance descriptors



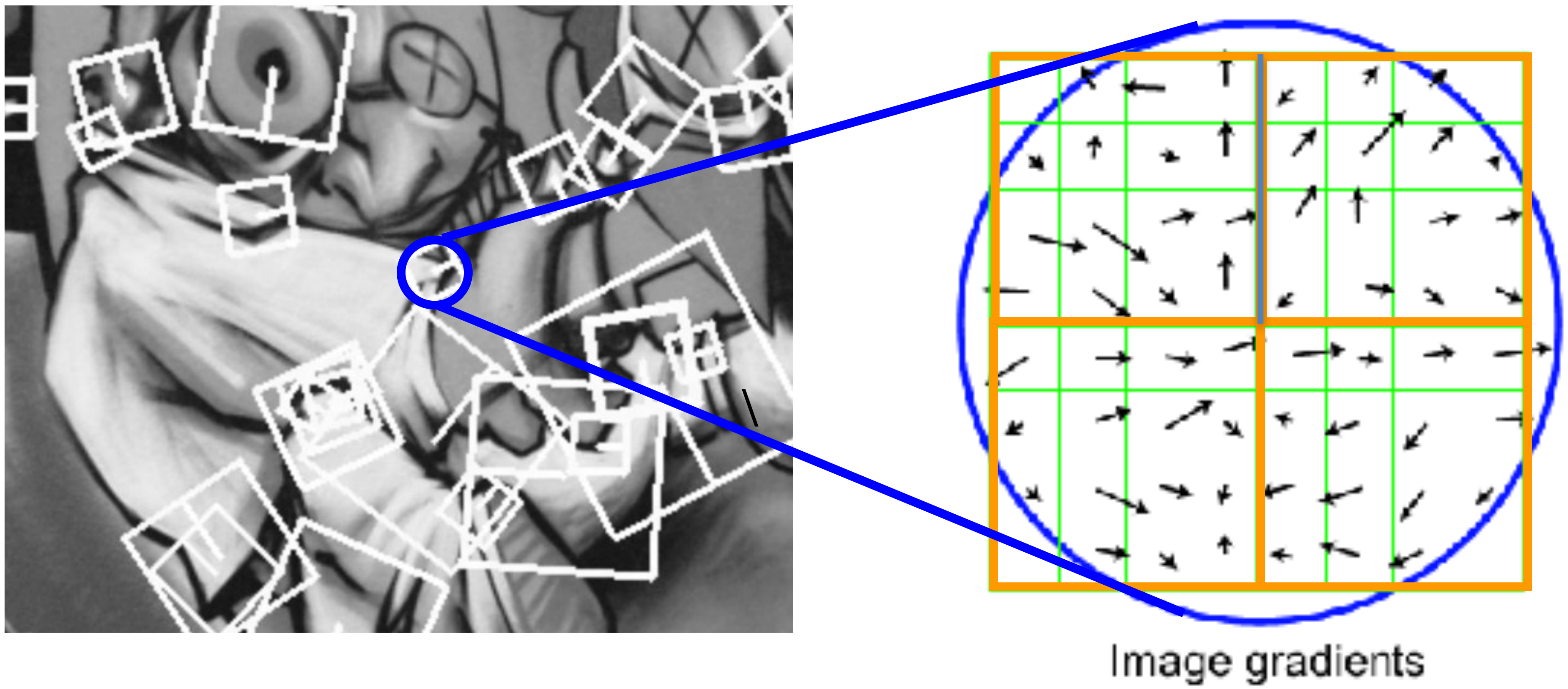
Represent each patch in a canonical scale and orientation (or general *affine* coordinate frame)

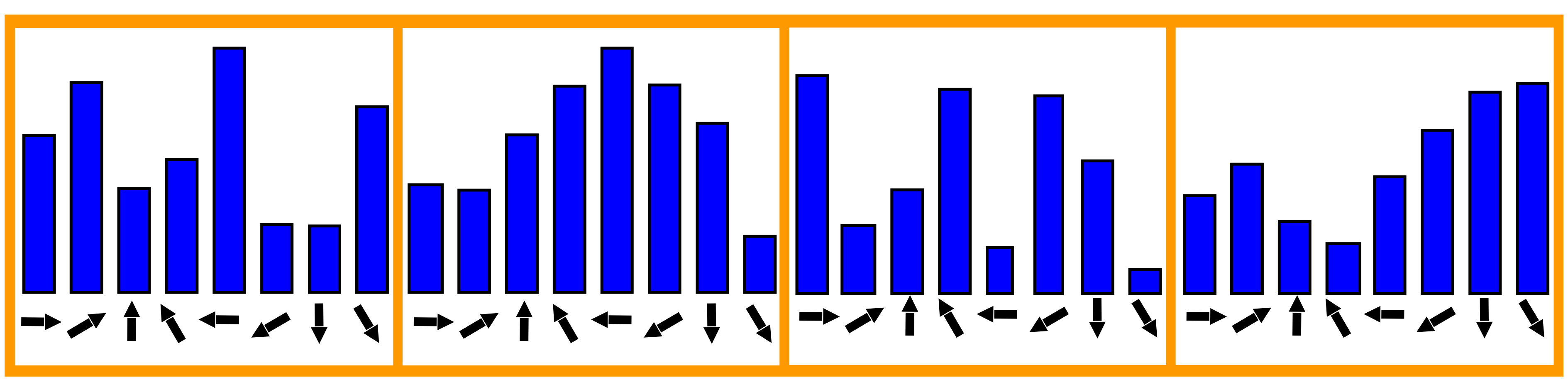
$$d(p_1, p_2) = \left\| \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} - \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} \right\|$$

Source: Deva Ramanan

Computing the SIFT Descriptor

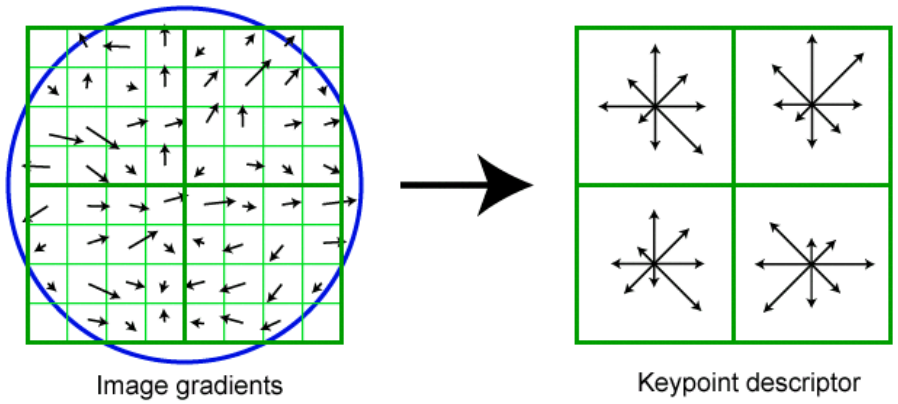
Histograms of gradient directions over spatial regions





Source: Deva Ramanan

Post-processing



1. Rescale 128-dim vector to have unit norm

$$x = \frac{x}{||x||}, \quad x \in R^{128}$$

"invariant to linear scalings of intensity"

2. Clip high values

$$x := \min(x, .2)$$
$$x := \frac{x}{||x||}$$

approximate binarization allows for for flat
patches with small gradients to remain stable

Source: Deva Ramanan

Evaluation

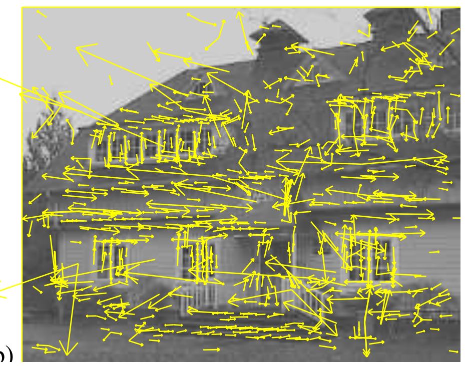
Historic problem in computer vision:
"wide-baseline matching"

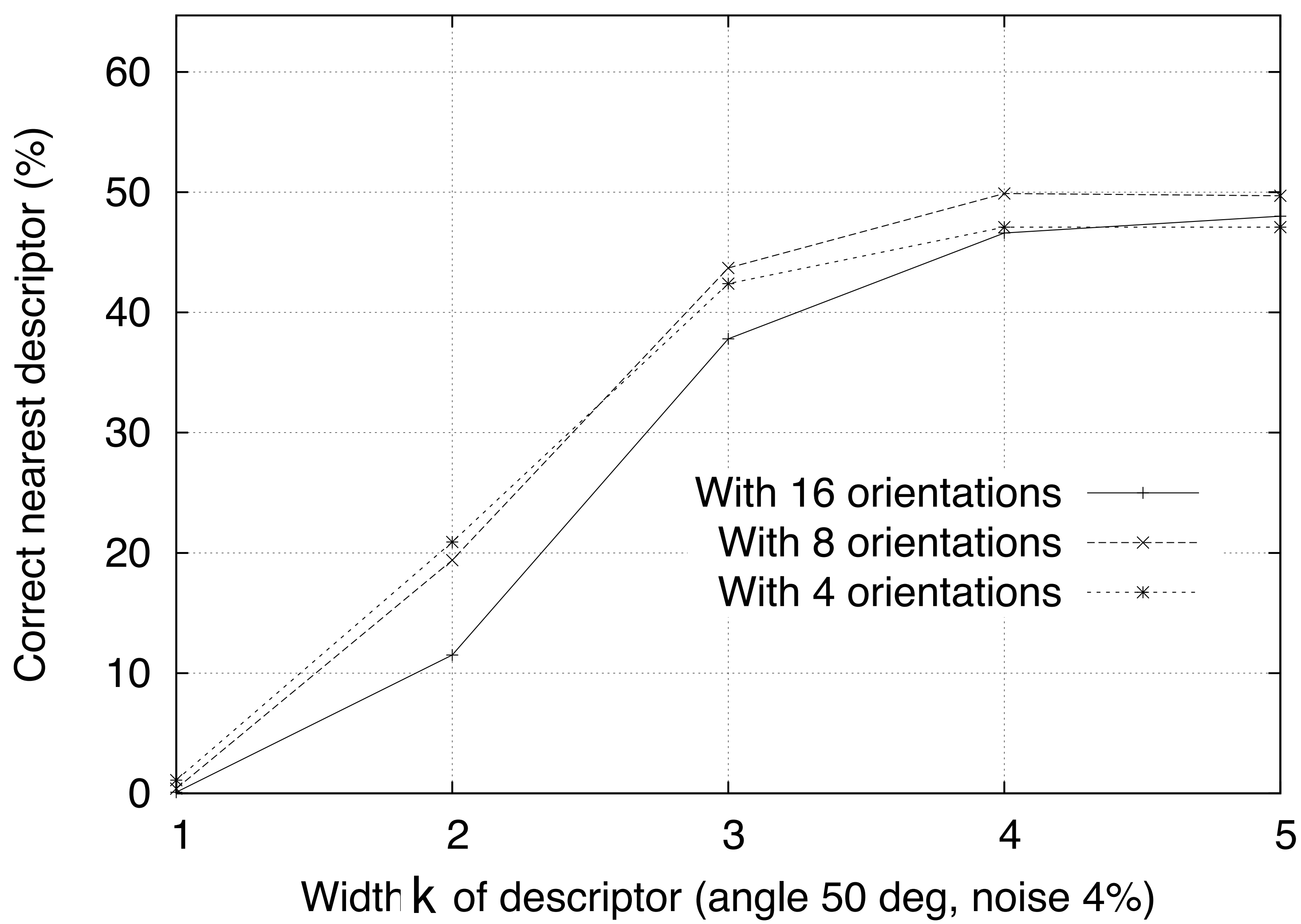
Source: Deva Ramanan

SIFT

What made this work? Exhaustive evaluation of hyper-parameters on annotated dataset

(b)





Source: Deva Ramanan

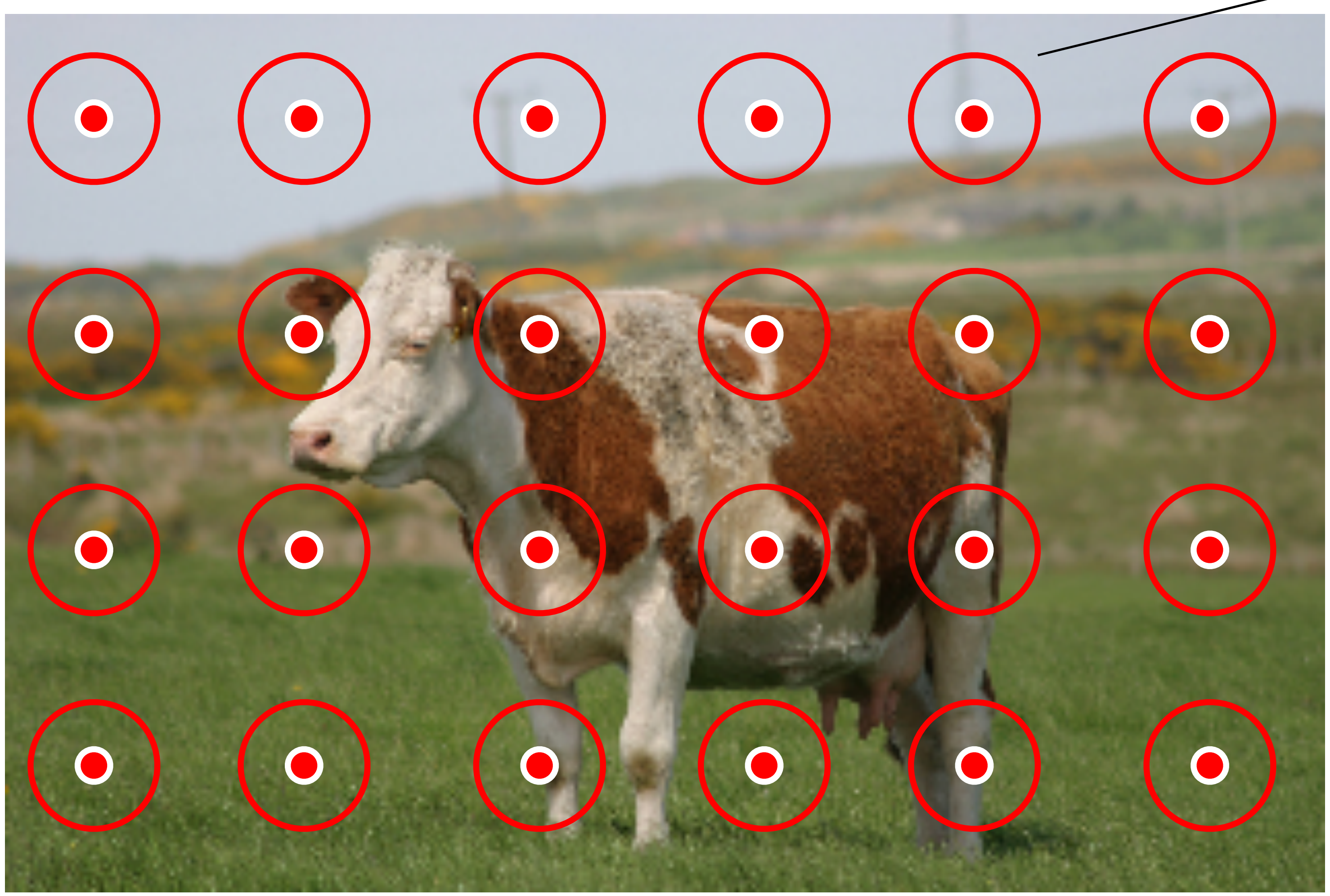
Properties of SIFT

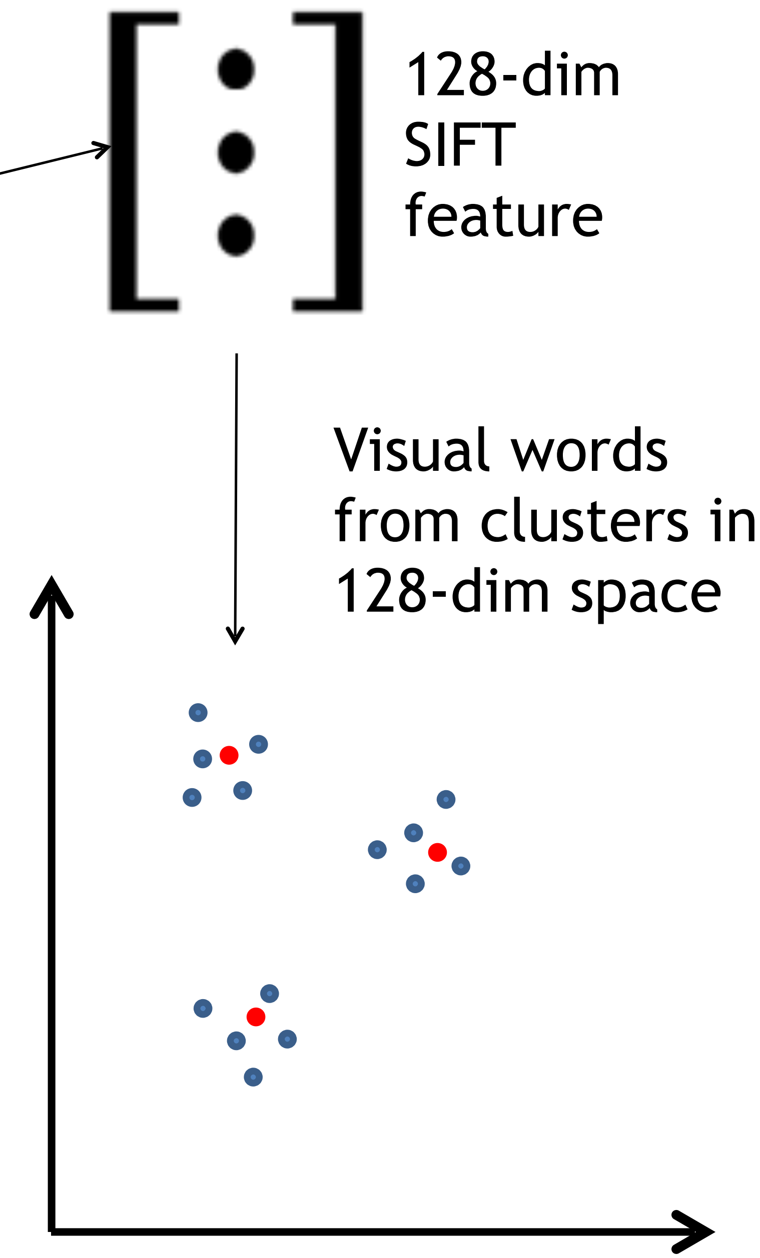
Extraordinarily robust matching technique

- Can handle changes in viewpoint
  - Up to about 60 degree out of plane rotation
- Can handle significant changes in illumination
  - Sometimes even day vs. night (below)
- Fast and efficient—can run in real time
- Lots of code available
  - http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known_implementations_of_SIFT

Dense sampling

- So far: Descriptors of patches centered at sparse interest points
- But we can use the descriptors at any point
- Common case:
  - Regularly sampled grid of points
  - Dense SIFT (or LBP, or…)



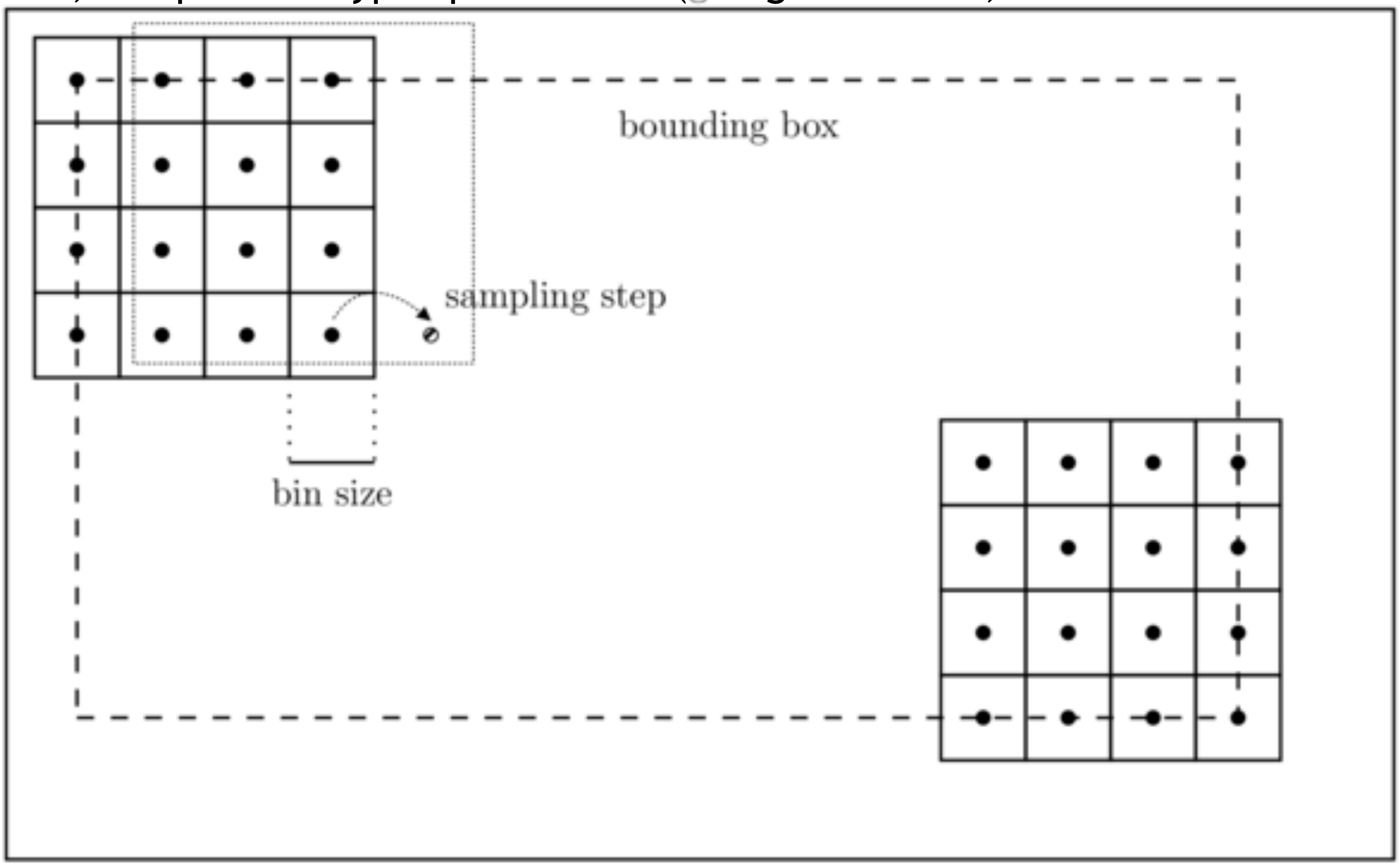


Source: Deva Ramanan

HOG

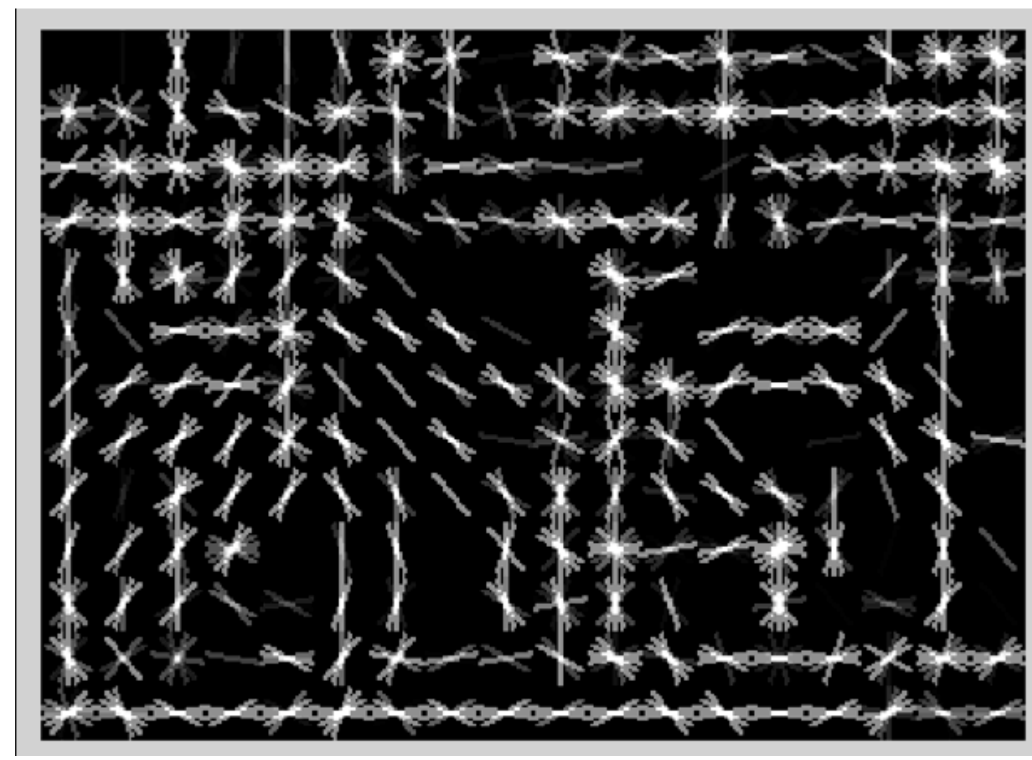
Compute SIFT descriptors on a grid equal to size of individual “cell”

In practice, re-optimize hyper-parameters (2x2 grid of cells, with each cell of 8x8 pixels



Source: Deva Ramanan

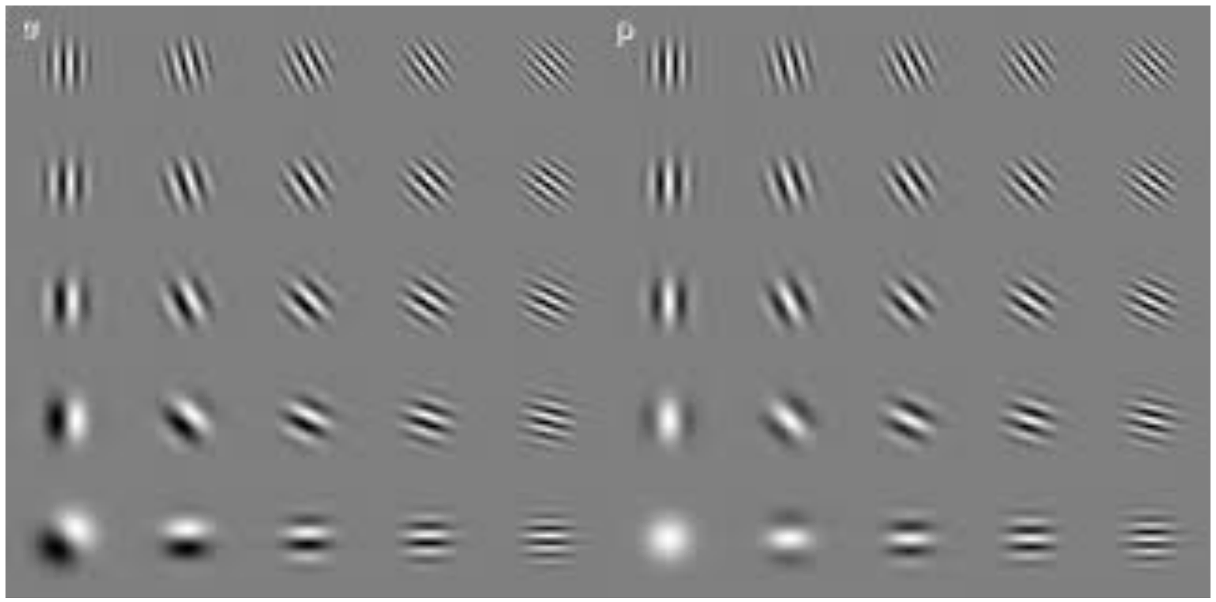
Common visualization

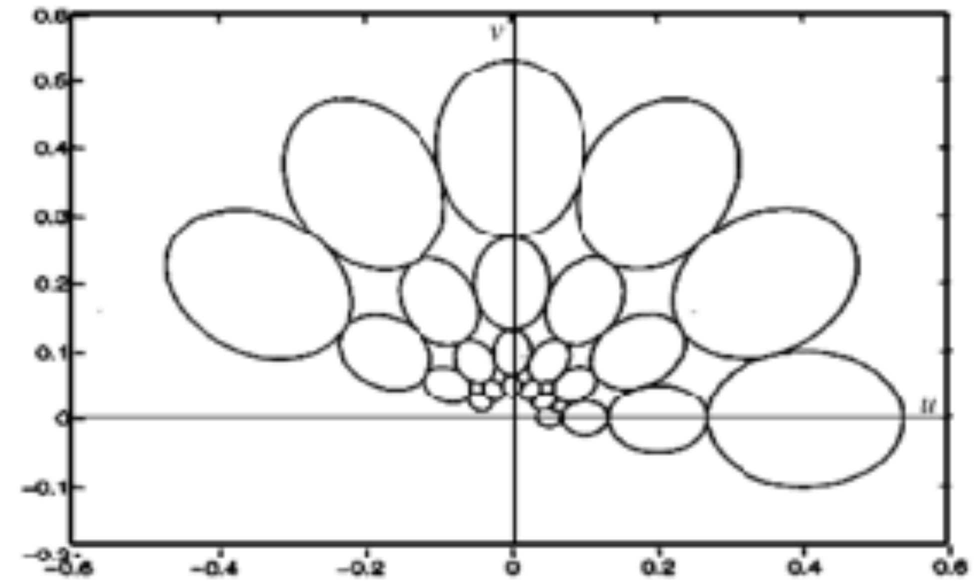


Source: Deva Ramanan

Alternative global desciptor: Gist

Oliva and Torralba, 2001





1.Compute frequency energy (magnitude) at each spatial (x,y) location with gabor filters

2. Average energy over 4x4 spatial grids

8	orientations
4	scales
x 16	spatial bins
512	dimensions

Source: Deva Ramanan


Chair


Car


Aeroplane

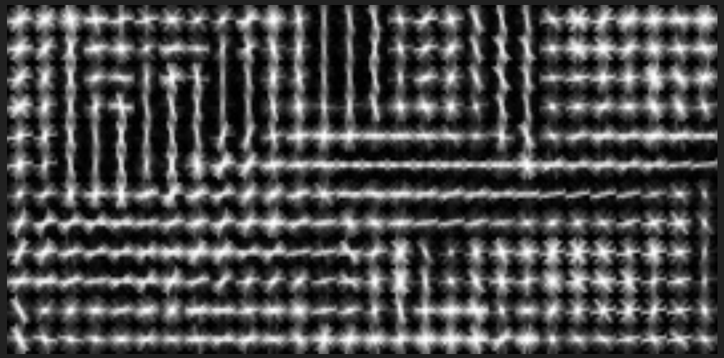

Car


Car

What information does HOG have?

Image

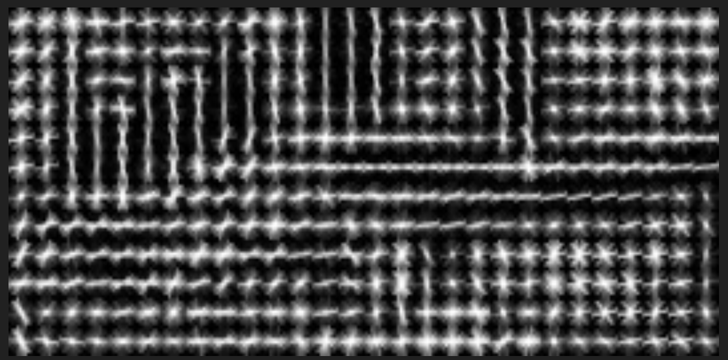
HOG

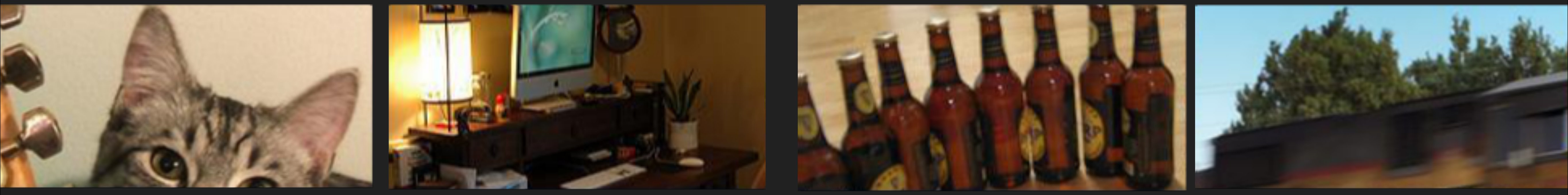


What information does HOG have?

Image

HOG



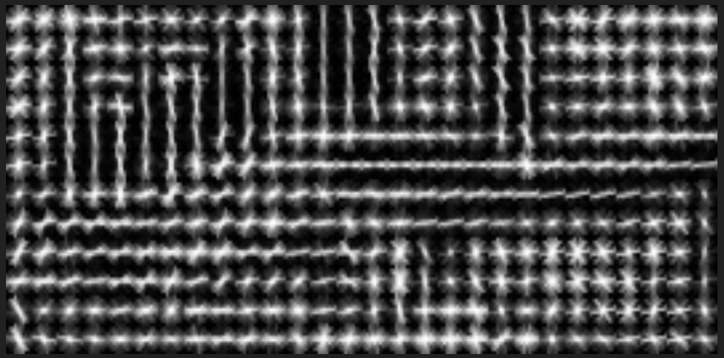


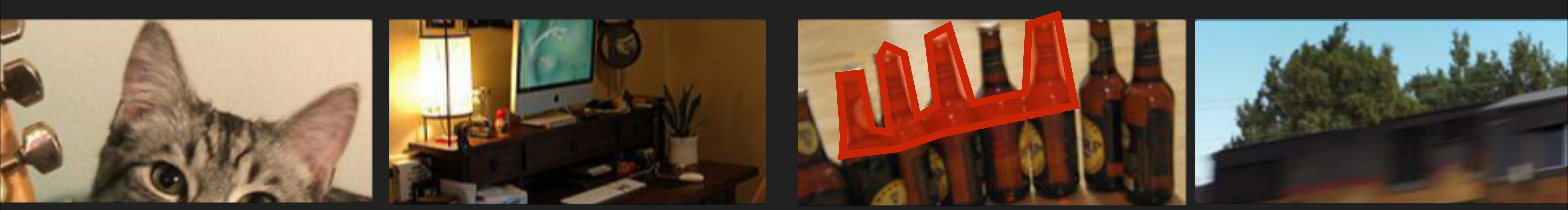
Nearest Neighbors

What information does HOG have?

Image

HOG

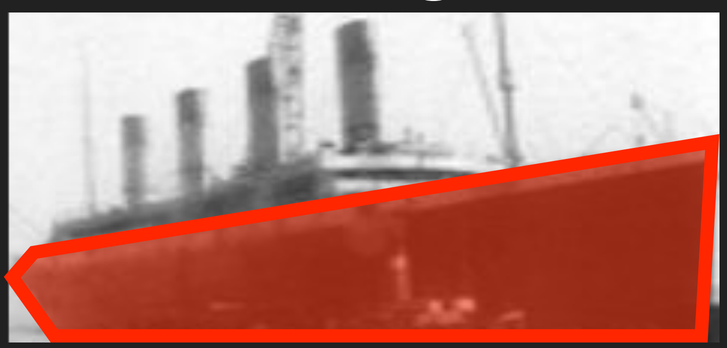




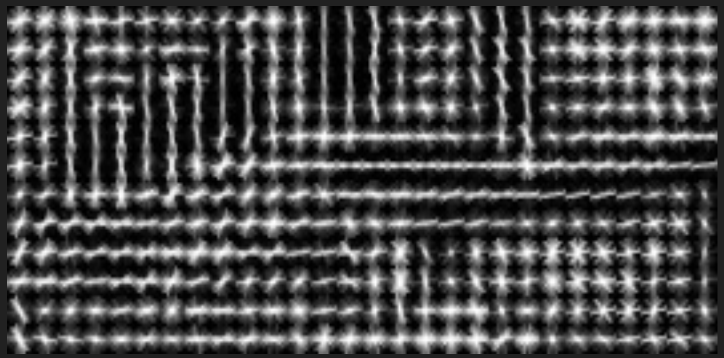
Nearest Neighbors

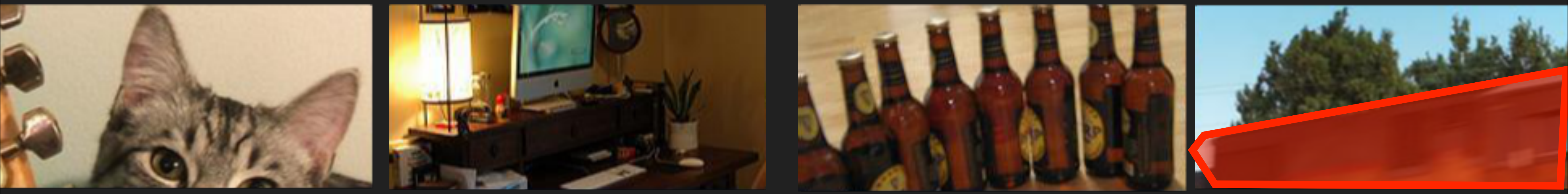
What information does HOG have?

Image



HOG



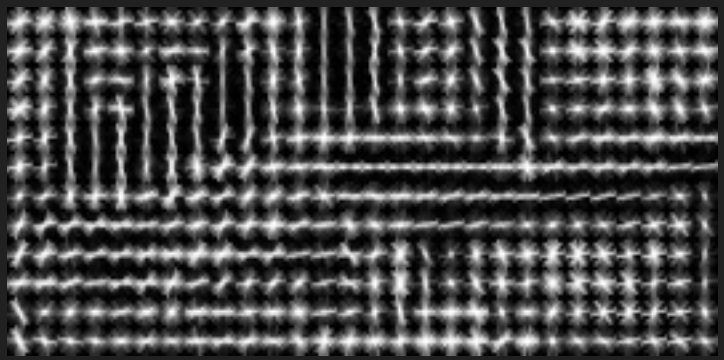


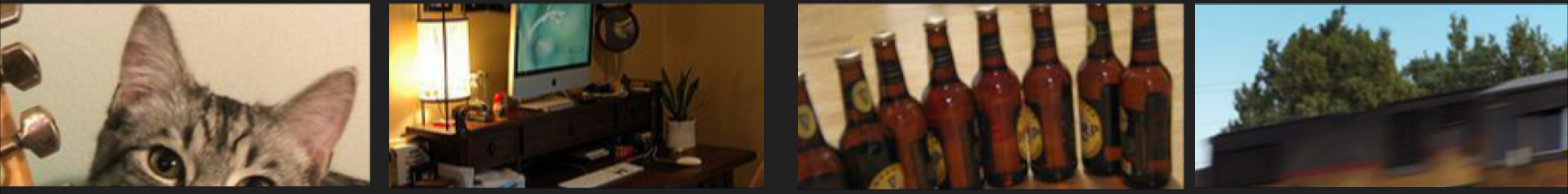
Nearest Neighbors

What information does HOG have?

Image

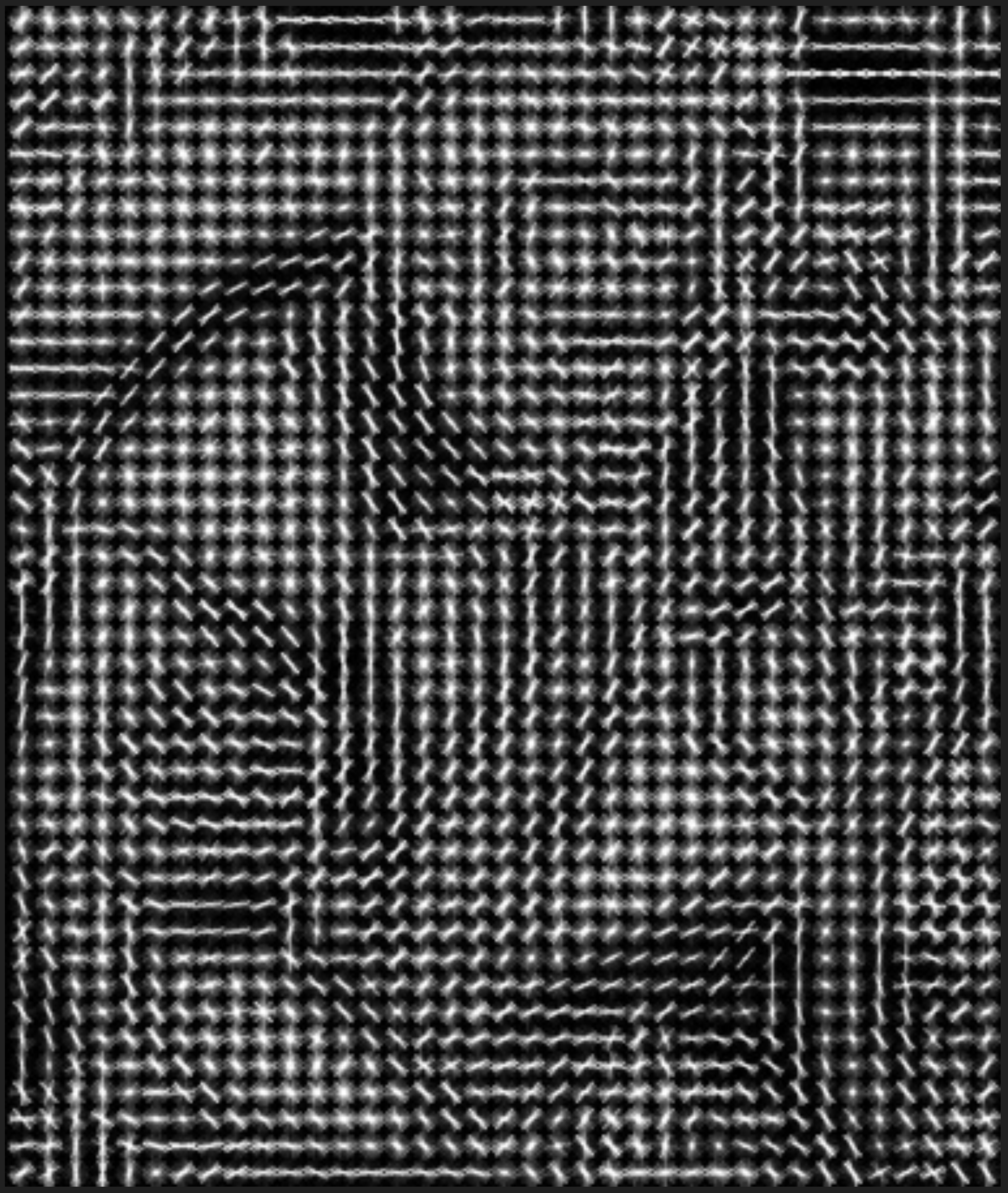
HOG





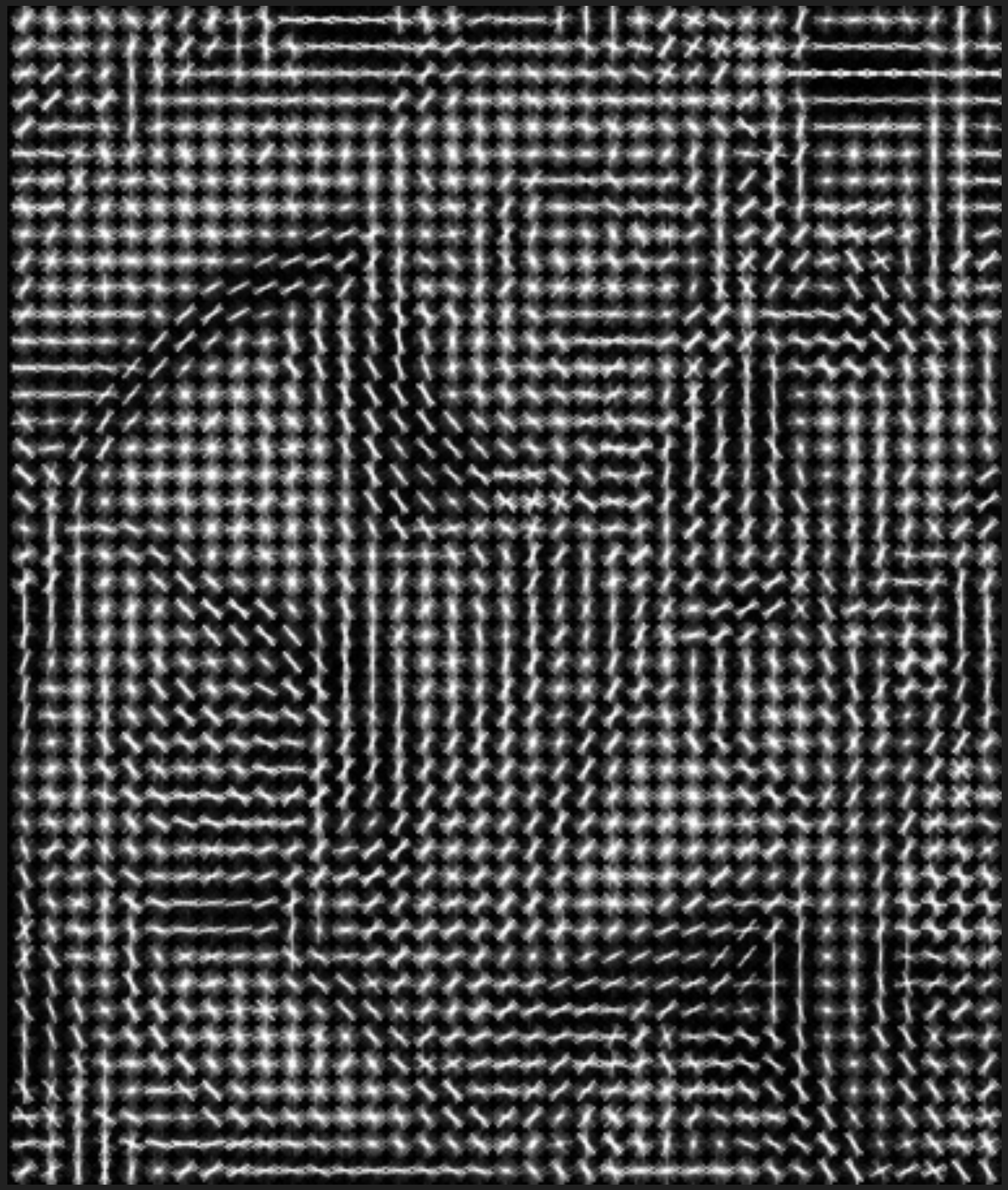
Nearest Neighbors

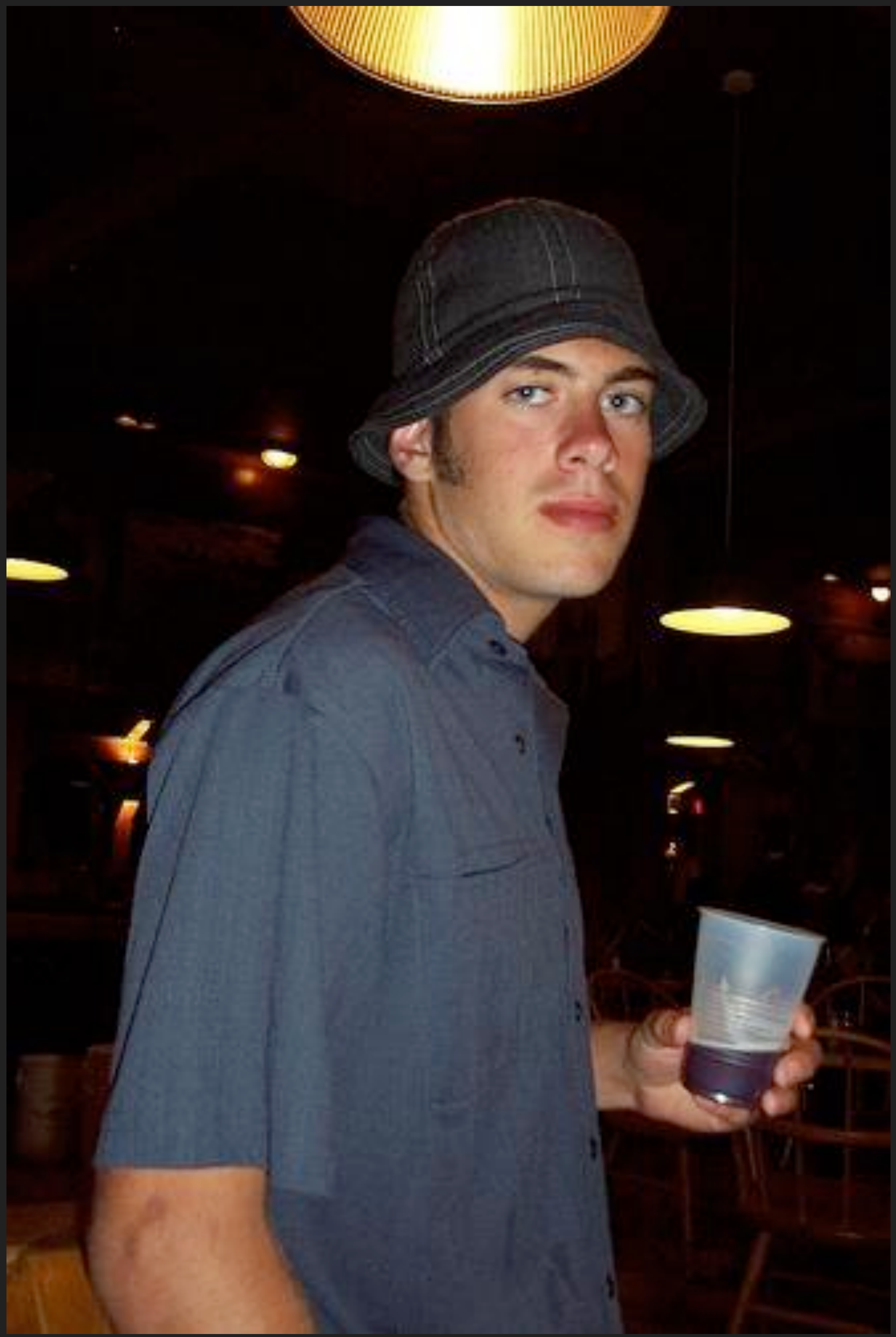
What information is lost?



What information is lost?

$$\min_{x \in \mathbb{R}^d} ||\phi(x) - y||_2^2$$





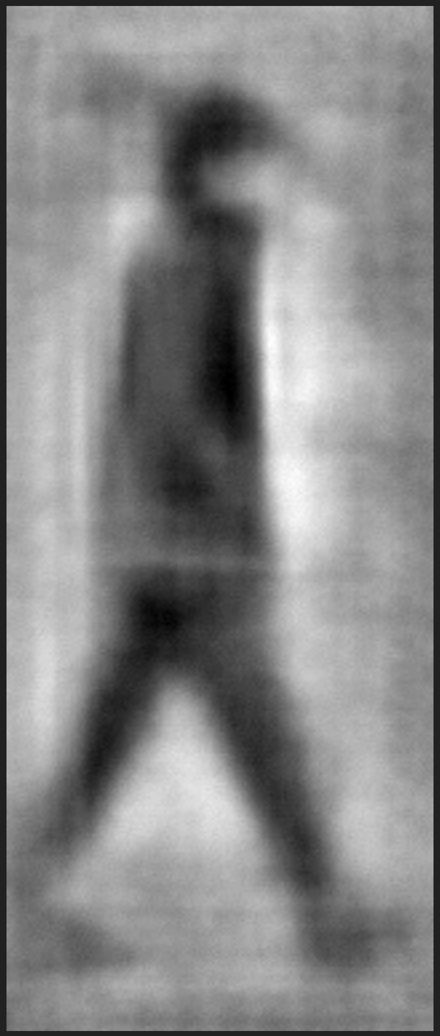
vs

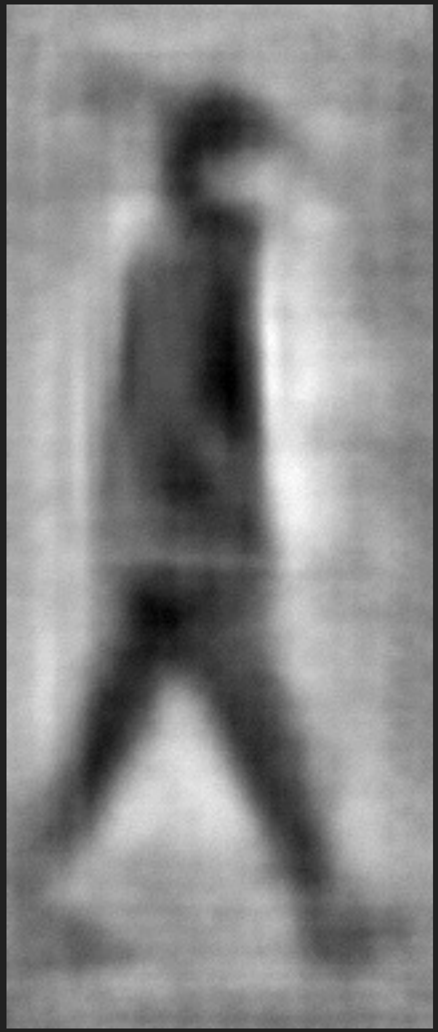
Human Vision

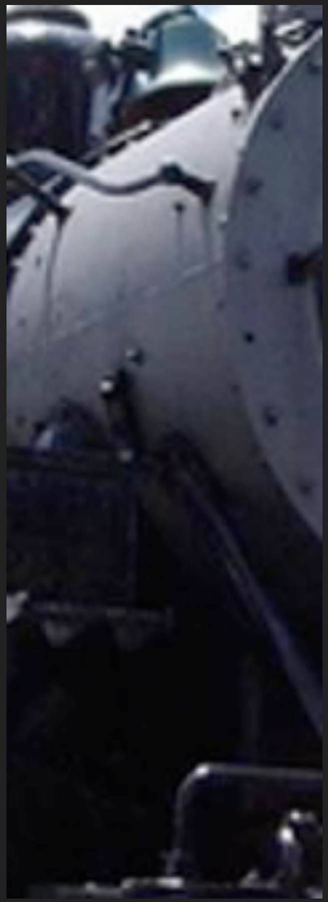
HOG Vision

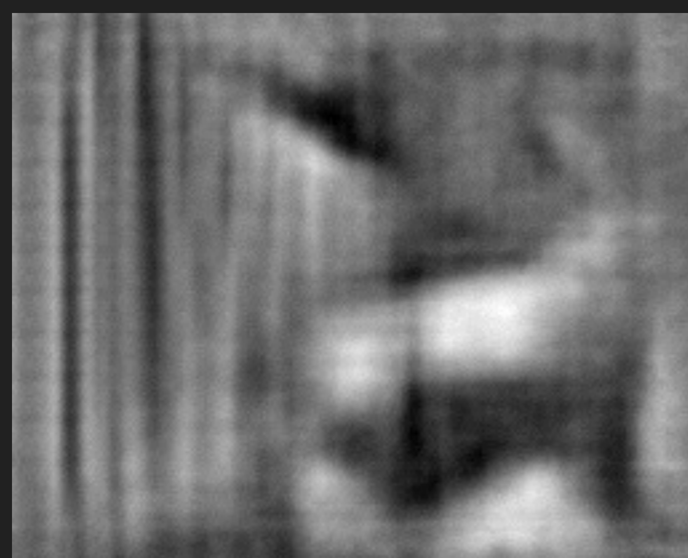
The HOGgles Challenge

Clap your hands when you see a person



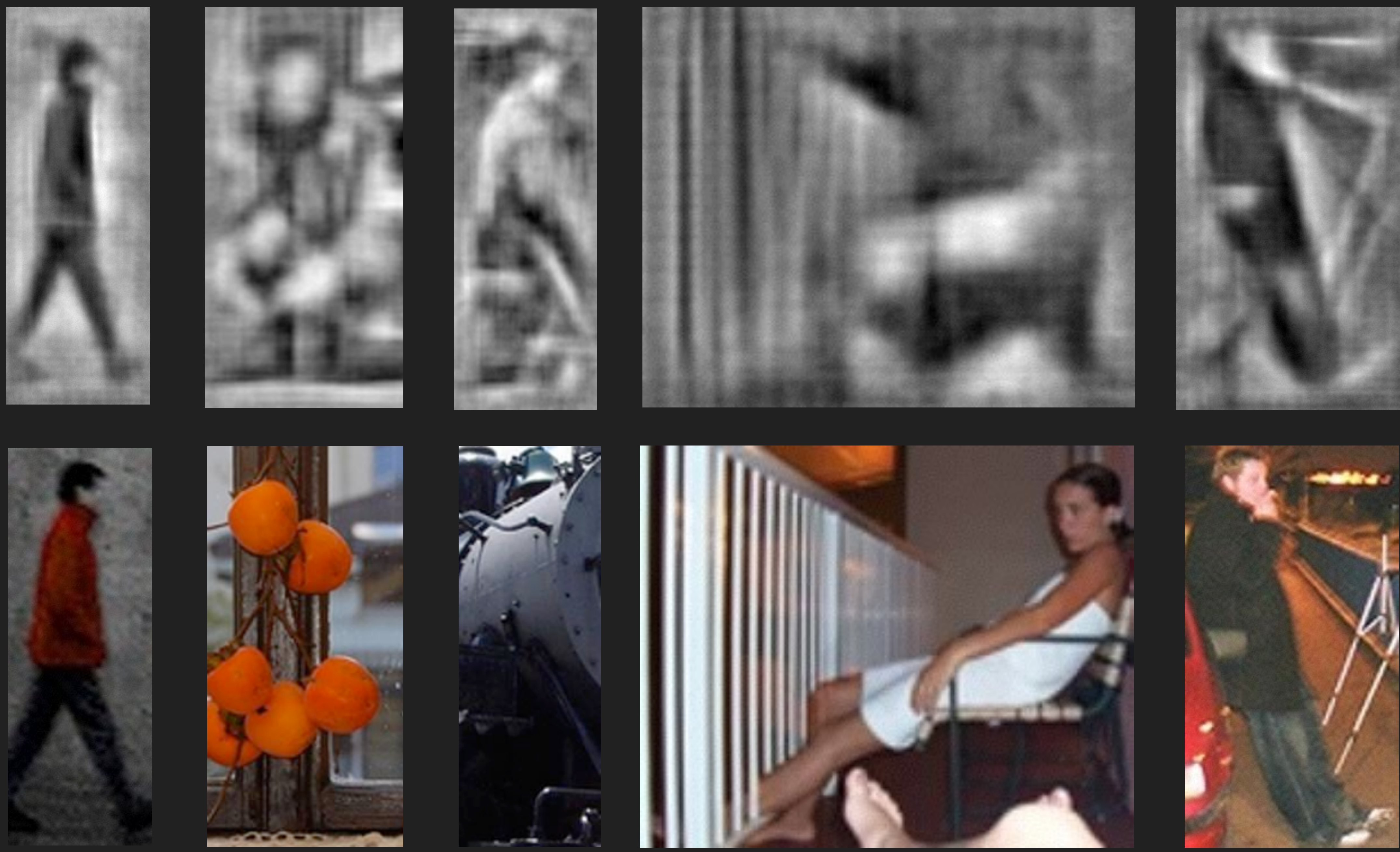


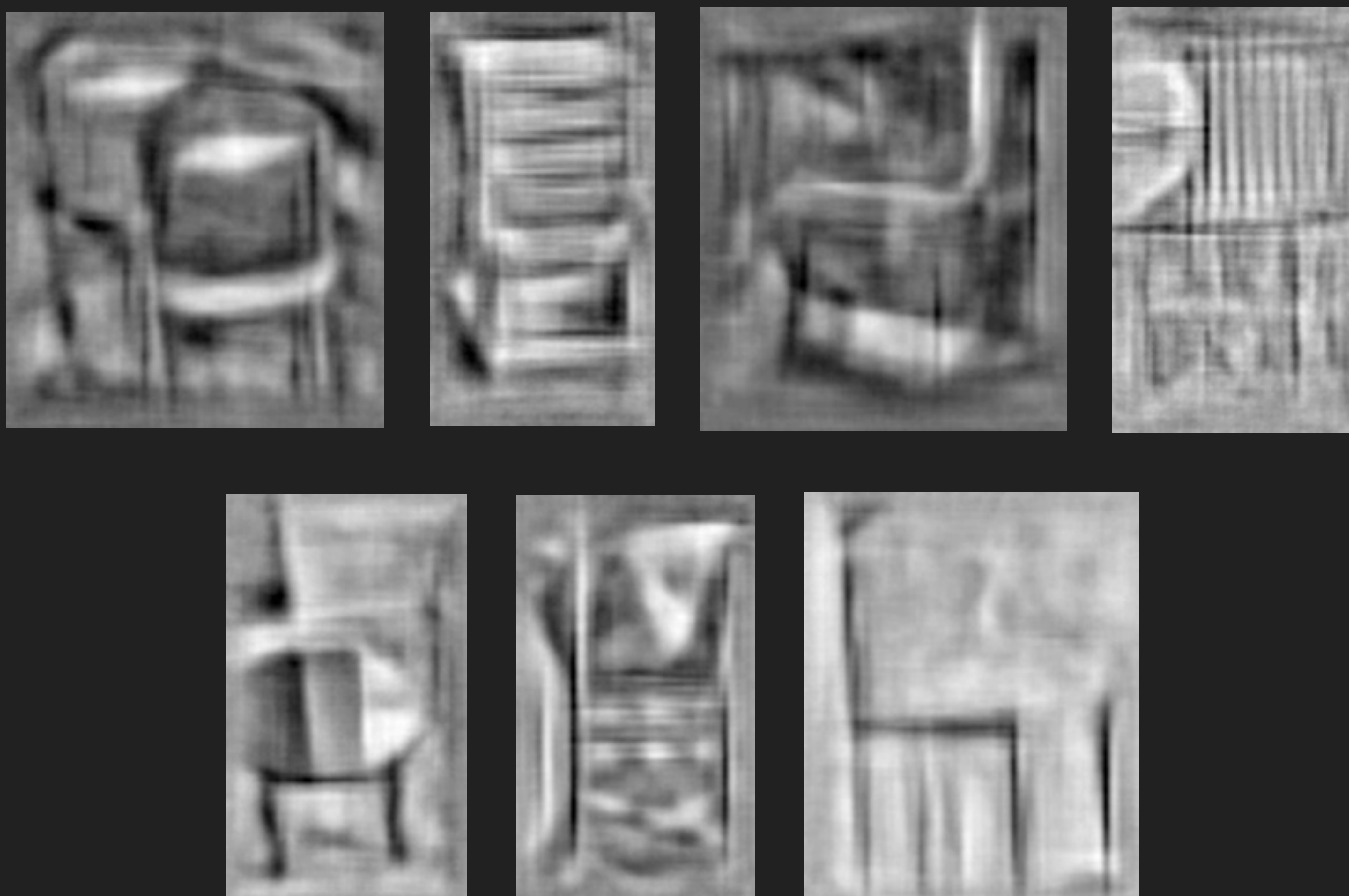




The HOGgles Challenge



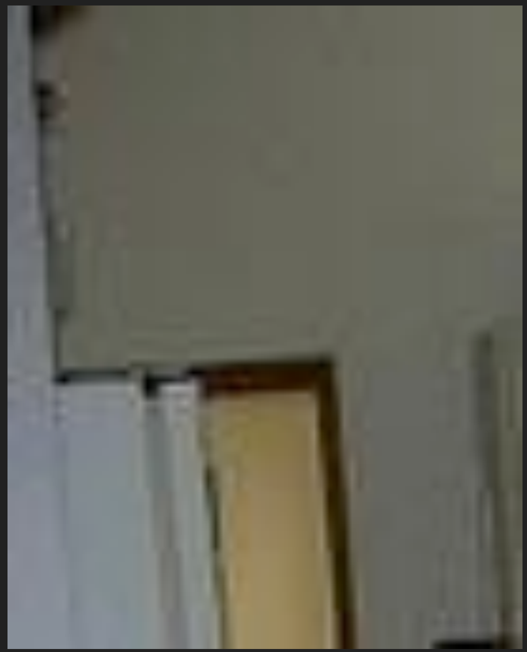
Chair Detections



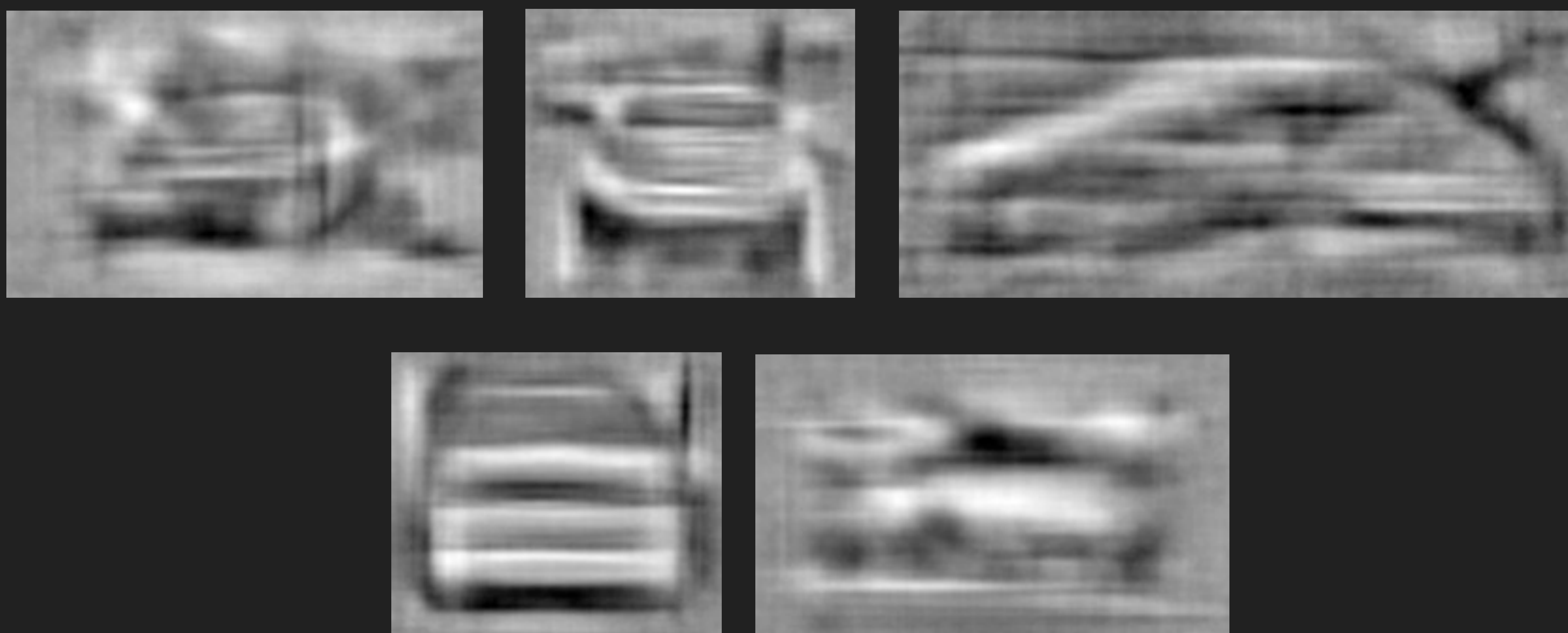
Chair Detections





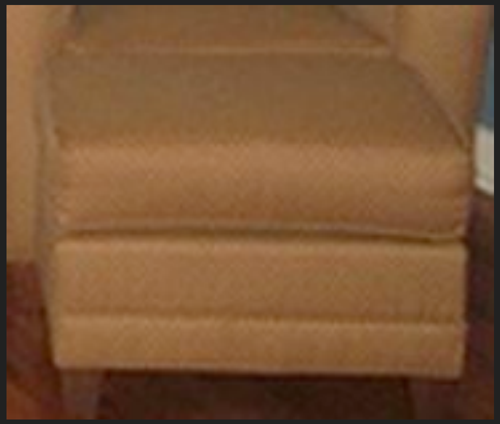



Car Detections



Car Detections




Why did the detector fail?