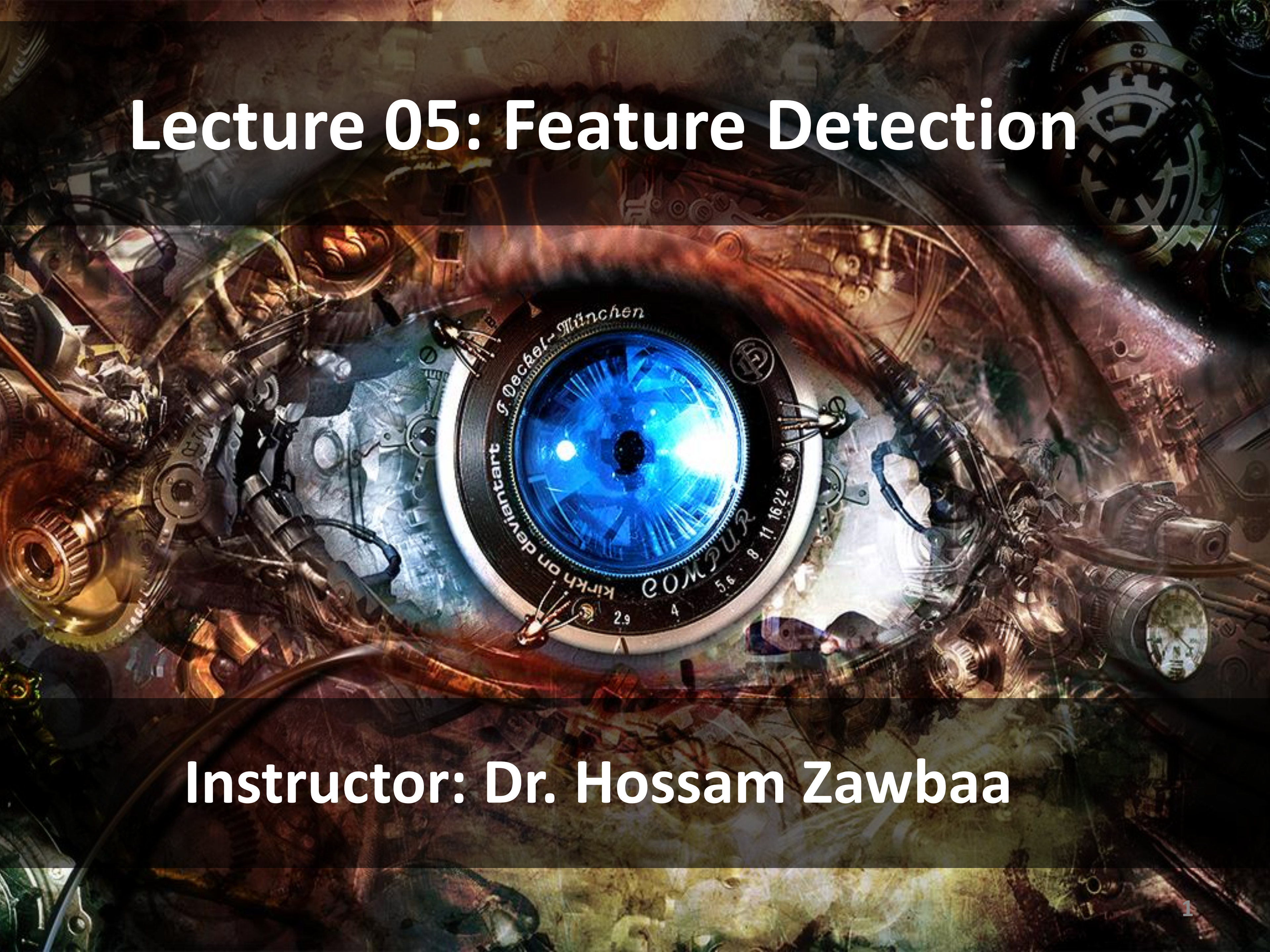
Lecture 05: Feature Detection



Instructor: Dr. Hossam Zawbaa

- What points would you choose?

Many Existing Detectors Available

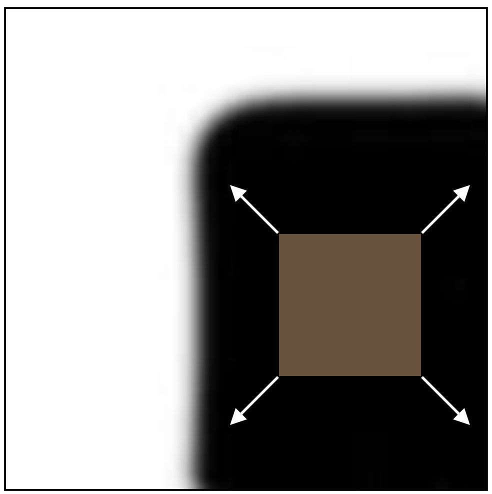
Hessian & Harris	[Beaudet '78], [Harris '88]
Laplacian, DoG	[Lindeberg '98], [Lowe 1999]
Harris-/Hessian-Laplace	[Mikolajczyk & Schmid '01]
Harris-/Hessian-Affine	[Mikolajczyk & Schmid '04]
EBR and IBR	[Tuytelaars & Van Gool '04]
MSER	[Matas '02]
Salient Regions	[Kadir & Brady '01]
Others…	

Feature extraction: Corners

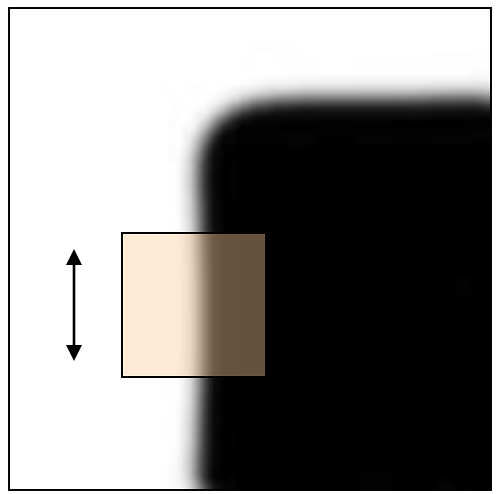
Slides from Rick Szeliski, Svetlana Lazebnik, and Kristin Grauman

Corners Detection: Basic Idea

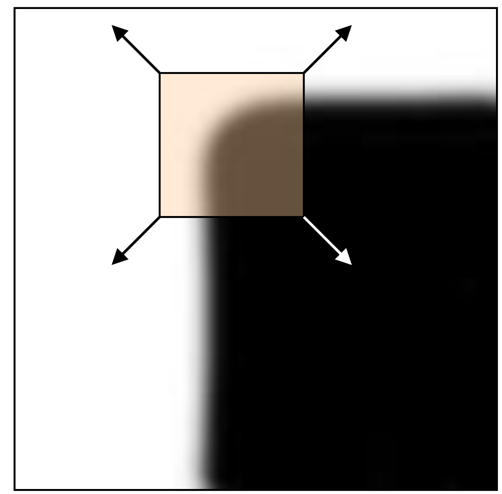
- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



"flat" region:
no change in
all directions



"edge":
no change
along the edge
direction



"corner":
significant
change in all
directions

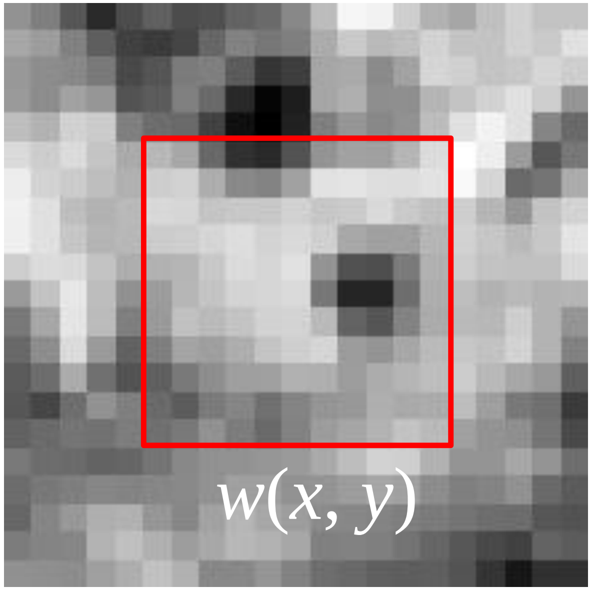
Source: A. Efros

Corners Detection: Mathematics

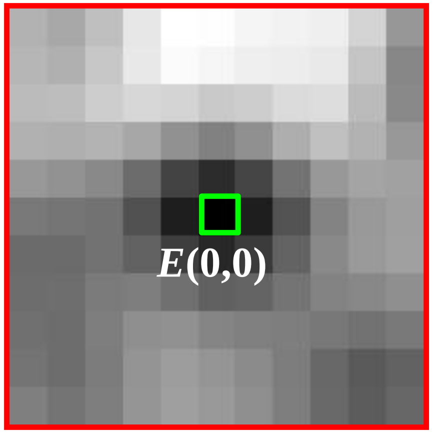
Change in appearance of window $w(x,y)$
for the shift $[u,v]$:

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

$I(x, y)$



$E(u, v)$



Corners Detection: Mathematics

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

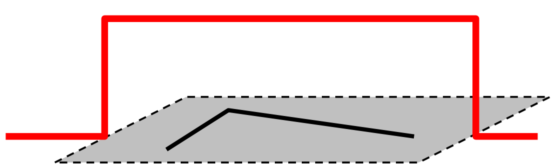
$$E(u,v)=\sum_{x,y} w(x,y)\left[I(x+u,y+v)-I(x,y)\right]^2$$

Window function

Shifted intensity

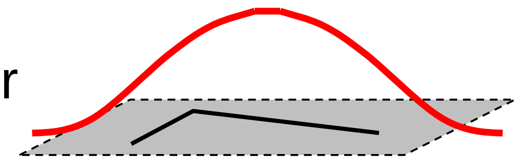
Intensity

Window function $w(x,y)$ =



1 in window, 0 outside

or



Gaussian

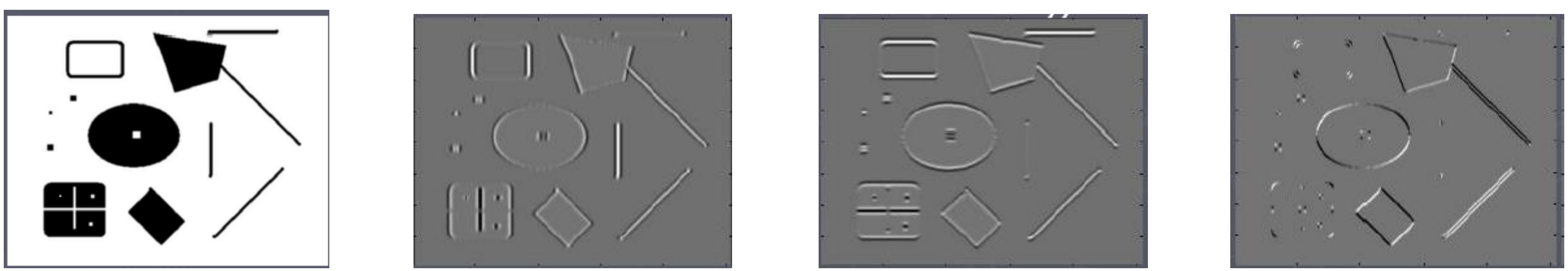
Source: R. Szeliski

Corners Detection

Corners as distinctive interest points

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

2 x 2 matrix of image derivatives (averaged in neighborhood of a point).



Notation: $I_x \Leftrightarrow \frac{\partial I}{\partial x}$ $I_y \Leftrightarrow \frac{\partial I}{\partial y}$ $I_x I_y \Leftrightarrow \frac{\partial I}{\partial x} \frac{\partial I}{\partial y}$

Harris corner detector

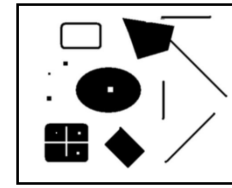
1) Compute *M* matrix for each image window to get their *cornerness* scores.
2) Find points whose surrounding window gave large corner response (*f*> threshold)
3) Take the points of local maxima, i.e., perform non-maximum suppression

C.Harris and M.Stephens. "A Combined Corner and Edge Detector." *Proceedings of the 4th Alvey Vision Conference*: pages 147—151, 1988.

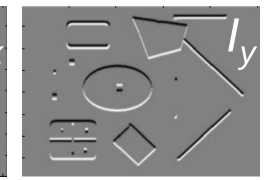
Harris Detector [Harris88]

- Second moment matrix

$$\mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$



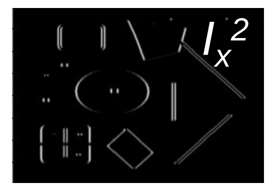
1. Image derivatives (optionally, blur first)

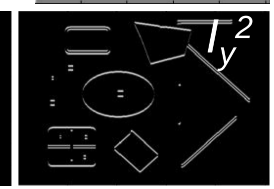


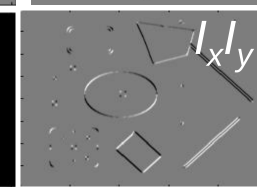
$$\det M = \lambda_1 \lambda_2$$

$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

2. Square of derivatives







3. Gaussian filter $g(\sigma_I)$

4. Cornerness function – both eigenvalues are strong

$$har = \det[\mu(\sigma_I, \sigma_D)] - \alpha[\operatorname{trace}(\mu(\sigma_I, \sigma_D))^2] =$$

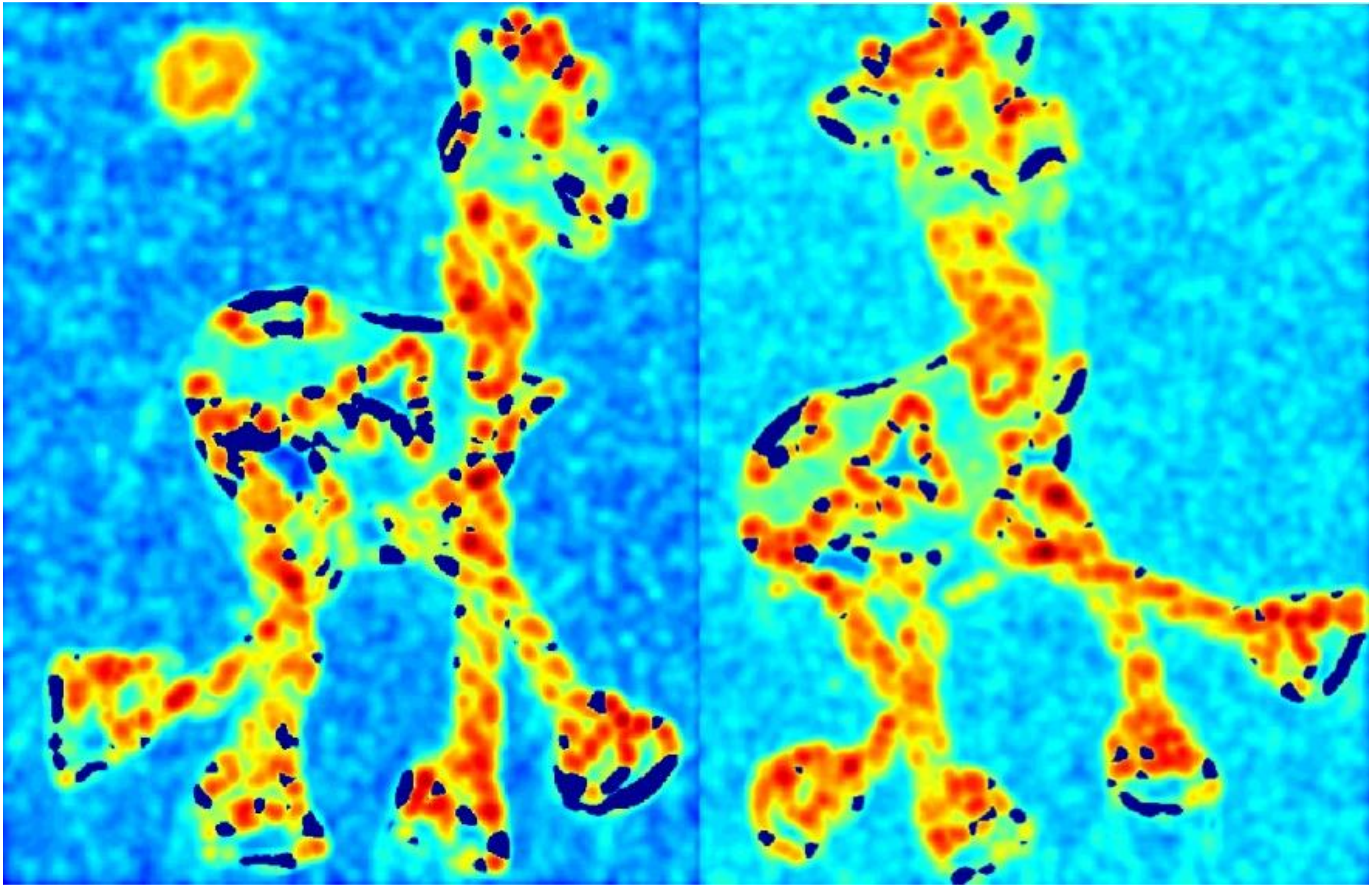
$$g(I_x^2)g(I_y^2) - [g(I_xI_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2$$

5. Non-maxima suppression

Harris Detector: Steps

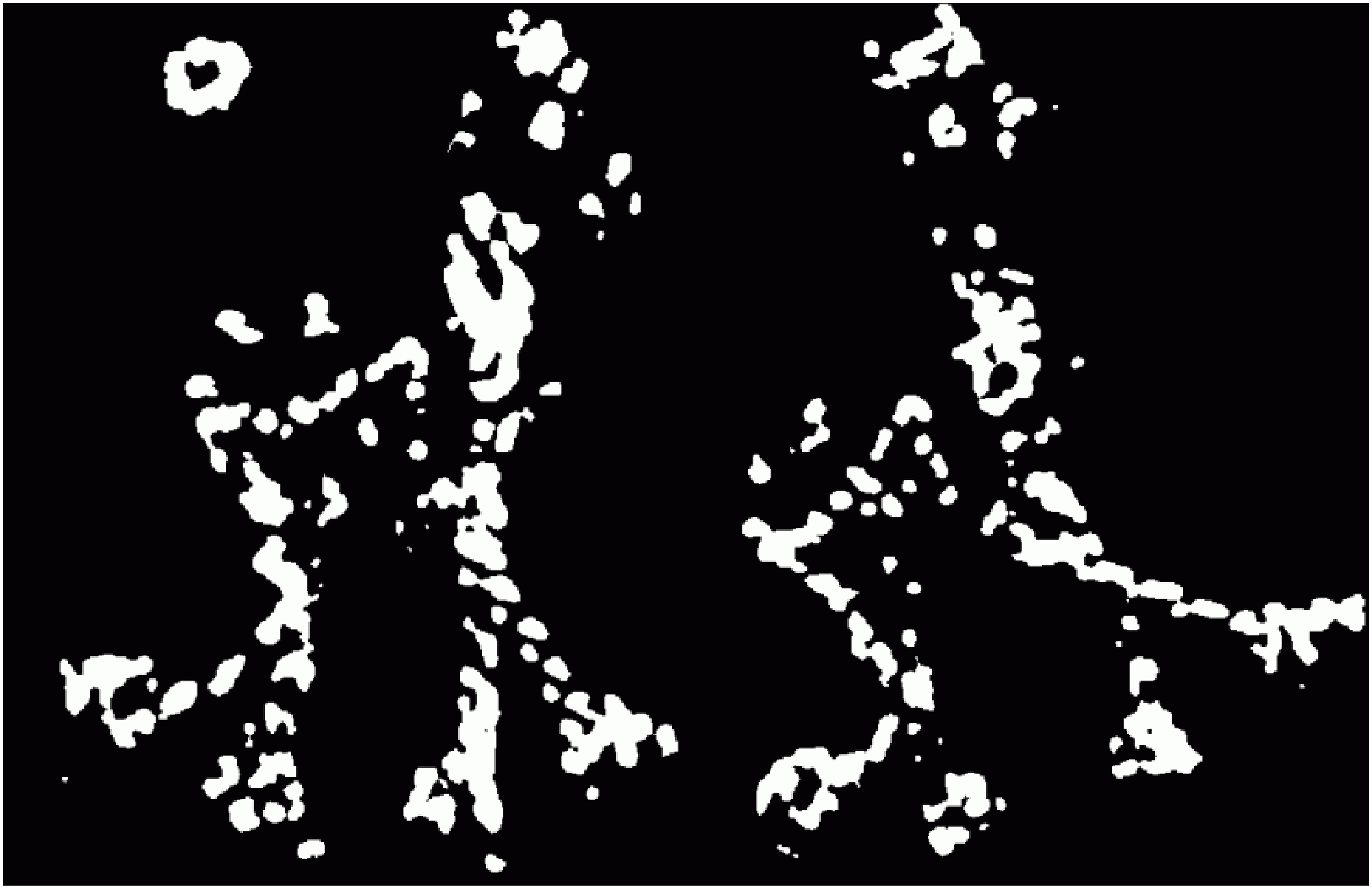
Harris Detector: Steps

Compute corner response R



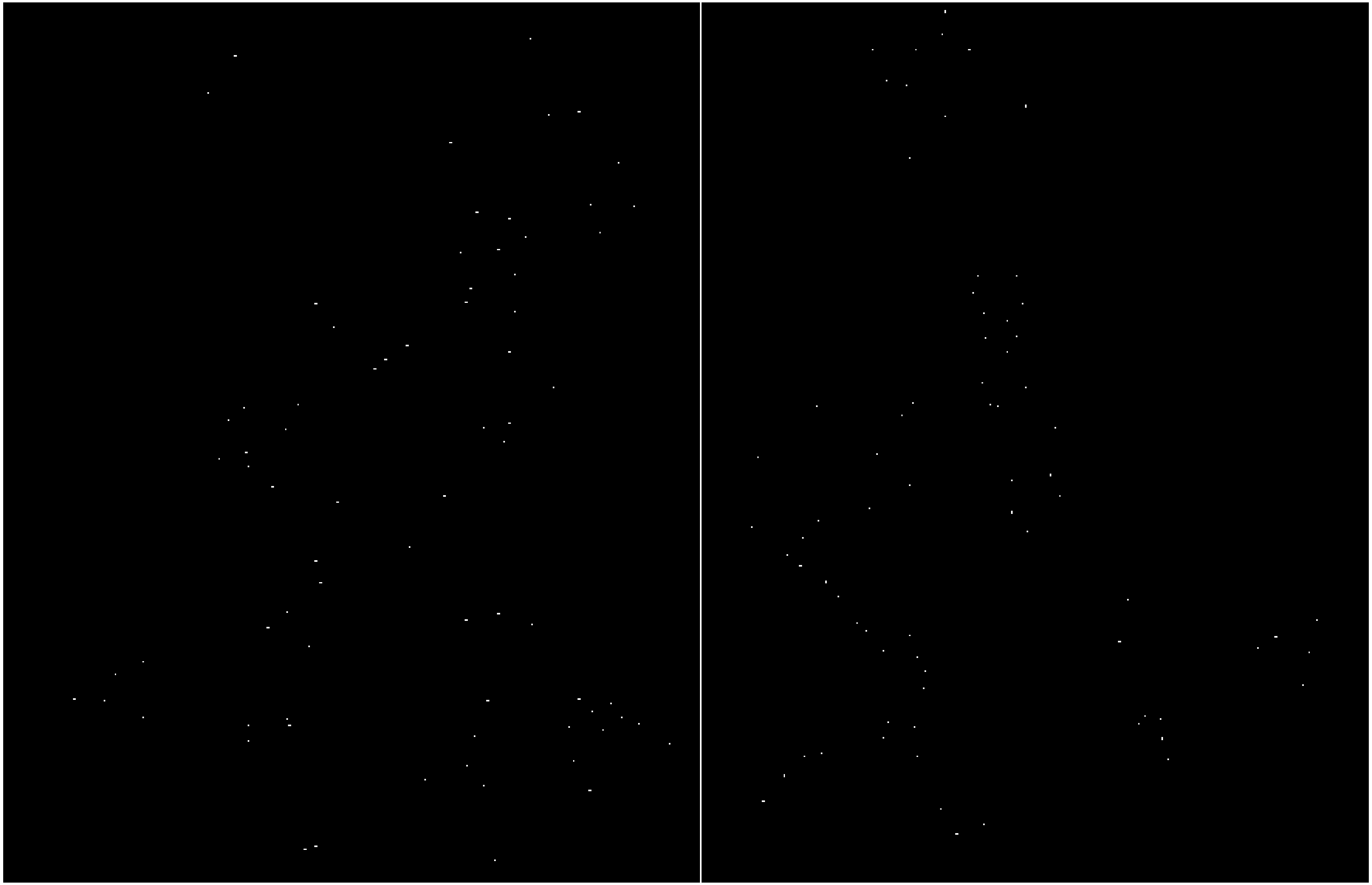
Harris Detector: Steps

Find points with large corner response: R>threshold



Harris Detector: Steps

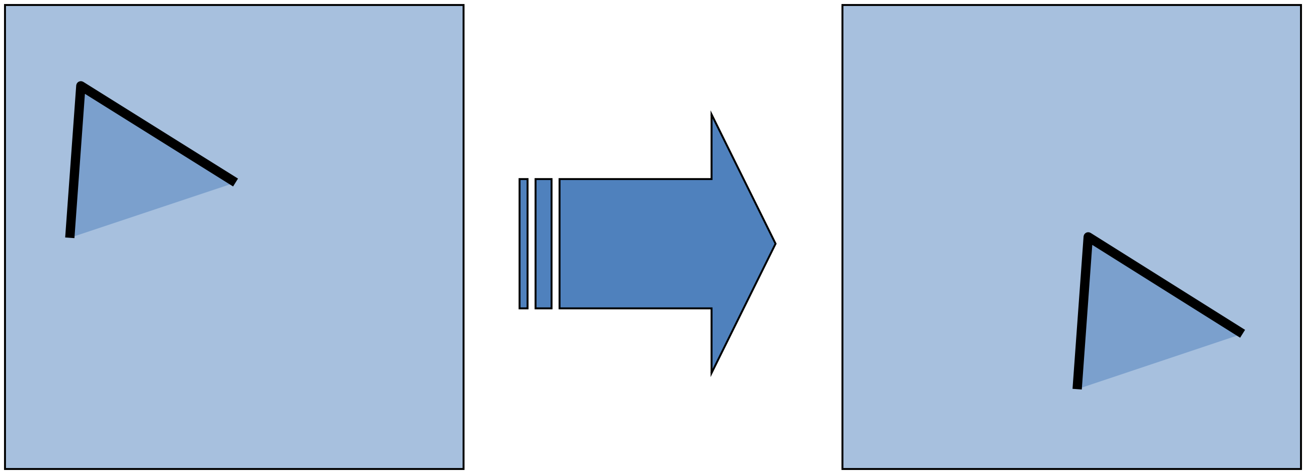
Take only the points of local maxima of R



Invariance and covariance

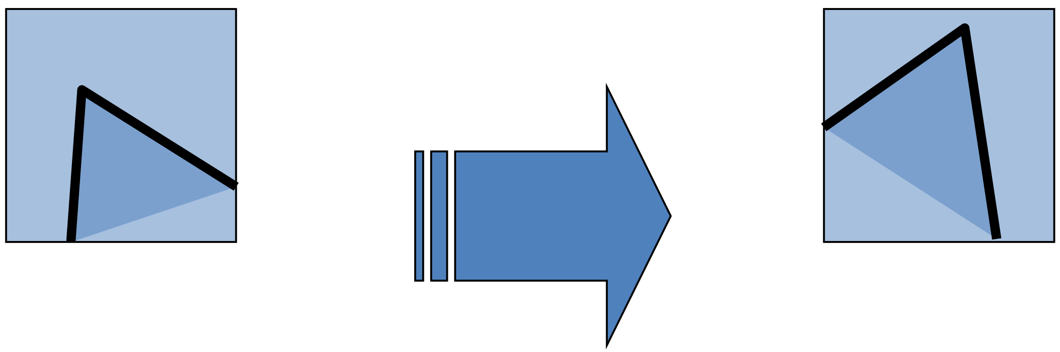
- We want corner locations to be ***invariant*** to photometric transformations and ***covariant*** to geometric transformations
  - **Invariance:** image is transformed and corner locations do not change
  - **Covariance:** if we have two transformed versions of the same image, features should be detected in corresponding locations

Image translation



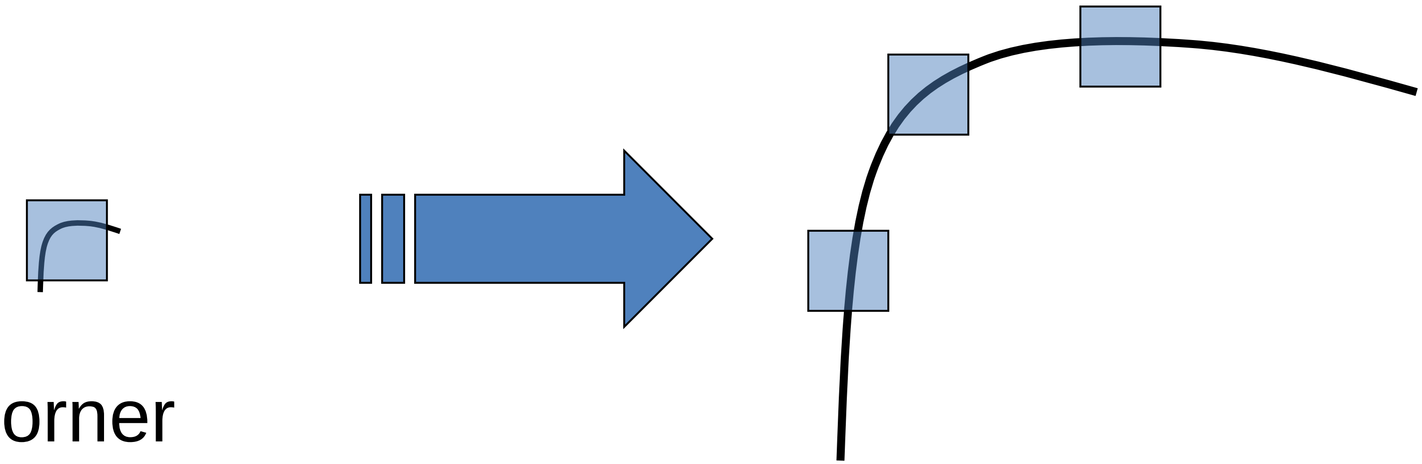
Corner location is covariant w.r.t. translation

Image rotation



Corner location is covariant w.r.t. rotation

Scaling

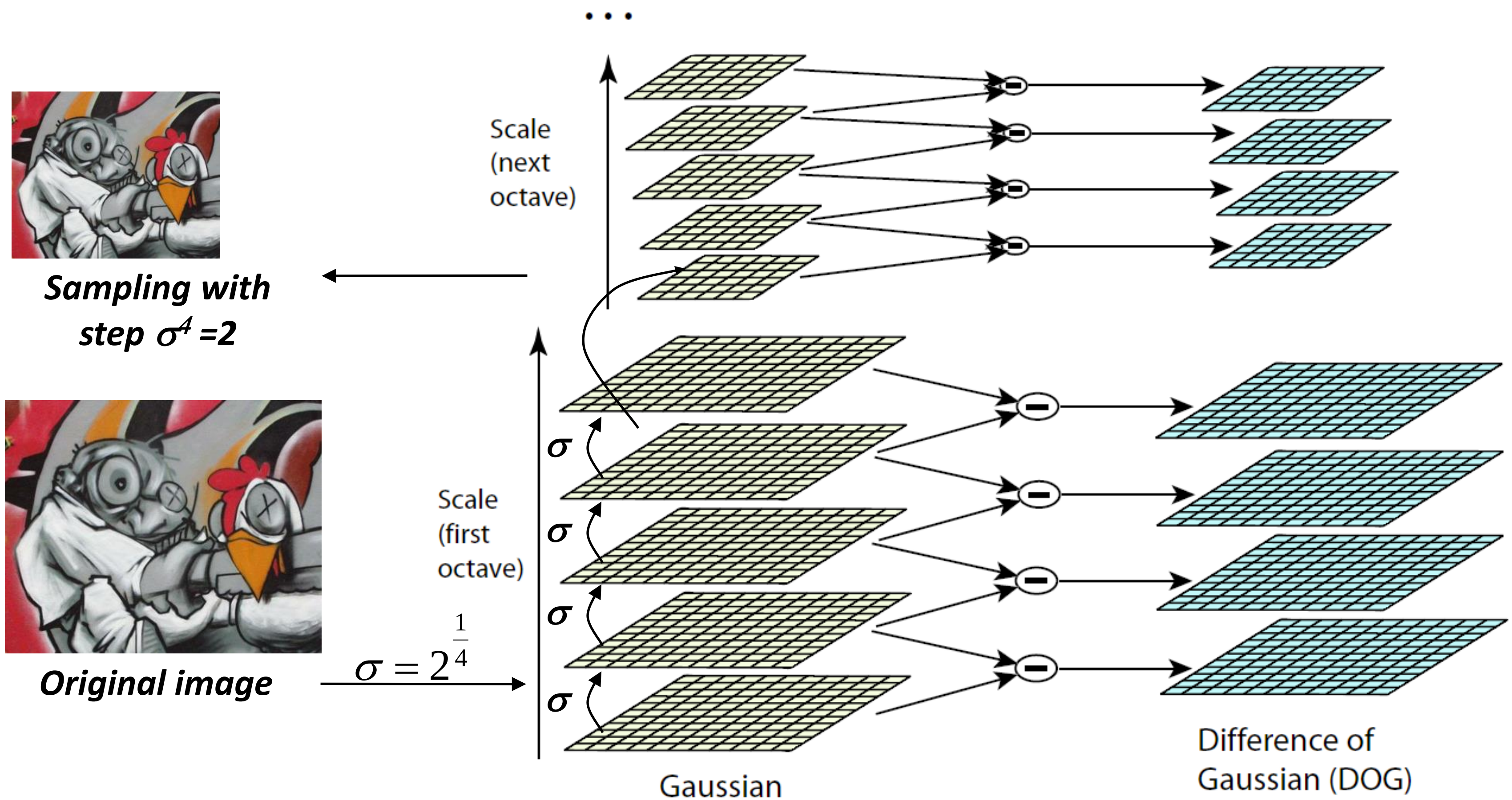


All points will be classified as edges

Corner location is not covariant w.r.t. Scaling

Difference-of-Gaussian (DoG)

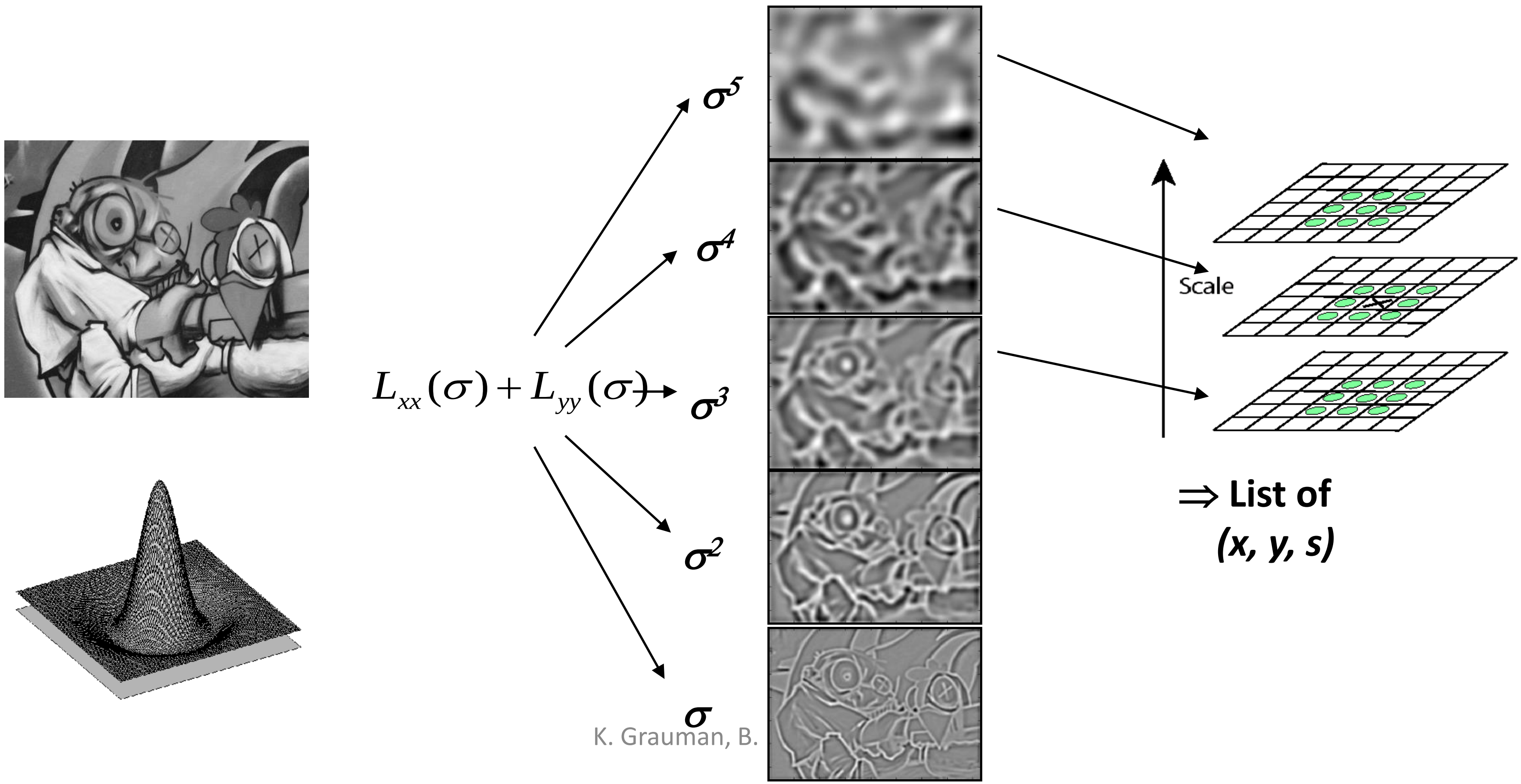
- Computation in **Gaussian scale pyramid**



K. Grauman, B. Leibe

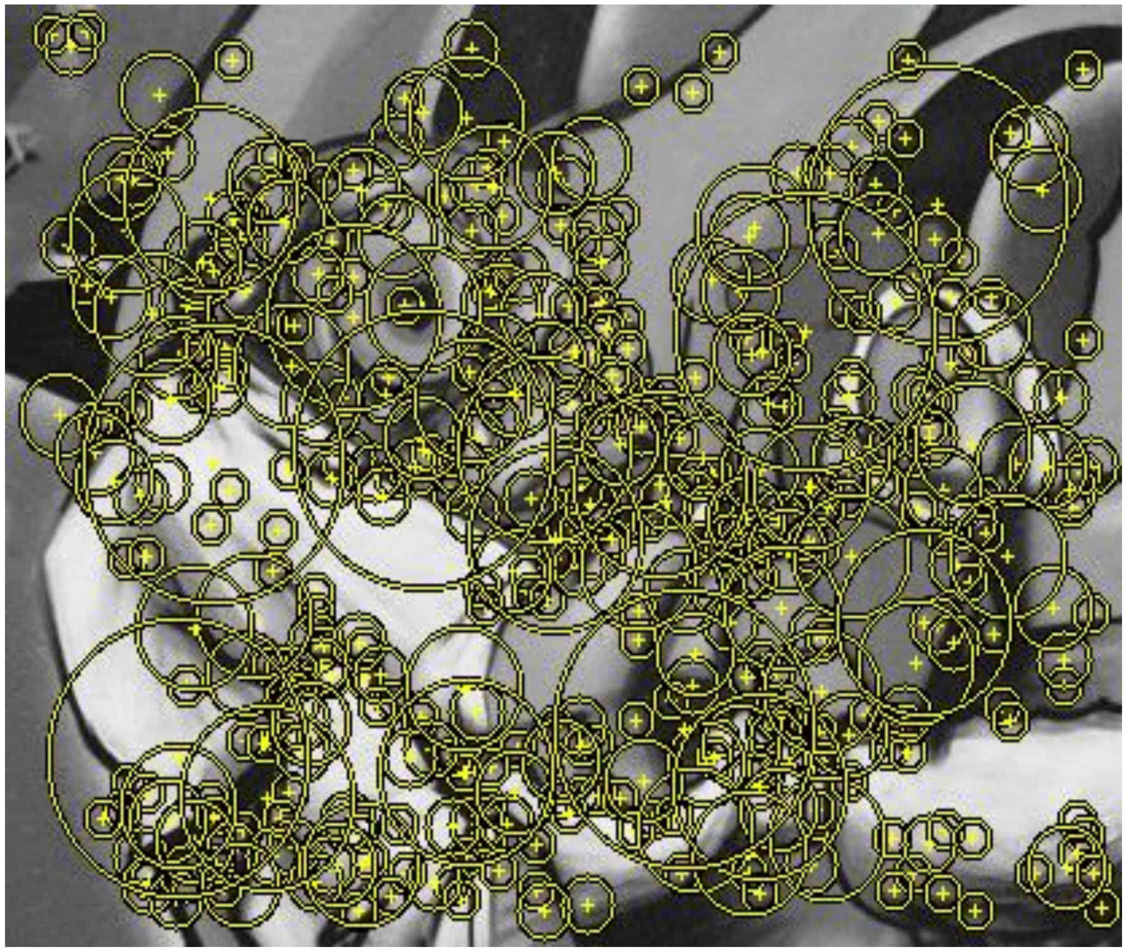
Find local maxima in position-scale space of Difference-of-Gaussian

Once this **DoG are found**, **images are searched for local extrema over scale and space**. For eg, one pixel in an image is compared with its 8 neighbours as well as 9 pixels in next scale and 9 pixels in previous scales. If it is a local extrema, it is a potential keypoint. It basically means that keypoint is best represented in that scale.



K. Grauman, B.

Results: Difference-of-Gaussian



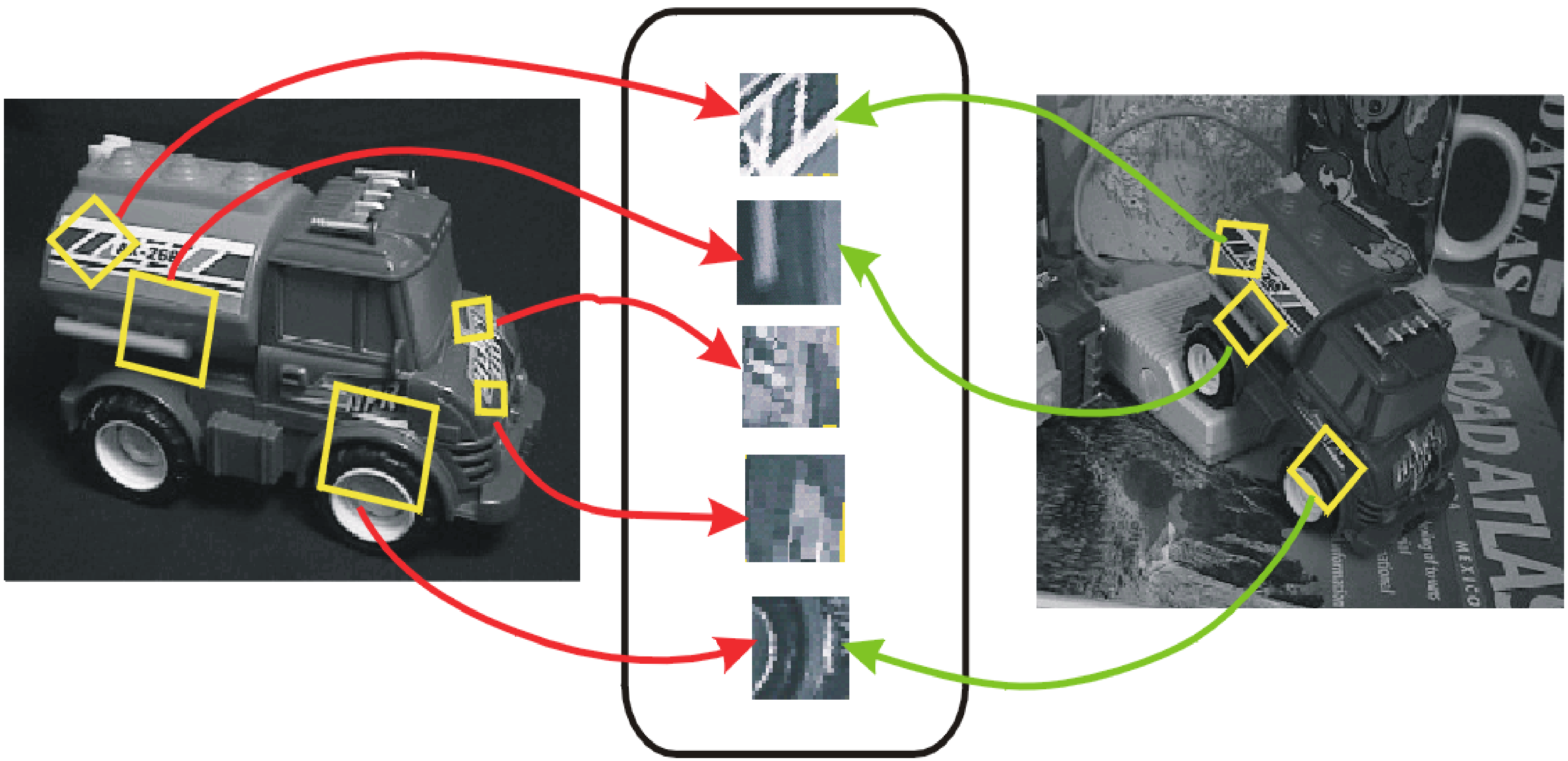
K. Grauman, B. Leibe

Local Descriptors

- Most features can be thought of as templates, histograms (counts), or combinations
- The ideal descriptor should be
  - Robust
  - Distinctive
  - Compact
  - Efficient
- Most available descriptors focus on edge/gradient information
  - Capture texture information
  - Color rarely used

Invariant Local Features

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters

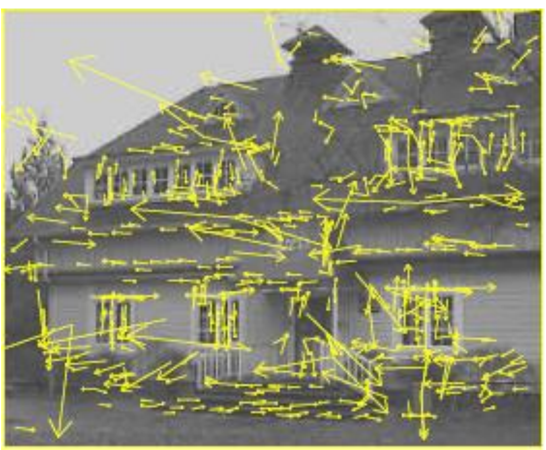


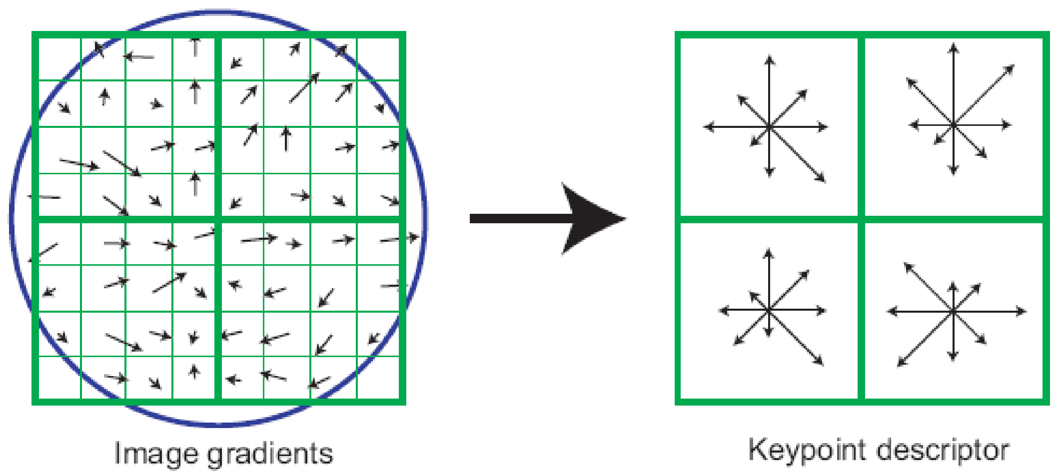
Features Descriptors

Scale-invariant feature transform (SIFT)

- **SIFT** is an algorithm in computer vision used to **detect and describe local features in images**.
- SIFT is based on Histograms of oriented gradients.

SIFT – Lowe IJCV 2004





- An **image gradient** is a directional change in the intensity or color in an **image**.

SIFT Main Steps

1. **Scale-space Extrema Detection:** we can't use the same window to detect keypoints with different scale. *To detect larger corners we need larger windows*. **For this, scale-space filtering is used**. In it, *Laplacian of Gaussian* is found for the image with various σ values.

2. **Keypoint Localization:** Once potential keypoints locations are found, they have to be **refined** to get more accurate results.

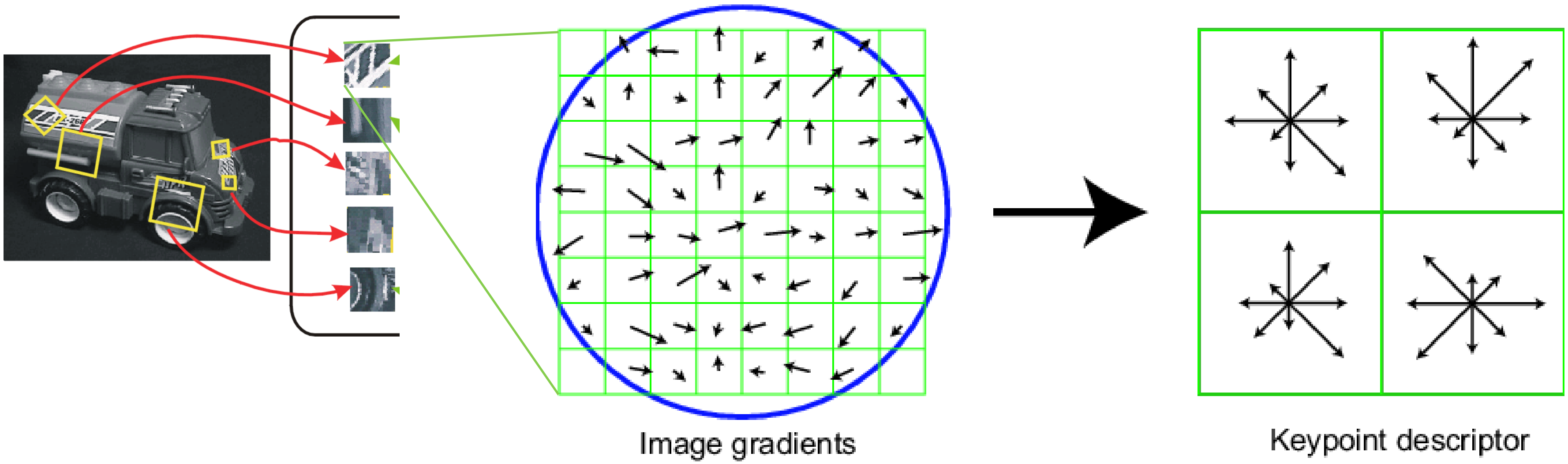
3. **Orientation Assignment:** Now an orientation is assigned to each keypoint to achieve invariance to image rotation. **A neigbourhood is taken around the keypoint location depending on the scale, and the gradient magnitude and direction is calculated in that region**. An orientation histogram with 36 bins covering 360 degrees is created.

SIFT Main Steps

4. **Keypoint Descriptor:** Now keypoint descriptor is created. **A 16x16 neighbourhood around the keypoint is taken**. **It is devided into 16 sub-blocks of 4x4 size**. For each sub-block, **8 bin orientation histogram** is created. Therefore, a total of **128 bin values are available**. It is represented as a vector to form keypoint descriptor.

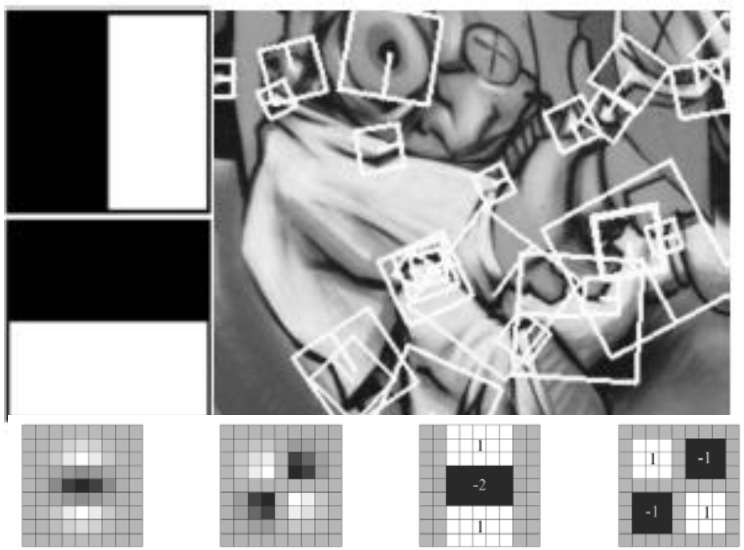
SIFT vector formation

- Computed on rotated and scaled version of window according to computed orientation & scale
- 4x4 array of gradient orientation histogram weighted by magnitude
- 8 orientations x 4x4 array = 128 dimensions
- Motivation: some sensitivity to spatial layout, but not too much.



showing only 2x2 here but is 4x4

Speeded-up robust features (SURF)



Fast approximation of SIFT idea

- Efficient computation by 2D box filters & integral images
  $\Rightarrow$ 6 times faster than SIFT
- Equivalent quality for object identification

GPU implementation available

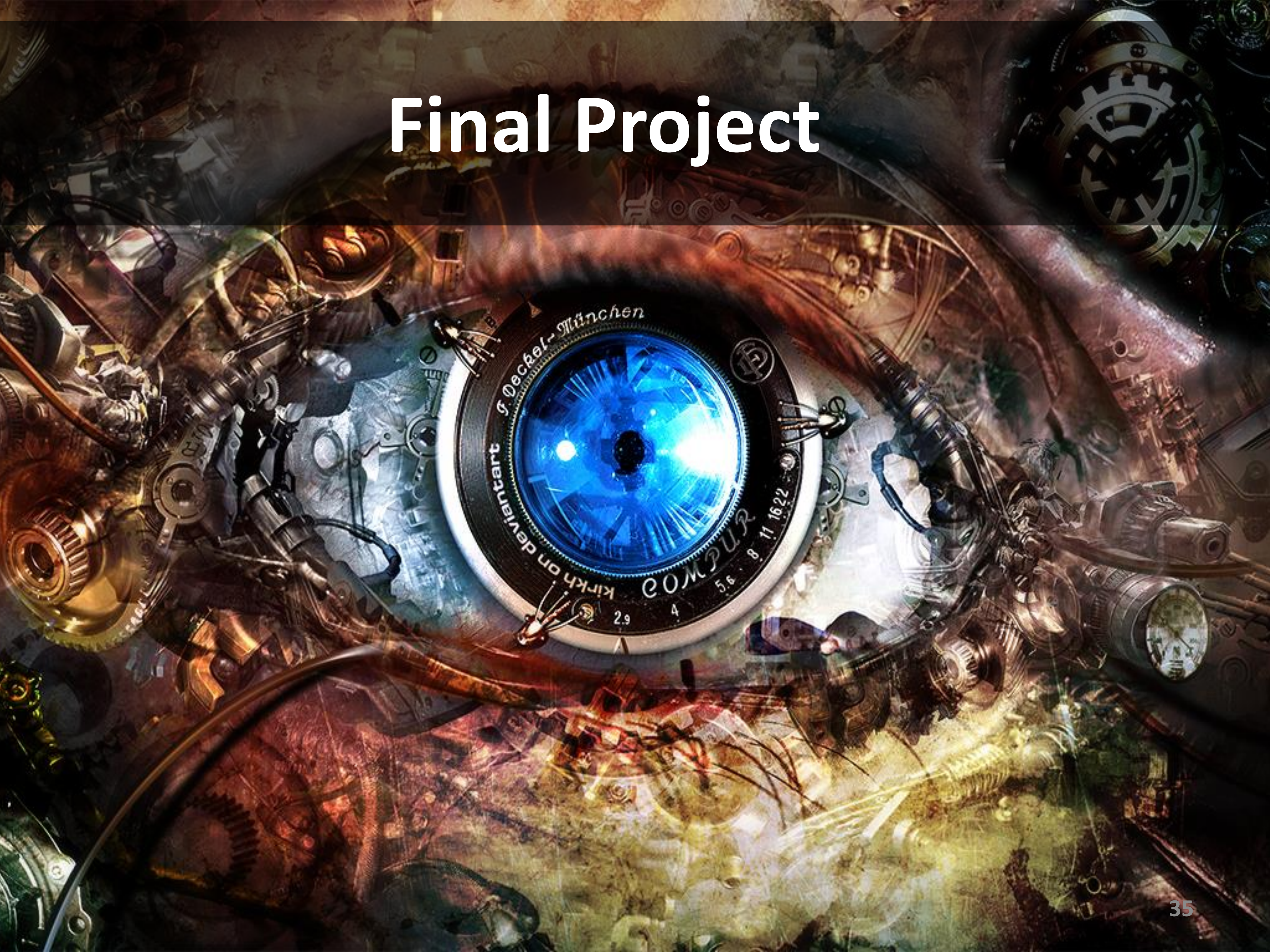
- Feature extraction @ 200Hz
  (detector + descriptor, 640×480 img)
- http://www.vision.ee.ethz.ch/~surf

[Bay, ECCV'06], [Cornelis, CVGPU'08]

Choosing a descriptor

- Again, need not stick to one
- For object instance recognition, SIFT or variant is a good choice

Final Project



Final Project

- **Team members from 3 (Min) to 5 (Max).**
- **Submit the names and short proposal (1 page) by 28 March (Deadline).**
- **The final project should have a short printed report at max 5 pages + Code.**
- **Prepare short presentation (5-10 slides) for max 10 minutes for all the team members.**
- **The short report should contains the following (Must):**
  1. ***Abstract***
  2. ***Short Introduction (Max 1 page)***
  3. ***Framework diagram***
  4. ***Problem statement***
  5. ***Objectives***
  6. ***The proposed System (Max 2 page)***
  7. ***Conclusion***
  8. ***References and citations along the text.***

Note: try to avoid copy and paste as much as you can in the report and code as well.

Final Project

- **Topics:**
  1. Location recognition
  2. Human Identification
  3. Face Detection and Recognition in Video
  4. Human tracking (in Video)
  5. OCR for Arabic/English
  6. Activity Recognition (in Video)
  7. Panoramas with image stitching
  8. Car Detection and Recognition.
  9. Car Tracking (in Video).
  10. Biometrics.
  11. Image Retrieval.
  12. Detection of possible cancer (medical imaging).
  13. Object Recognition (at least 5 different objects).
  14. Video Summarization.
  15. Sport Identification (at least 5 different sports).