## Photogrammetry & Robotics Lab

# Visual Features: Keypoints (Harris, Shi-Tomasi, Förstner, DoG)

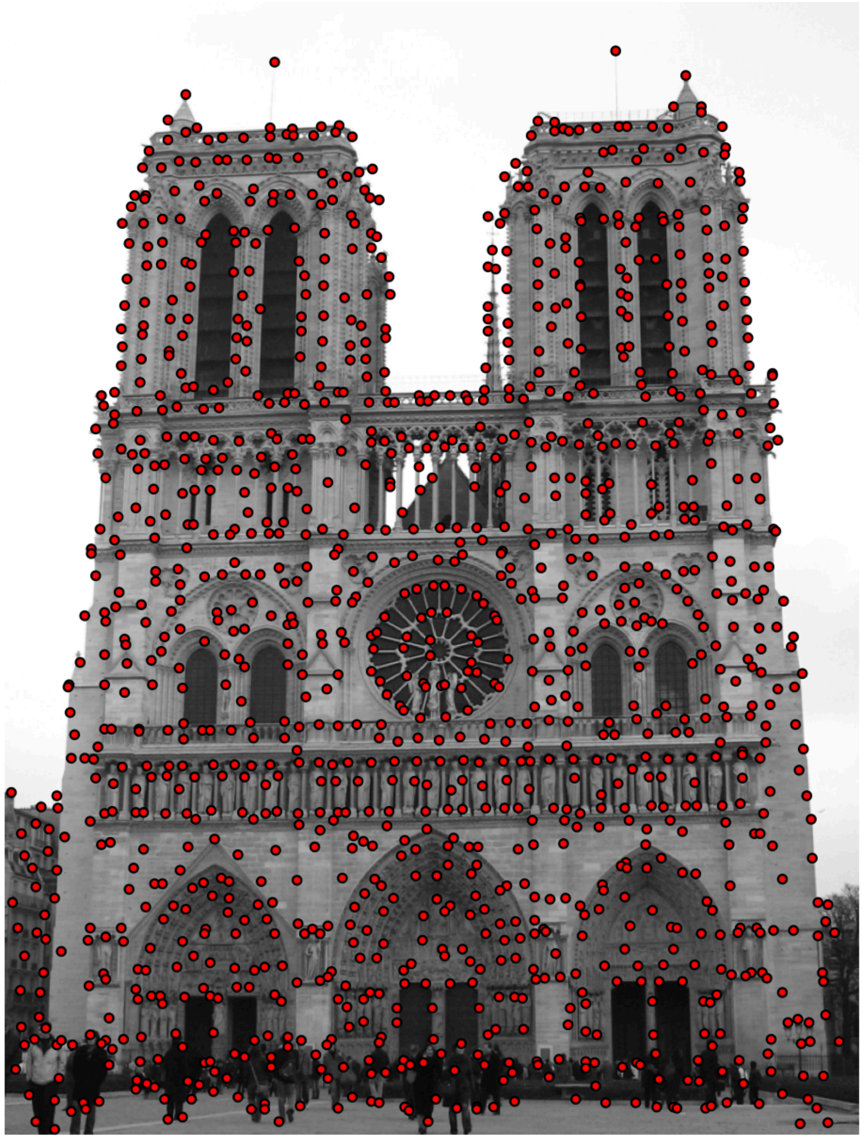
**Cyrill Stachniss**

Most slides have been created by Cyrill Stachniss but for several slides courtesy by Gil Levi, A. Efros, J. Hayes, D. Lowe and S. Savarese

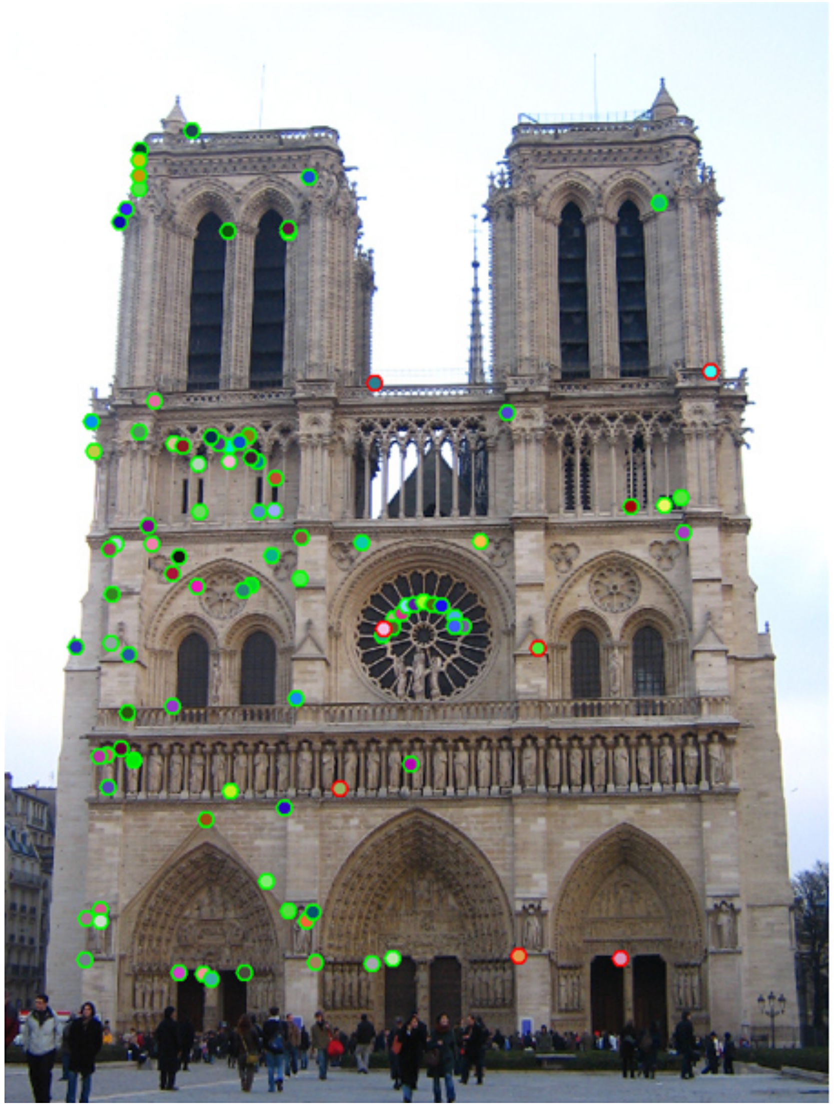
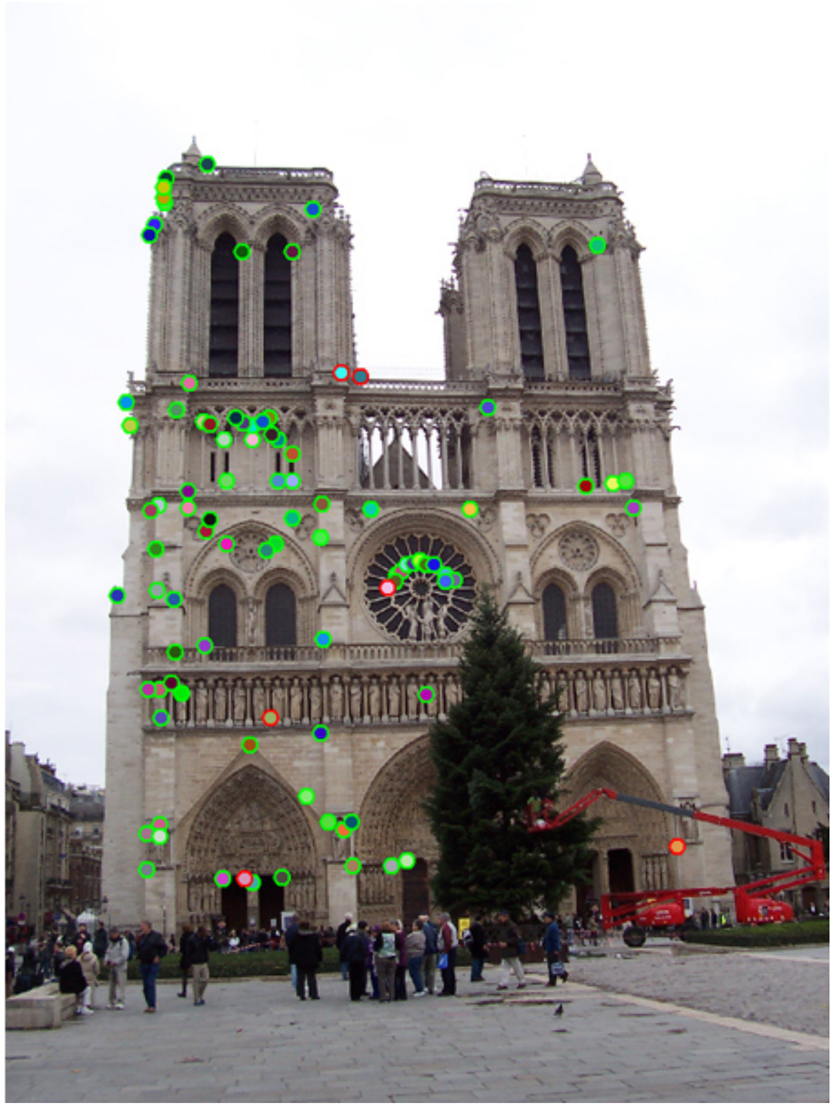
# 5 Minute Preparation for Today


https://www.ipb.uni-bonn.de/5min/

# Motivation



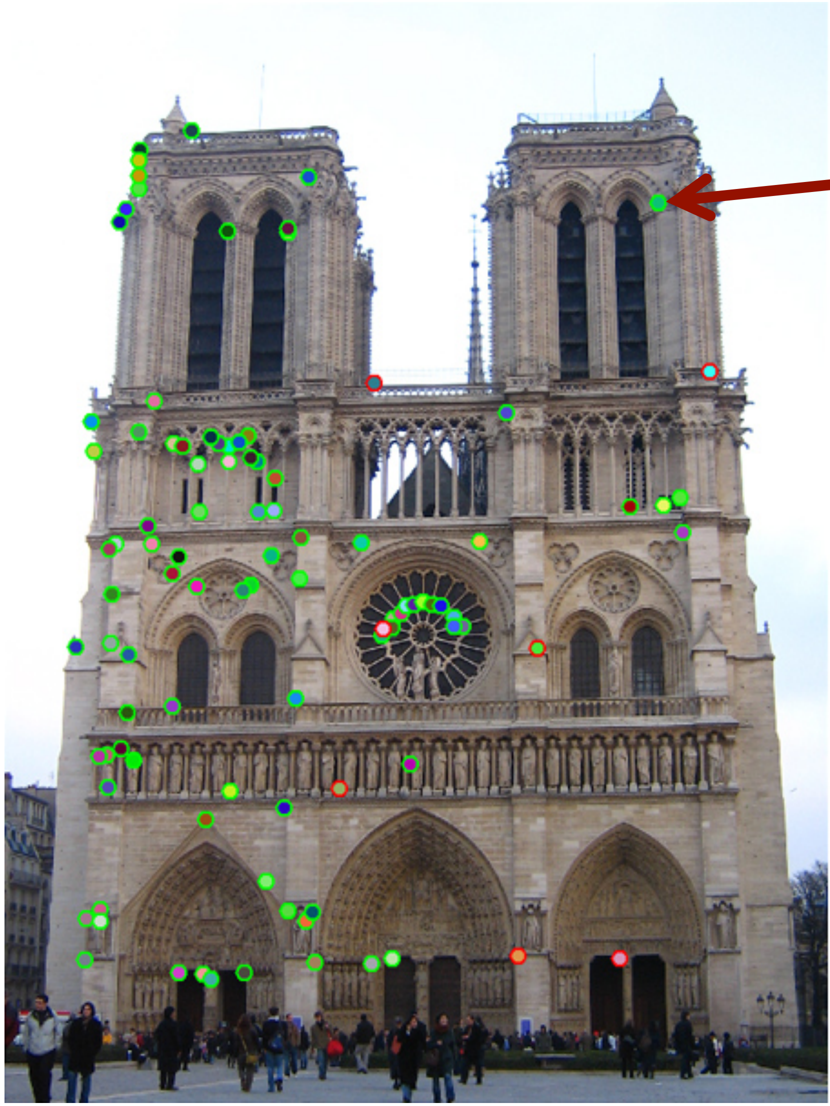
# Motivation

# Visual Features: Keypoints and Descriptors

- **Keypoint** is a (locally) distinct location in an image
- The feature **descriptor** summarizes the local structure around the keypoint

# Keypoint and Descriptor



**keypoint**

**descriptor** at the keypoint

$$f = \begin{bmatrix} 0.02 \\ 0.04 \\ 0.1 \\ 0.03 \\ 0 \\ \ldots \end{bmatrix}$$

# Today's Topics

- **Keypoints: Finding distinct points**
  - **Harris** corners
  - **Shi-Tomasi** corner detector
  - **Förstner** operator
  - **Difference of Gaussians**
- **Features: Describing a keypoint**
  - **SIFT** – Scale Invariant Feature Transform
  - **BRIEF** – Binary Robust Independent Elementary Features
  - **ORB** – Oriented FAST Rotated BRIEF

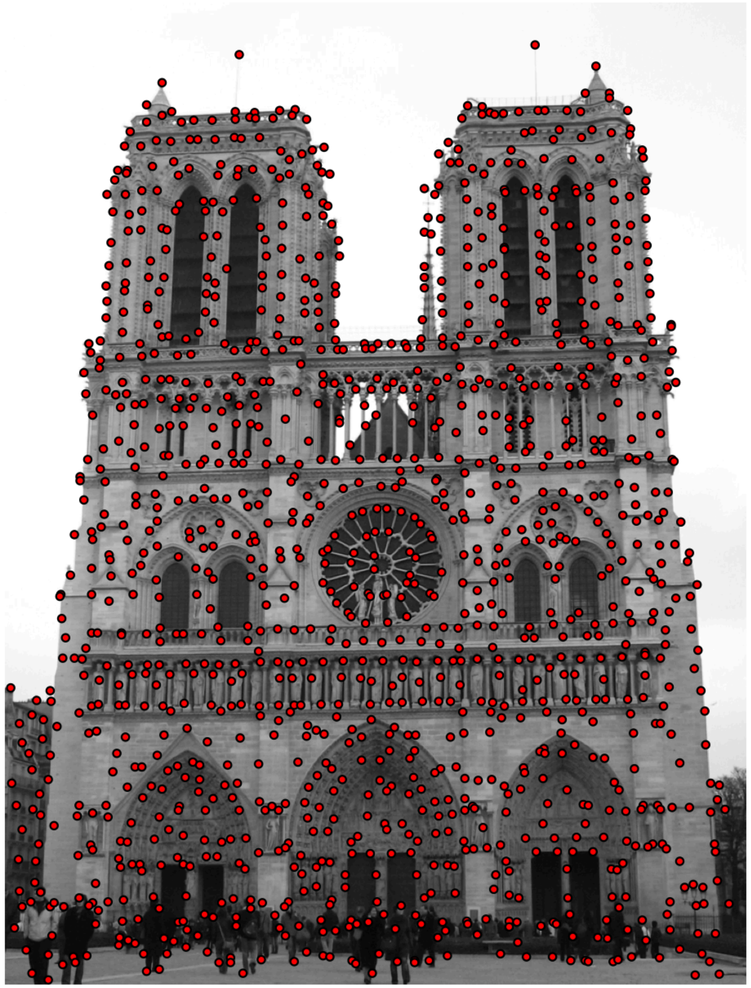
# Keypoints

"Finding locally distinct points"

Part 1: Corners

# Corners

- Corners are often highly distinct points



# Corners & Edges

- Corners are often highly distinct points
- Corners are invariant to translation, rotation, and illumination
- **Corner** = **two edges** in roughly orthogonal directions
- **Edge** = a sudden **brightness change**

# Finding Corners

- To find corners we need to **search for intensity changes** in two directions
- Compute the SSD of neighbor pixels around $(x, y)$

$$f(x,y) = \sum_{(u,v)\in W_{xy}} \left(I(u,v) - I(u+\delta u, v+\delta v)\right)^2$$

**local patch around (x,y)**

**sum of squared differences of image intensity values of pixels under a given shift (du, dv)**

# Finding Corners

- To find corners we need to **search for intensity changes** in two directions
- Compute the SSD of neighbor pixels around $(x, y)$

$$f(x, y) = \sum_{(u,v)\in W_{xy}} \left(I(u, v) - I(u + \delta u, v + \delta v)\right)^2$$

- Using Taylor expansion, we obtain

$$I(u + \delta u, v + \delta v) \approx I(u, v) + [J_x \ J_y] \begin{bmatrix} \delta u \\ \delta v \end{bmatrix}$$

Jacobian

# Finding Corners

- The Taylor approximation leads to

$$f(x,y) \approx \sum_{(u,v)\in W_{xy}} \left( [J_x \ J_y] \begin{bmatrix} \delta u \\ \delta v \end{bmatrix} \right)^2$$

- Written in matrix form as

$$f(x,y) \approx \sum_{(u,v)\in W_{xy}} \begin{bmatrix} \delta u \\ \delta v \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} J_x^2 & J_x J_y \\ J_x J_y & J_y^2 \end{bmatrix} \begin{bmatrix} \delta u \\ \delta v \end{bmatrix}$$

# Finding Corners

- Given

$$f(x,y) \approx \sum_{(u,v)\in W_{xy}} \begin{bmatrix} \delta u \\ \delta v \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} J_x^2 & J_x J_y \\ J_x J_y & J_y^2 \end{bmatrix} \begin{bmatrix} \delta u \\ \delta v \end{bmatrix}$$

- Move the sums inside the matrix

$$f(x,y) \approx \begin{bmatrix} \delta u \\ \delta v \end{bmatrix}^{\mathsf{T}} \underbrace{\begin{bmatrix} \sum_W J_x^2 & \sum_W J_x J_y \\ \sum_W J_y J_x & \sum_W J_y^2 \end{bmatrix}}_{\text{structure matrix}} \begin{bmatrix} \delta u \\ \delta v \end{bmatrix}$$

**structure matrix**

# Structure Matrix

- The structure matrix is key to finding edges and corners
- It encodes the changes in image intensities in a local area

$$M = \begin{bmatrix} \sum_W J_x^2 & \sum_W J_x J_y \\ \sum_W J_y J_x & \sum_W J_y^2 \end{bmatrix}$$

- Built from the image gradients

# Computing the Structure Matrix

- Matrix build from the image gradients

$$M = \begin{bmatrix} \sum_W J_x^2 & \sum_W J_x J_y \\ \sum_W J_y J_x & \sum_W J_y^2 \end{bmatrix}$$

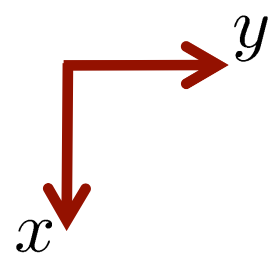
- Jacobians computed via a convolution with a gradient kernel such as Scharr or Sobel:

$$\begin{aligned} J_x^2 &= (D_x * I)^2 \\ J_x J_y &= (D_x * I)(D_y * I) \\ J_y^2 &= (D_y * I)^2 \end{aligned}$$

# Computing the Structure Matrix

- Matrix build from the image gradients

$$M = \begin{bmatrix} \sum_W J_x^2 & \sum_W J_x J_y \\ \sum_W J_y J_x & \sum_W J_y^2 \end{bmatrix}$$




- Jacobians via Scharr or Sobel Op:

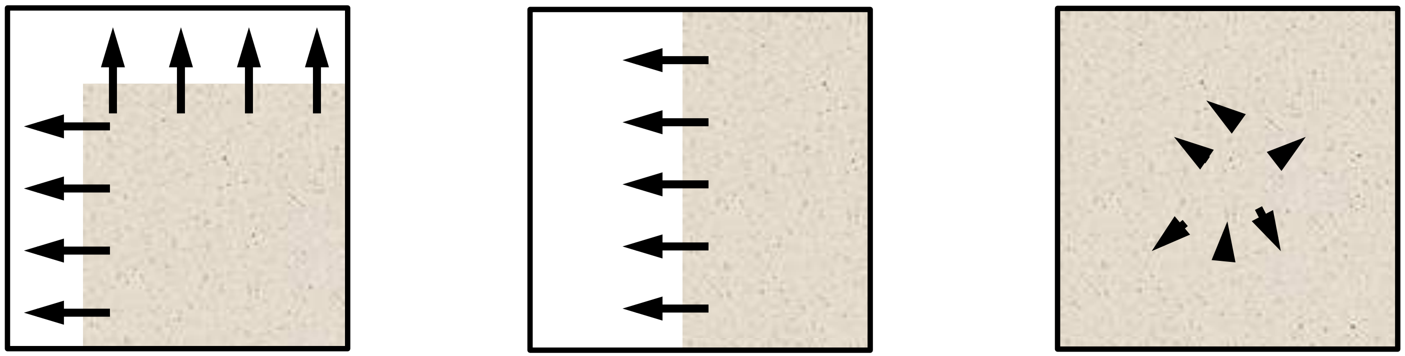
$$D_x^{\text{Scharr}} = \frac{1}{32}\begin{bmatrix} 3 & 10 & 3 \\ 0 & \underline{0} & 0 \\ -3 & -10 & -3 \end{bmatrix} \qquad D_x^{\text{Sobel}} = \frac{1}{8}\begin{bmatrix} 1 & 2 & 1 \\ 0 & \underline{0} & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$D_y^{\text{Scharr}} = \frac{1}{32}\begin{bmatrix} 3 & 0 & -3 \\ 10 & \underline{0} & -10 \\ 3 & 0 & -3 \end{bmatrix} \qquad D_y^{\text{Sobel}} = \frac{1}{8}\begin{bmatrix} 1 & 0 & -1 \\ 2 & \underline{0} & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

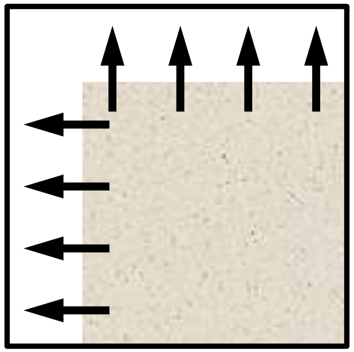
# Structure Matrix

- Summarizes the dominant directions of the gradient around a point

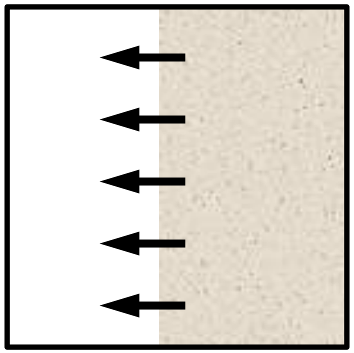
$$M = \begin{bmatrix} \sum_W J_x^2 & \sum_W J_x J_y \\ \sum_W J_y J_x & \sum_W J_y^2 \end{bmatrix}$$



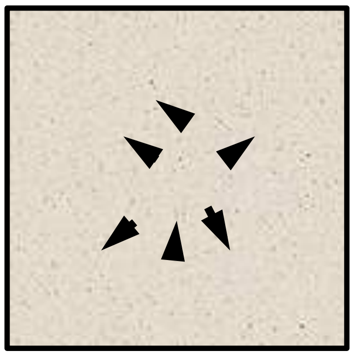
# Structure Matrix Examples



$$M = \begin{bmatrix} \gg 1 & \sim 0 \\ \sim 0 & \gg 1 \end{bmatrix}$$

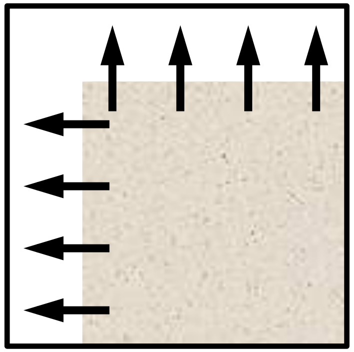


$$M = \begin{bmatrix} \sim 0 & \sim 0 \\ \sim 0 & \gg 1 \end{bmatrix}$$

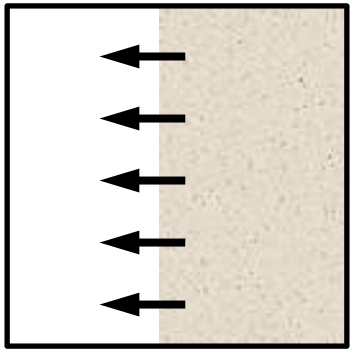


$$M = \begin{bmatrix} \sim 0 & \sim 0 \\ \sim 0 & \sim 0 \end{bmatrix}$$

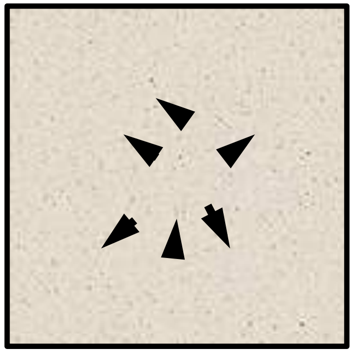
# Structure Matrix Examples



$$M = \begin{bmatrix} \gg 1 & \sim 0 \\ \sim 0 & \gg 1 \end{bmatrix}$$ **YES!**

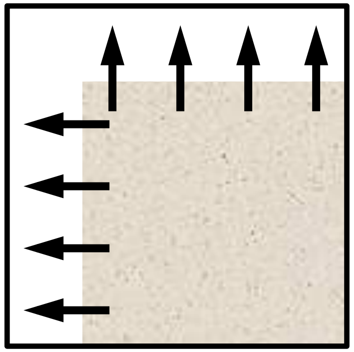


$$M = \begin{bmatrix} \sim 0 & \sim 0 \\ \sim 0 & \gg 1 \end{bmatrix}$$ **NO!**



$$M = \begin{bmatrix} \sim 0 & \sim 0 \\ \sim 0 & \sim 0 \end{bmatrix}$$ **NO!**

# Corners from Structure Matrix



$$\Rightarrow M = \begin{bmatrix} \gg 1 & \sim 0 \\ \sim 0 & \gg 1 \end{bmatrix}$$ **YES!**

**Key idea:**

**Considers points as corners if their structure matrix has two large Eigenvalues**

# Harris, Shi-Tomasi & Förstner

- Three similar approaches
- Proposed in
  - 1987 (Förstner)
  - 1988 (Harris)
  - 1994 (Shi-Tomasi)
- All rely on the structure matrix
- Use different criterion for deciding of a point is a corner or not
- Förstner offers subpixel estimation

# Harris Corner Criterion

- Criterion

$$R = \det(M) - k\,(\text{trace}(M))^2$$
$$= \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$$

$k \in [0.04, 0.06]$

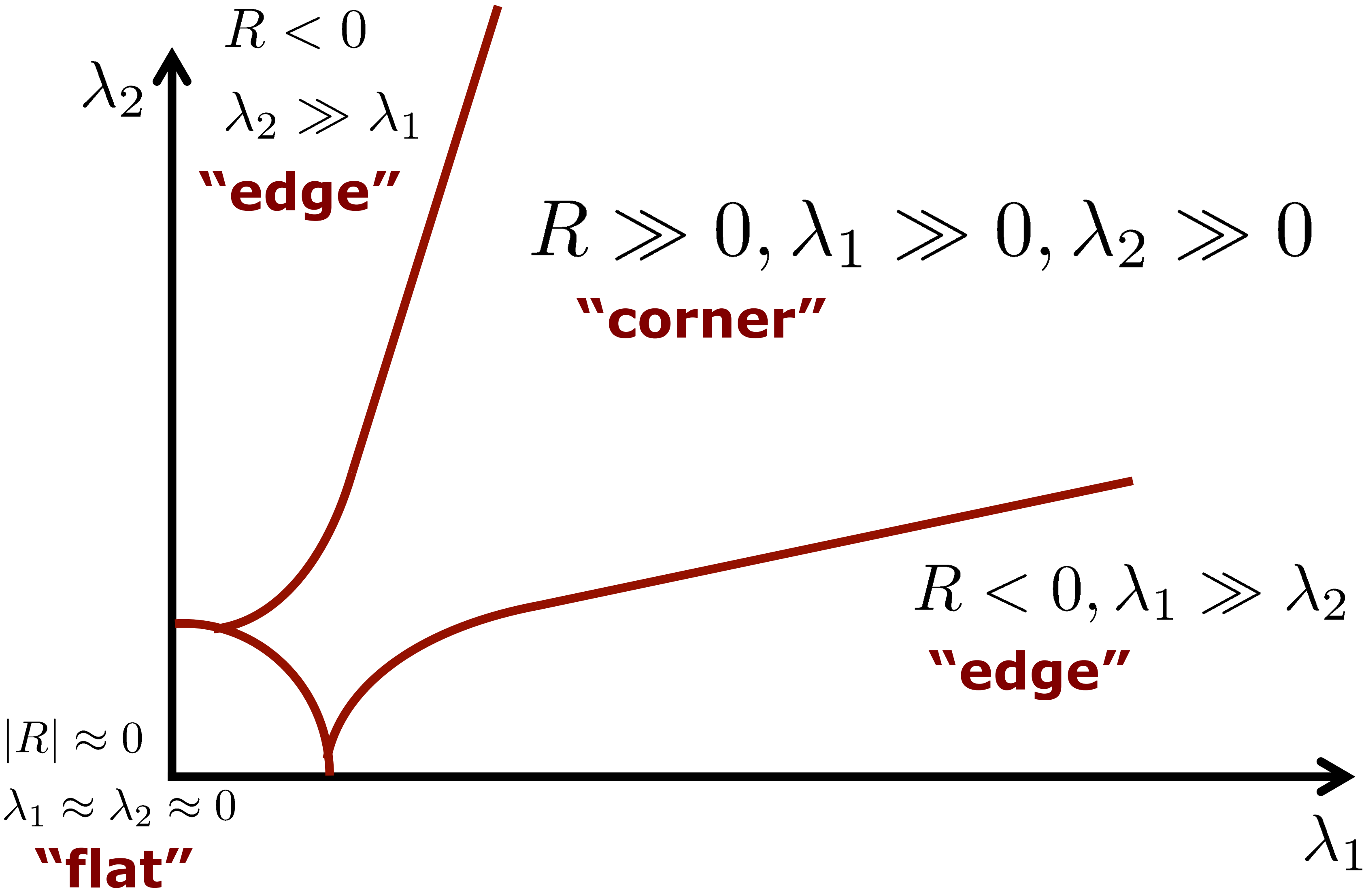
- with

$$|R| \approx 0 \Rightarrow \lambda_1 \approx \lambda_2 \approx 0$$ **: flat region**

$$R < 0 \Rightarrow \lambda_1 \gg \lambda_2 \text{ or } \lambda_2 \gg \lambda_1$$ **: edge**

$$R \gg 0 \Rightarrow \lambda_1 \approx \lambda_2 \gg 0$$ **: corner**

# Harris Criterion Illustrated



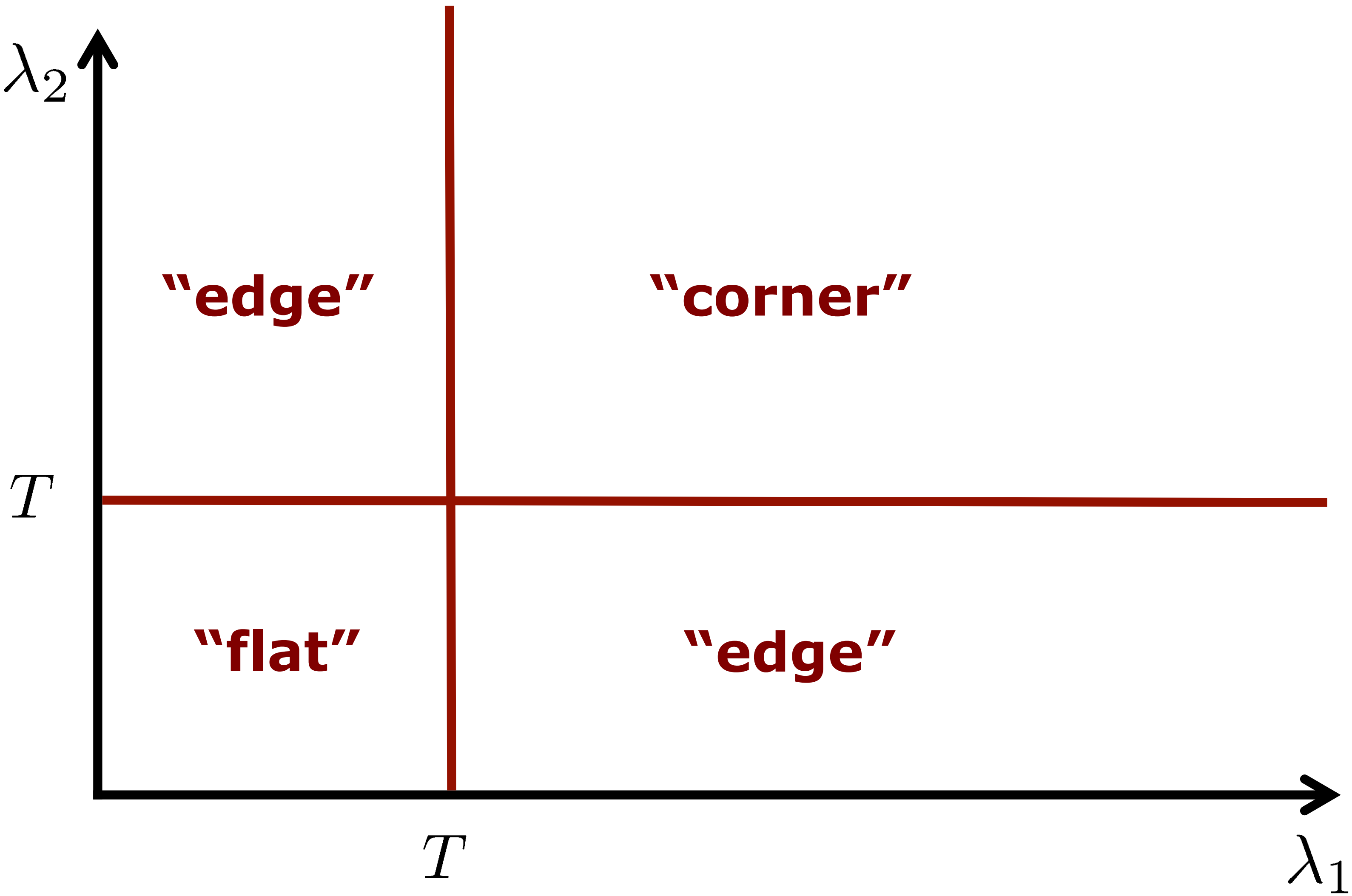
# Shi-Tomasi Corner Detector

- Criterion: Threshold smallest Eigenvalue

$$\lambda_{\min}(M) = \frac{\mathrm{trace}(M)}{2} - \frac{1}{2}\sqrt{(\mathrm{trace}(M))^2 - 4\det(M)}$$

$\lambda_{\min}(M) \geq T$ **: corner**

# Shi-Tomasi Criterion Illustrated



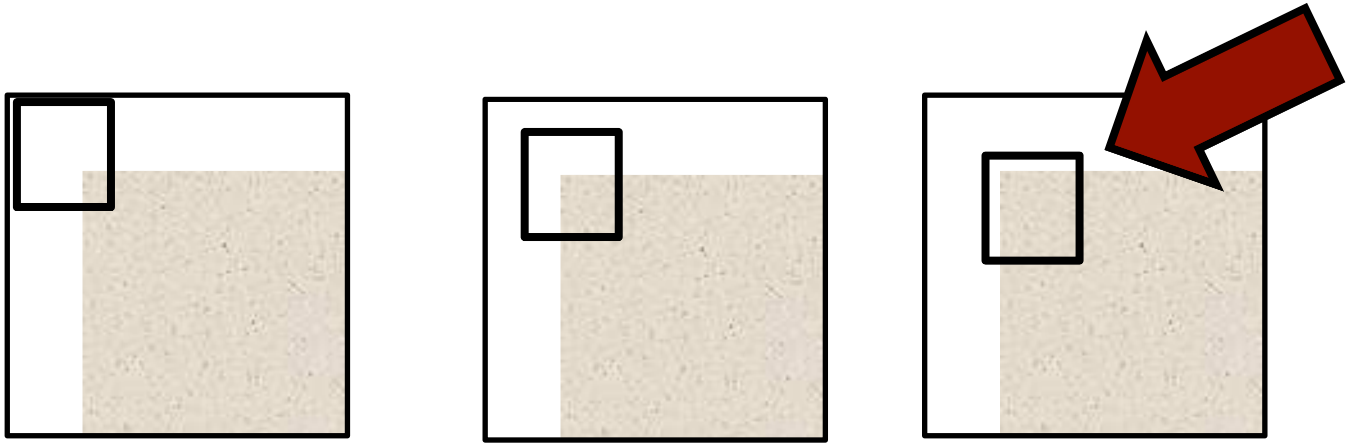
# Förstner Operator Criterion

- Very similar to Harris corner detector
- Defined on the inverse of the M (covariance matrix of possible shifts)
- Similar criterion on size and roundness of the error ellipse of covariance matrix
- Extension for sub-pixel estimation

# Non-Maxima Supression

- Within a local region, looks for the position with the maximum value ( $R$ or $\lambda_{\min}$ ) and select this point

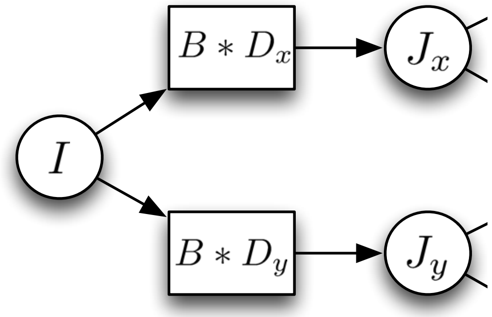
- Example for the Förstner operator



# Implementation Remarks

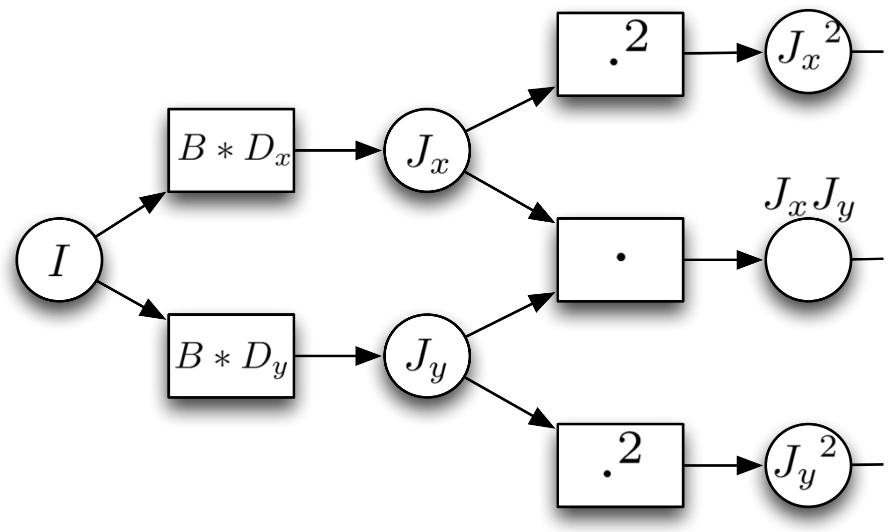
- RGB to gray-scale conversion first
- Real images are affected by noise, smoothing of the input is suggested

# Summary Corner Detection




convolutions
(smoothing
& derivatives)

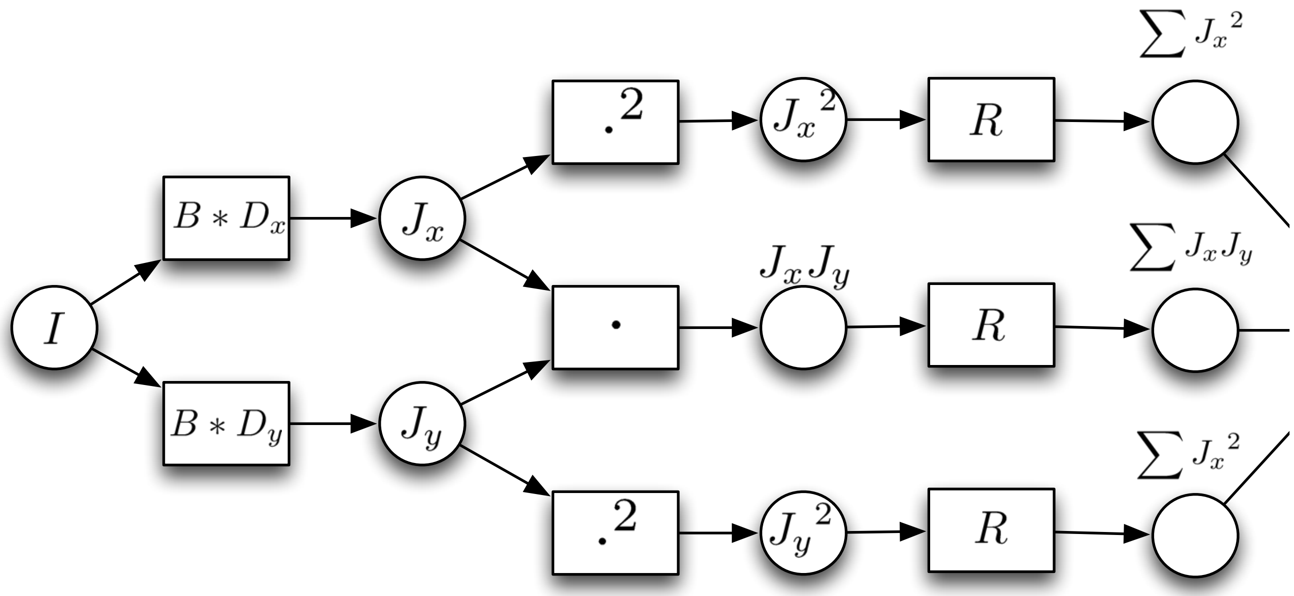
# Summary Corner Detection




convolutions
(smoothing
& derivatives)

multiplications

# Summary Corner Detection

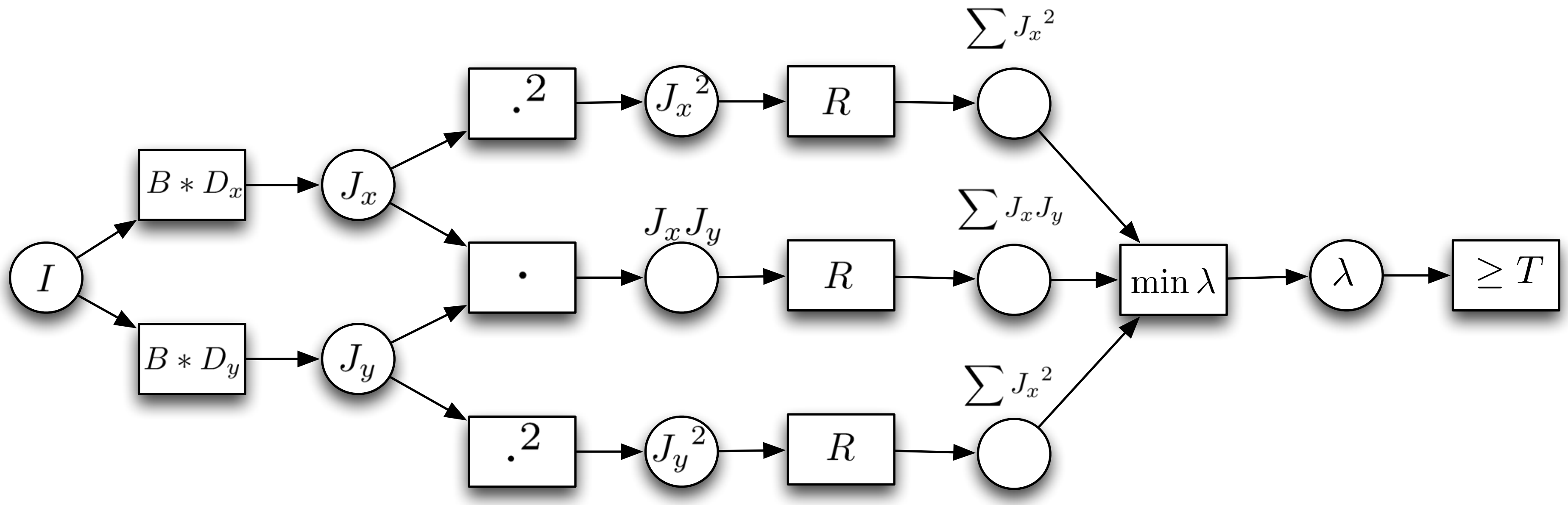



convolutions
(smoothing
& derivatives)

multiplications

convolutions
(box-summing)

# Summary Corner Detection




convolutions
(smoothing
& derivatives)

multiplications

convolutions
(box-summing)

multiplications,
sums, sqrt

thresholding
non-max suppression

# Example

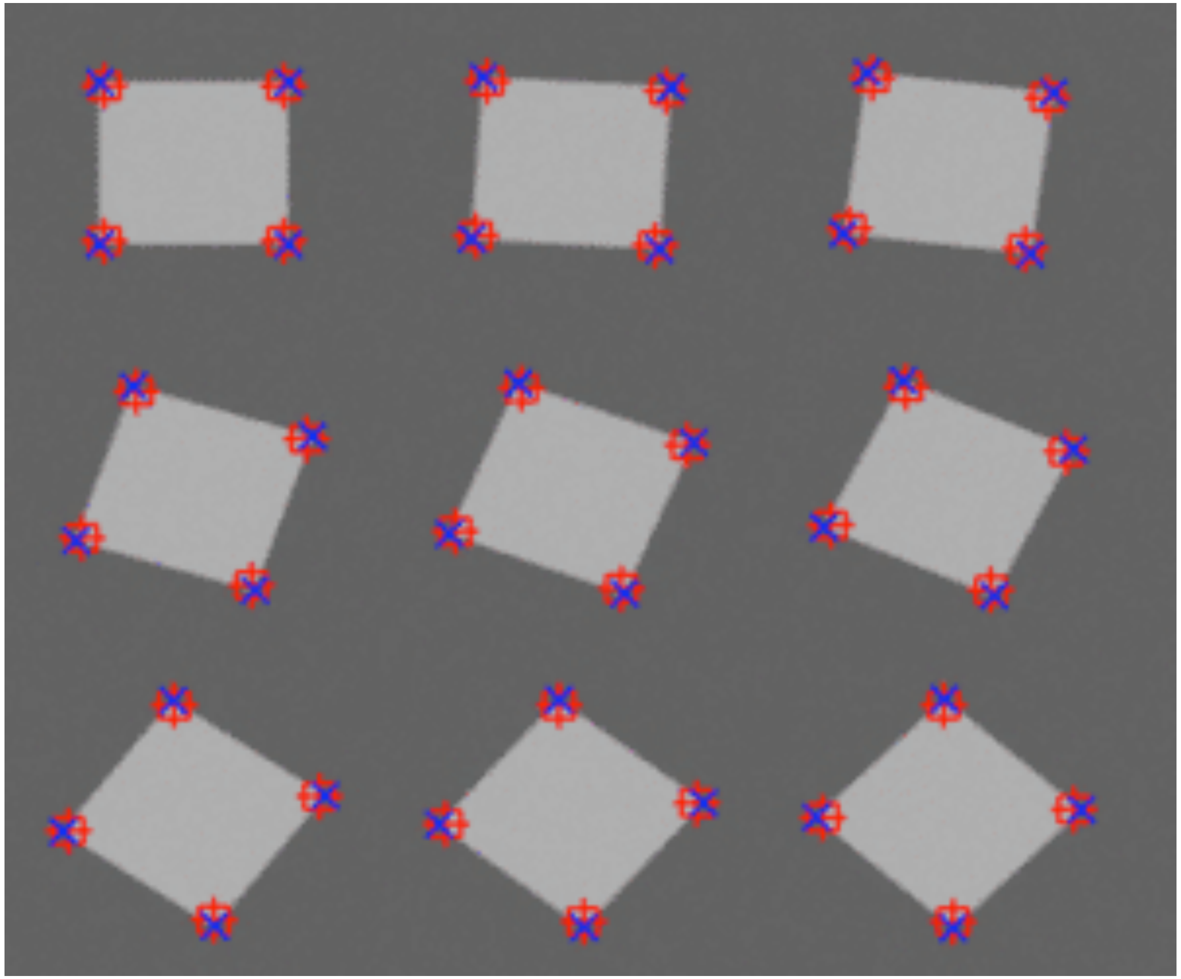


Image courtesy: Förstner

# Harris Corners Example

# Corner Detectors Comparison

- All three detectors perform similarly
- Förstner was the first one and additionally described subpixel estim.
- Harris became the most famous corner detector in the past
- Shi-Tomasi seems to slightly outperform Harris corners
- Most libraries use Shi-Tomasi as the default corner detector (e.g., openCV)

# Keypoints

"Finding locally distinct points"

Part 2: Difference of Gaussians

# Difference of Gaussians Keypoints

- A variant of corner detection
- Provides responses at corners, edges, and blobs
- Blob = mainly constant region but different to its surroundings

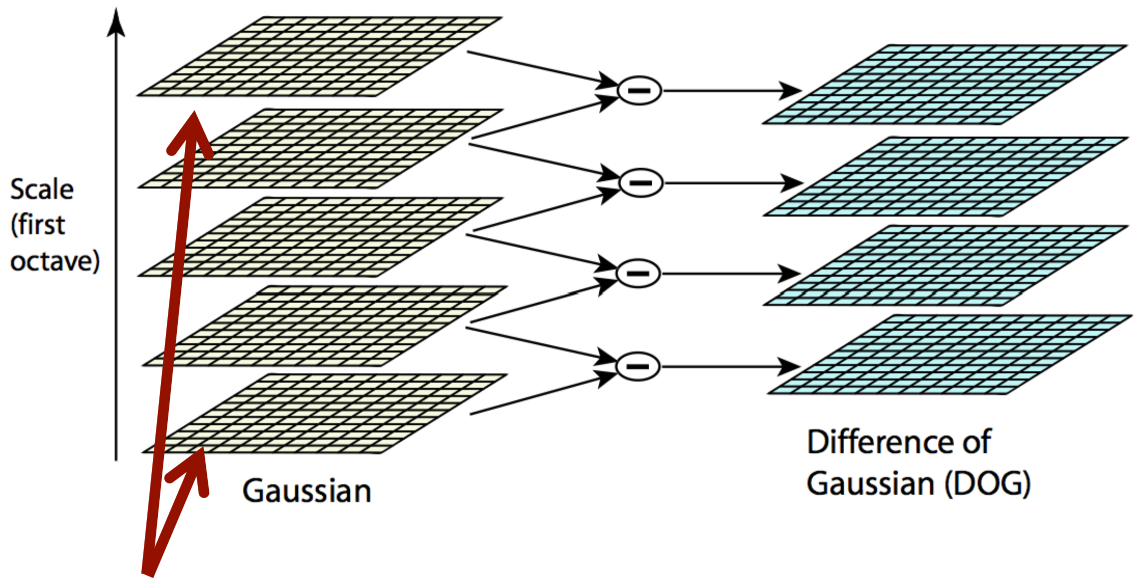
# Keypoints: Difference of Gaussians Over Scale-Space Pyramid

**Procedure**

Over different image pyramid levels

- Step 1: Gaussian smoothing
- Step 2: Difference-of-Gaussians: find extrema (over smoothing scales)
- Step 3: maxima suppression at edges

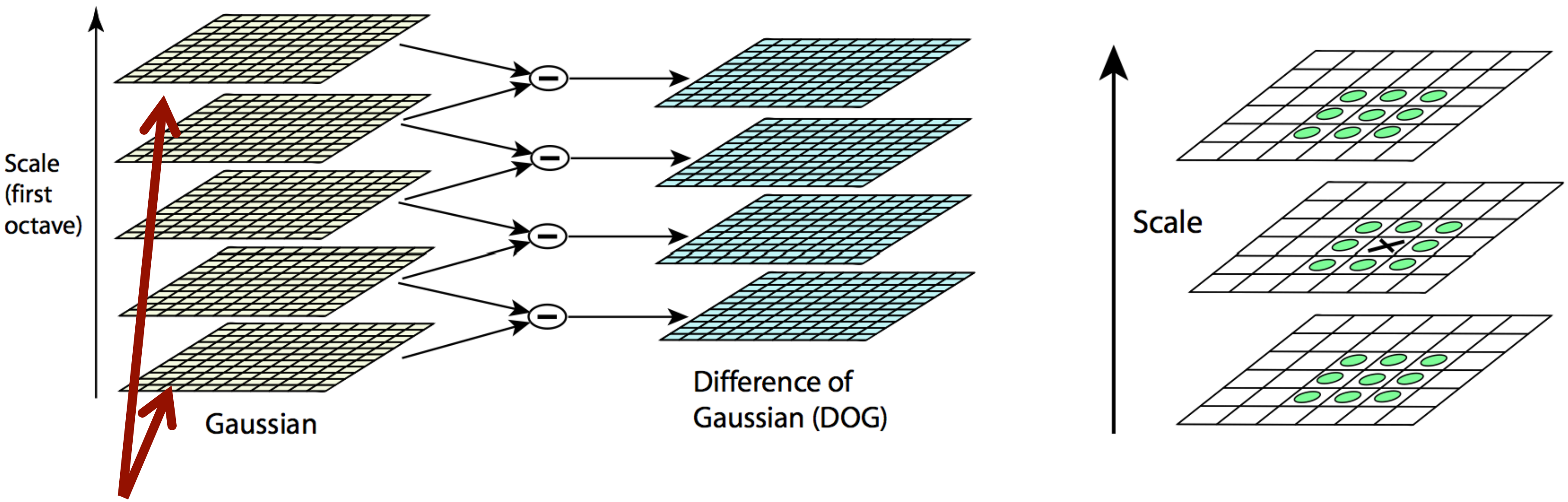
# Illustration




**differently blurred images**

Image courtesy: Lowe

# Illustration




**differently blurred images**

Image courtesy: Lowe

# Illustration

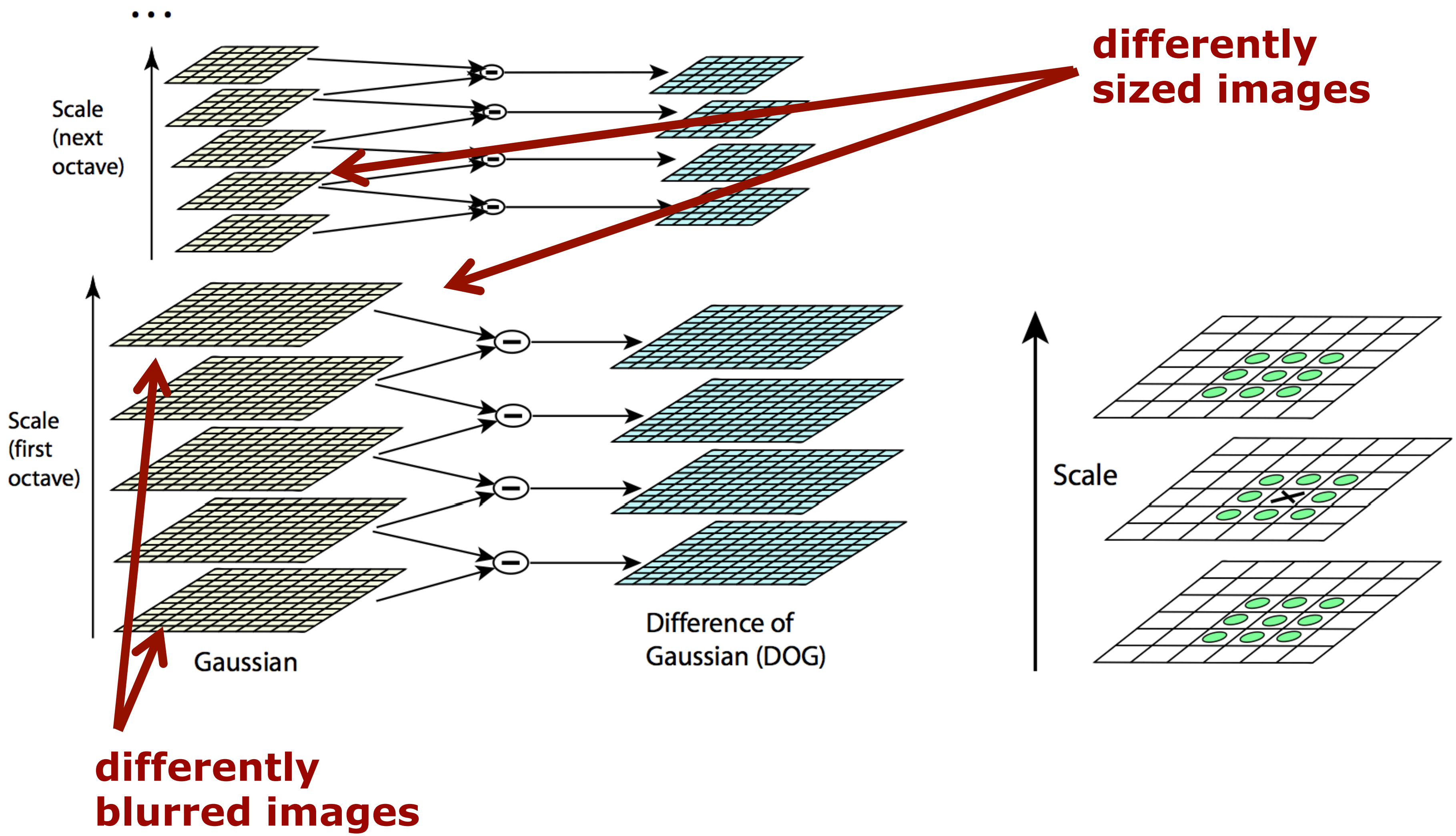



Image courtesy: Lowe

# Difference of Gaussians

- Subtract differently blurred images from each other

- Increases visibility of corners, edges, and other detail present in the image

# Scale Space Representation

t=0, 1, 4, 16, 64, 265

# Difference of Gaussians

- Blurring filters out high-frequencies (noise)
- Subtracting differently blurred images from each other only keeps the frequencies that lie between the blur level of both images
- DoG acts as a band-pass filter

# Difference of Gaussians

**Keypoints are extrema in the DoG over different (smoothing) scales**

# Illustration

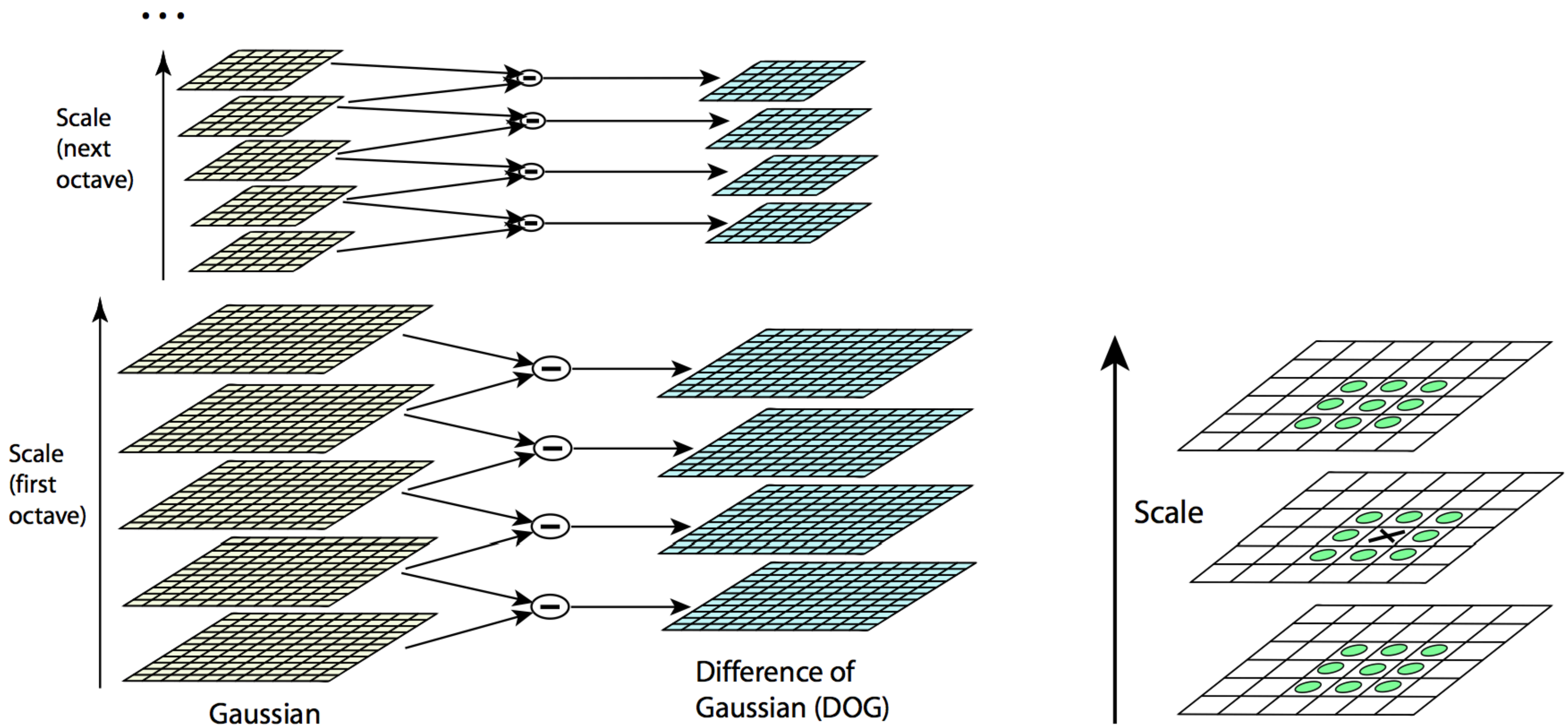



Image courtesy: Lowe

# Extrema Suppression

- The DoG finds blob-like and corner-like image structures but also leads to strong responses along edges
- Edges are bad for matching
- Eliminate edges via Eigenvalue test (similar to Harris corners)

# Keypoints

- Two groups of approaches for finding locally distinct points:
- 1. Corners via structure matrix
  - Harris, Shi-Tomasi, Förstner
- 2. Difference of Gaussians
  - Iterates over scales and blur
  - Finds corners and blobs
- These approaches are key ingredients of most hand-designed features

# Summary

- Keypoints and descriptor together define common visual features
- Keypoint defines the location
- Most keypoints use image gradients
- Corners and blobs are good keypoints

**Outlook: Part 2 – Feature Descriptors**

# Slide Information

- These slides have been created by Cyrill Stachniss as part of the Photogrammetry courses taught in 2014 and 2019
- The slides *heavily* reply on material by Gil Levi, Alexai Efros, James Hayes, David Lowe, and Silvio Savarese
- I tried to acknowledge all people from whom I used images or videos. In case I made a mistake or missed someone, please let me know.
- If you are a university lecturer, feel free to use the course material. If you adapt the course material, please make sure that you keep the acknowledgements to others and please acknowledge me as well. To satisfy my own curiosity, please send me email notice if you use my slides.

Cyrill Stachniss, 2014
cyrill.stachniss@igg.uni-bonn.de