Image Processing & Vision

Lecture 05: Feature Descriptors

Hak Gu Kim

hakgukim@cau.ac.kr

Immersive Reality & Intelligent Systems Lab (IRIS LAB)

Graduate School of Advanced Imaging Science, Multimedia & Film (GSAIM)

Chung-Ang University (CAU)

03 Apr. 2023

Recap: Local Feature

- Global template matching ⟶ **Local feature detection**
- Consider the problem of finding images of an elephant using a template

— An elephant looks different from different viewpoints

— What happens if parts of an elephant are obscured from view by trees, rocks, other elephants?

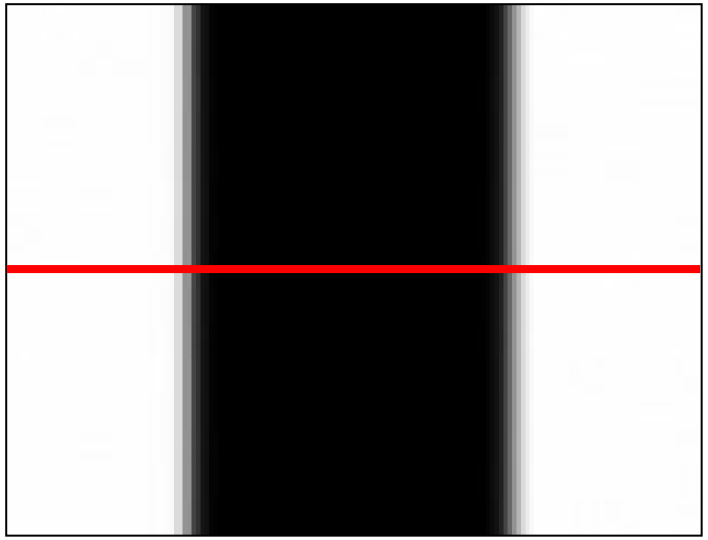
Template

Image

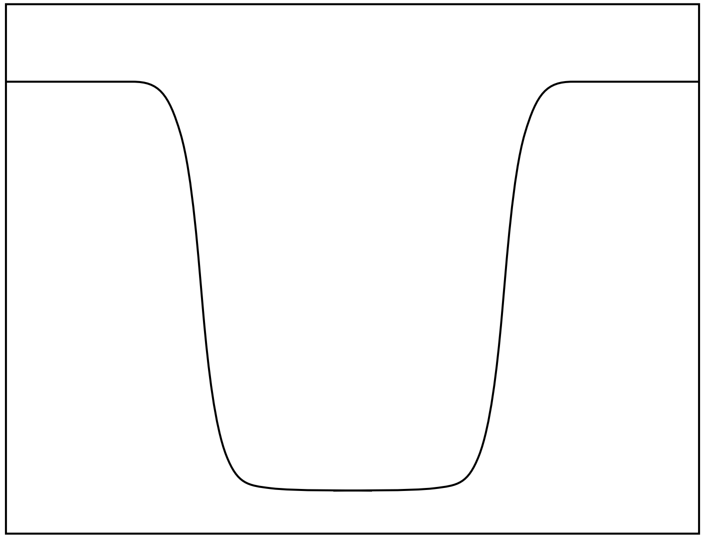
Normalized correlation map

Recap: Edges

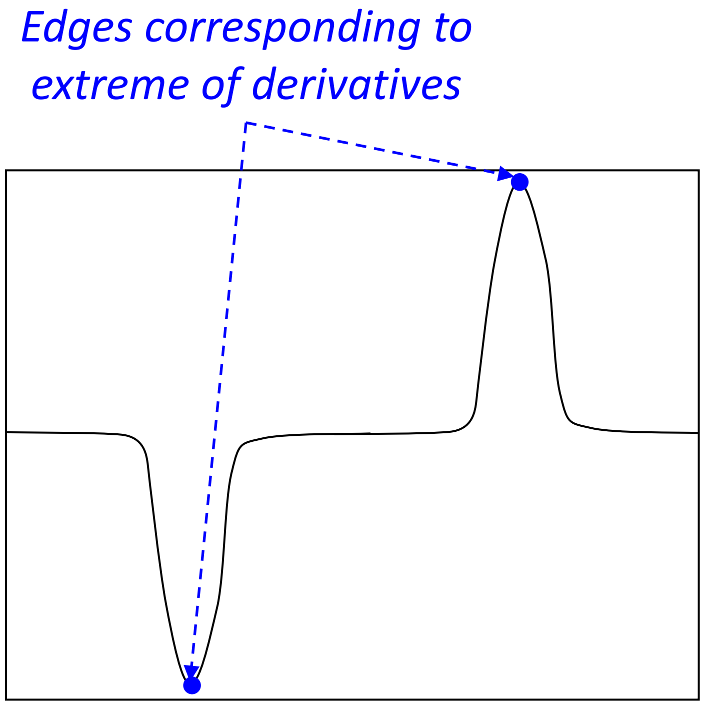
- An edge is a place of rapid change in the image intensity function



Image



Intensity function
along horizontal scanline



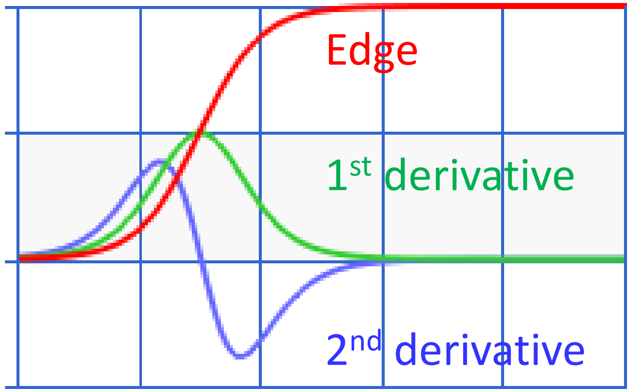
1^{st} derivative
in horizontal axis

Recap: DoG & LoG

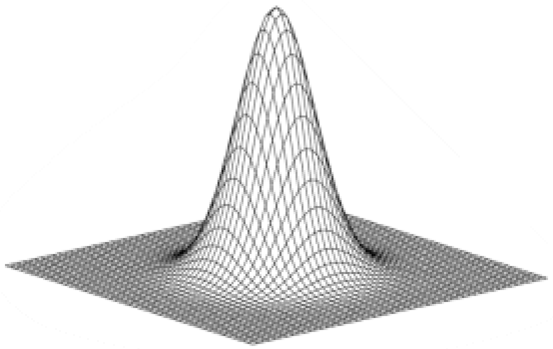
- **Two Generic Edge Detection Approaches:**

— Local extrema of a first derivative operator

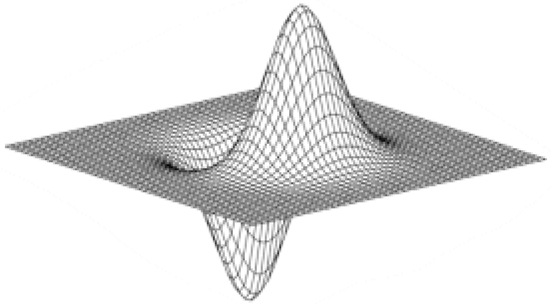
— Zero crossings of a second derivative operator



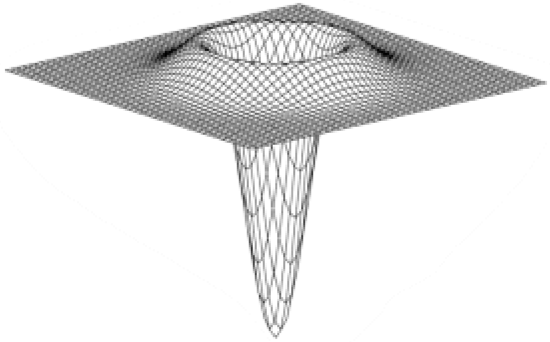
- **Laplacian of Gaussian** is based on **a zero crossings of a second derivative** operator method



G_σ



∇G_σ



$\nabla^2 G_\sigma$

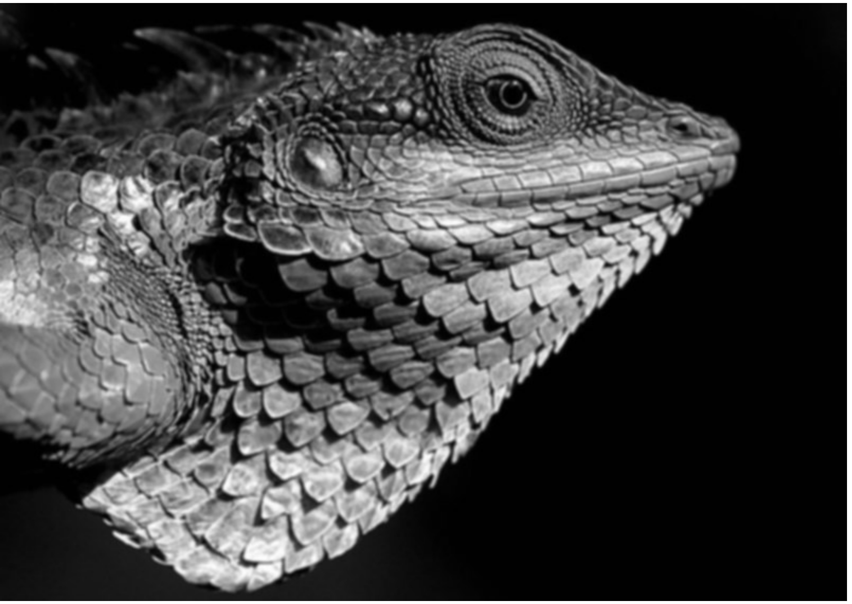
Recap: Canny Edge Detection

① Apply **directional derivatives of Gaussian**

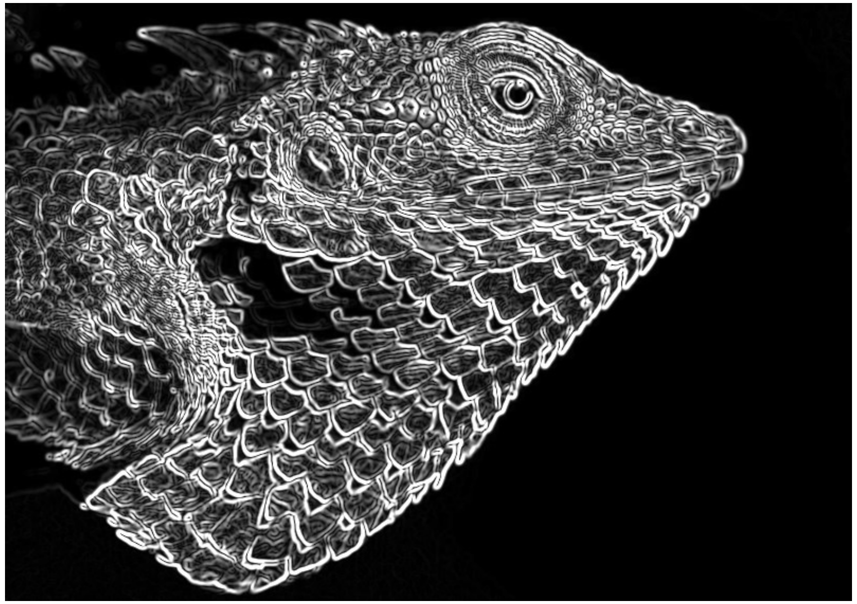
② Compute **gradient magnitude** and **gradient direction**

③ **Non-maximum suppression (NMS)**

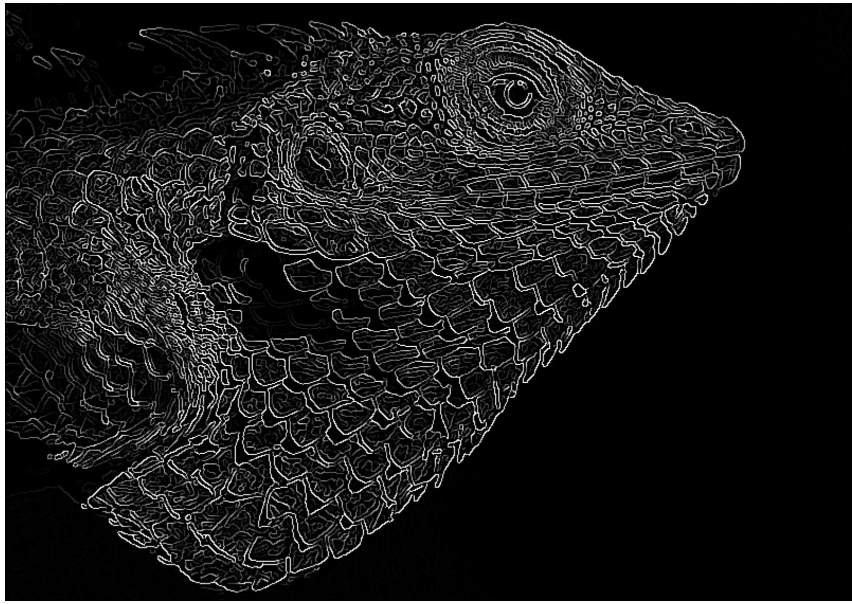
④ **Link** and **threshold**



Original image



Gradient magnitude



Result of non-maximum suppression

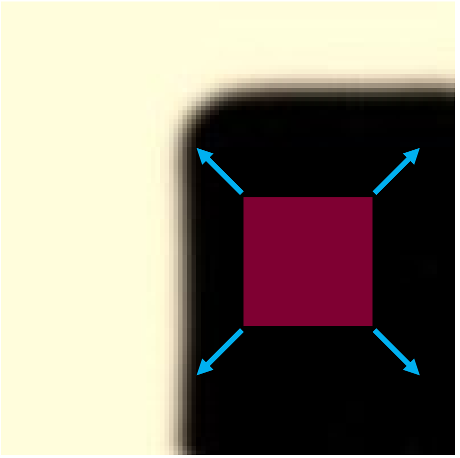
Recap: Corner

- **Key Property:**

— In the region around a corner, image gradient has two or more dominant directions

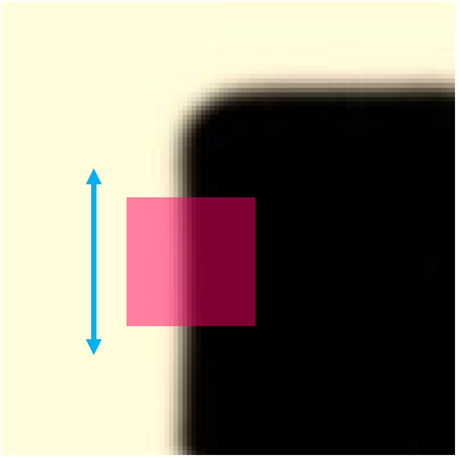
— Corners are robust and distinctive

Flat



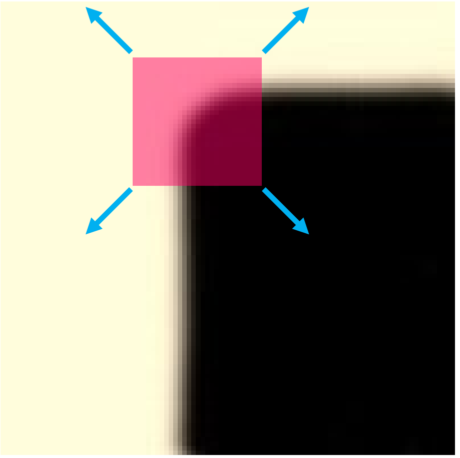
No changes
in all directions

Edge



No changes
along the edge direction

Corner



Significant changes
in all directions

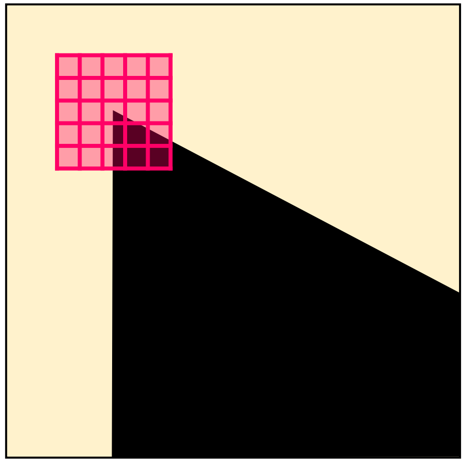
Recap: Harris Corner Detection

① Compute **image gradient** over small region

② Compute the **covariance matrix**

③ Compute **eigenvectors** and **eigenvalues**

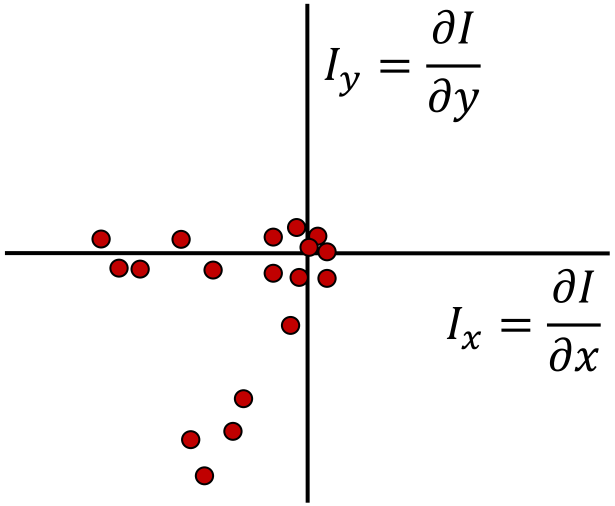
④ Use **threshold on eigenvalues** to detect corners



Sum over local window region around corner

Gradient with respect to each direction (x or y)

$$\mathbf{M} = \begin{bmatrix} \sum_{p \in P} I_x I_x & \sum_{p \in P} I_x I_y \\ \sum_{p \in P} I_y I_x & \sum_{p \in P} I_y I_y \end{bmatrix}$$



Recap: Harris Corner Detection

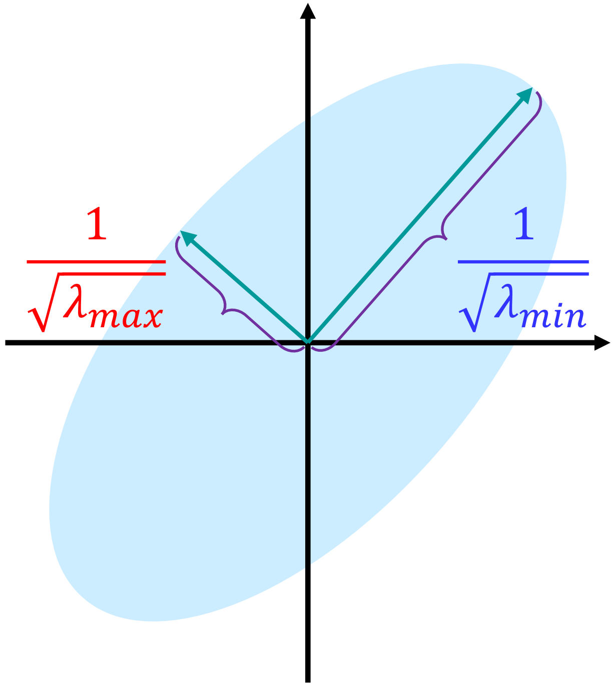
- **Visualization** of Eigenvalues and Eigenvectors as an **Ellipse**

— **Eigenvectors** determines the **orientation of the ellipse**

— **Eigenvalues** determines the axis **lengths of the ellipse**

- $\frac{1}{\sqrt{\lambda_{max}}}$: The direction of the fastest change
- $\frac{1}{\sqrt{\lambda_{min}}}$: The direction of the slowest change

$$\mathbf{M} = \begin{bmatrix} \sum_{p\in P} I_x I_x & \sum_{p\in P} I_x I_y \\ \sum_{p\in P} I_y I_x & \sum_{p\in P} I_y I_y \end{bmatrix} = \underbrace{[\mathbf{v}_1 \quad \mathbf{v}_2]}_{\text{Eigenvectors}} \underbrace{\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}}_{\text{Eigenvalues}} [\mathbf{v}_1 \quad \mathbf{v}_2]^{-1}$$



Recap: Harris Corner Detection

- Use **threshold on eigenvalues** to detect corners

- **Cornerness:**

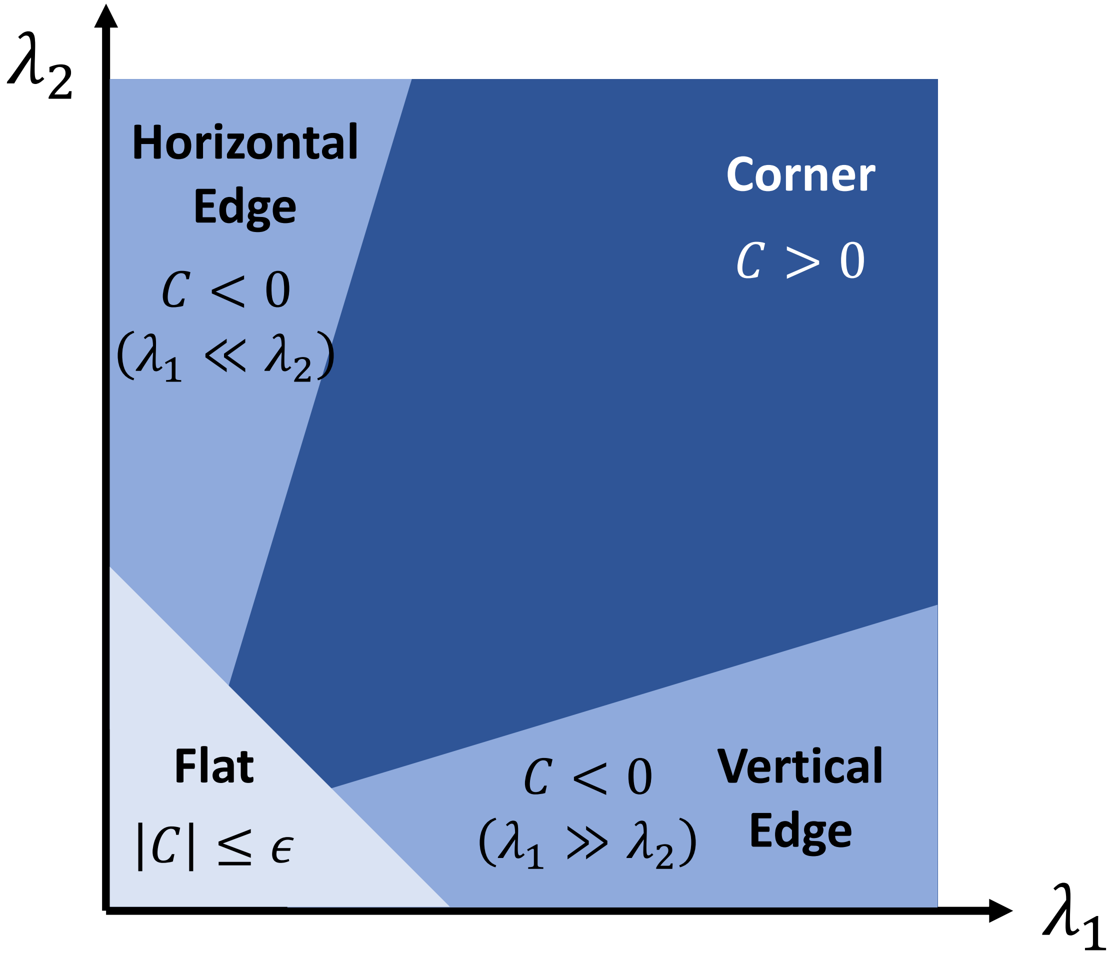
$$C = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$
$$= \det(\mathbf{M}) - \alpha \cdot \mathrm{tr}(\mathbf{M})^2$$

$0.04 \leq \alpha \leq 0.06$

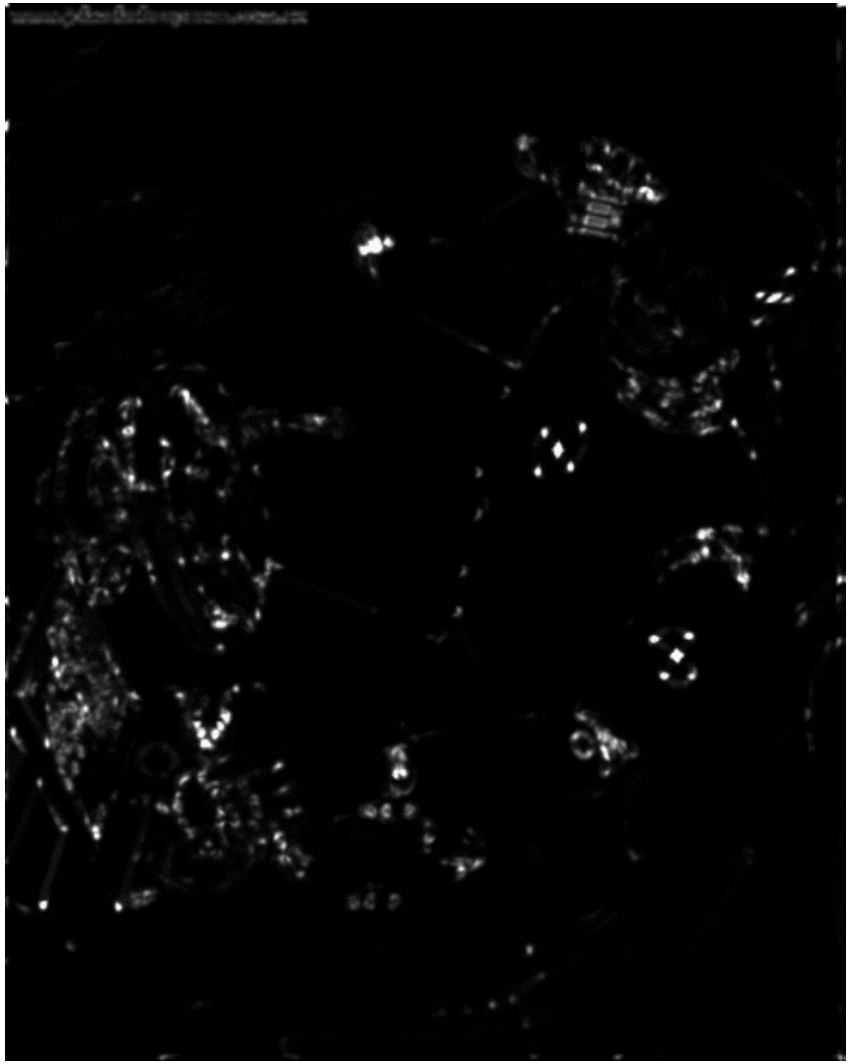
— Flat region: $|C| \leq \epsilon$

— Edge: $C < 0$

— Corner: $C > 0$



Recap: Harris Corner Detection



Corner response map

Corner detection result ($\sigma = 1$)

Topics

- Feature Descriptors

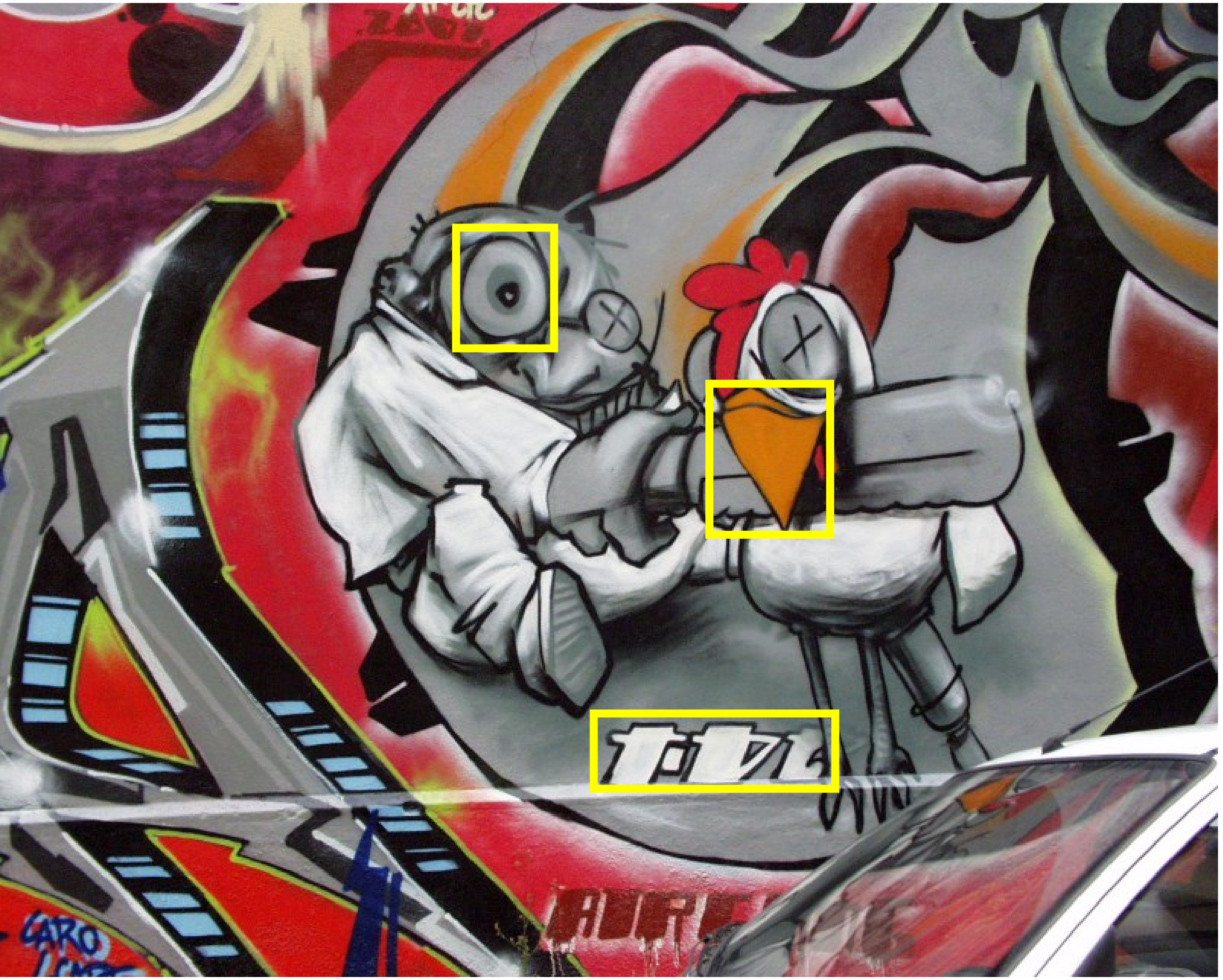
— Basic Feature Descriptors

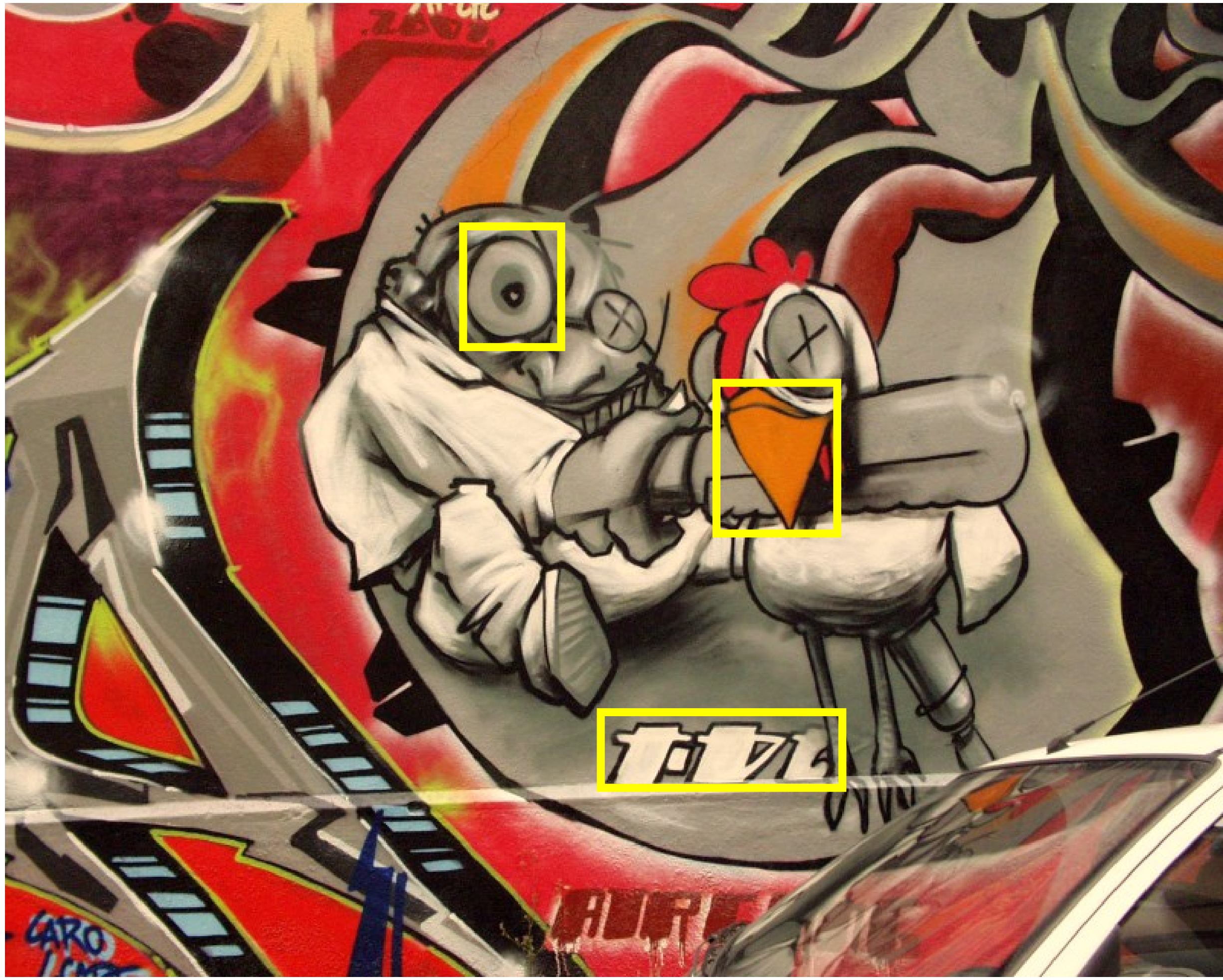
— SIFT

*Note:** Many of these slides in this course were adapted from Computer Vision at CMU (16-385) and UBC (CPSC425)

Difficulty of Matching: Photometric Transformations

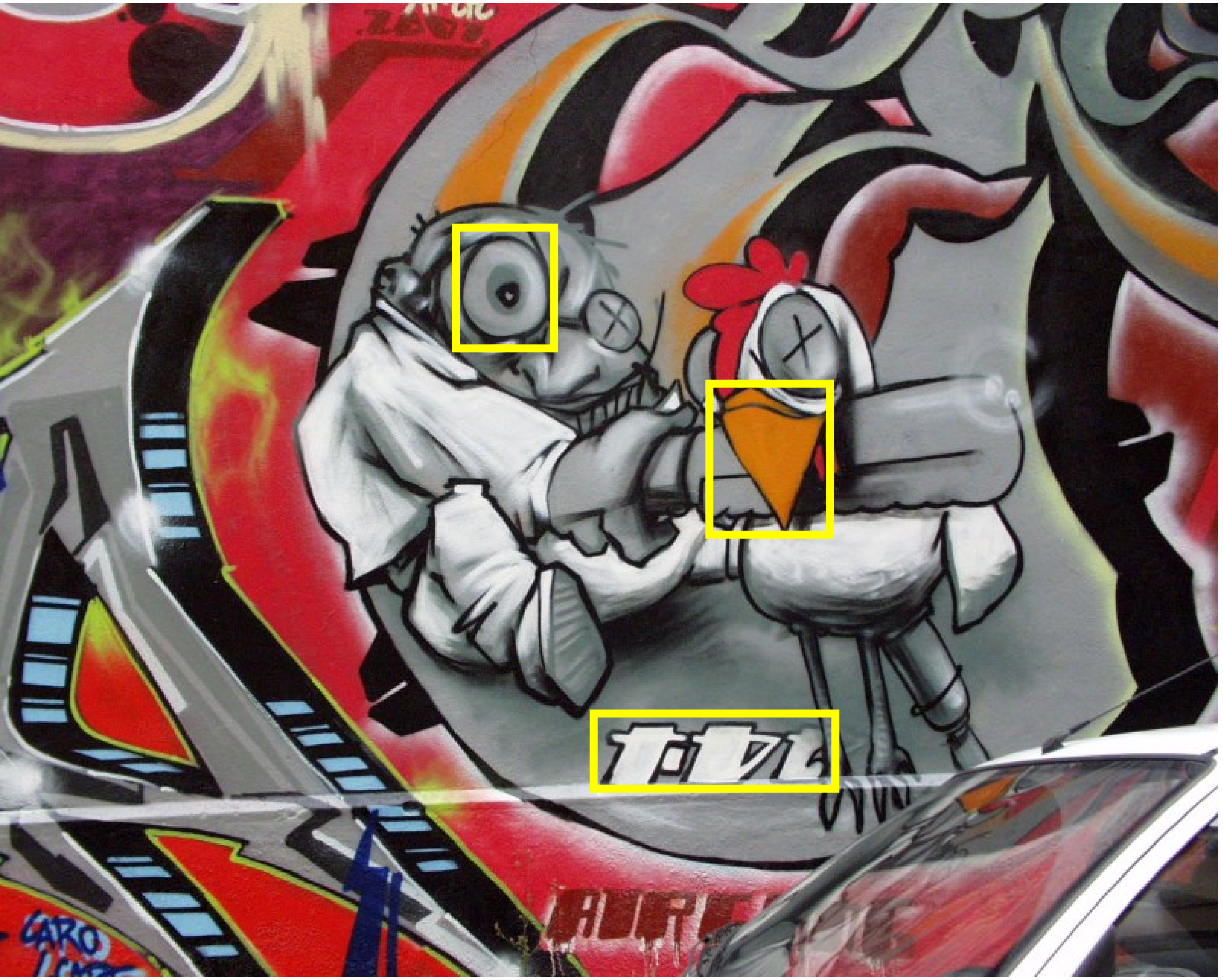
- How do we match the correspondences between two images?

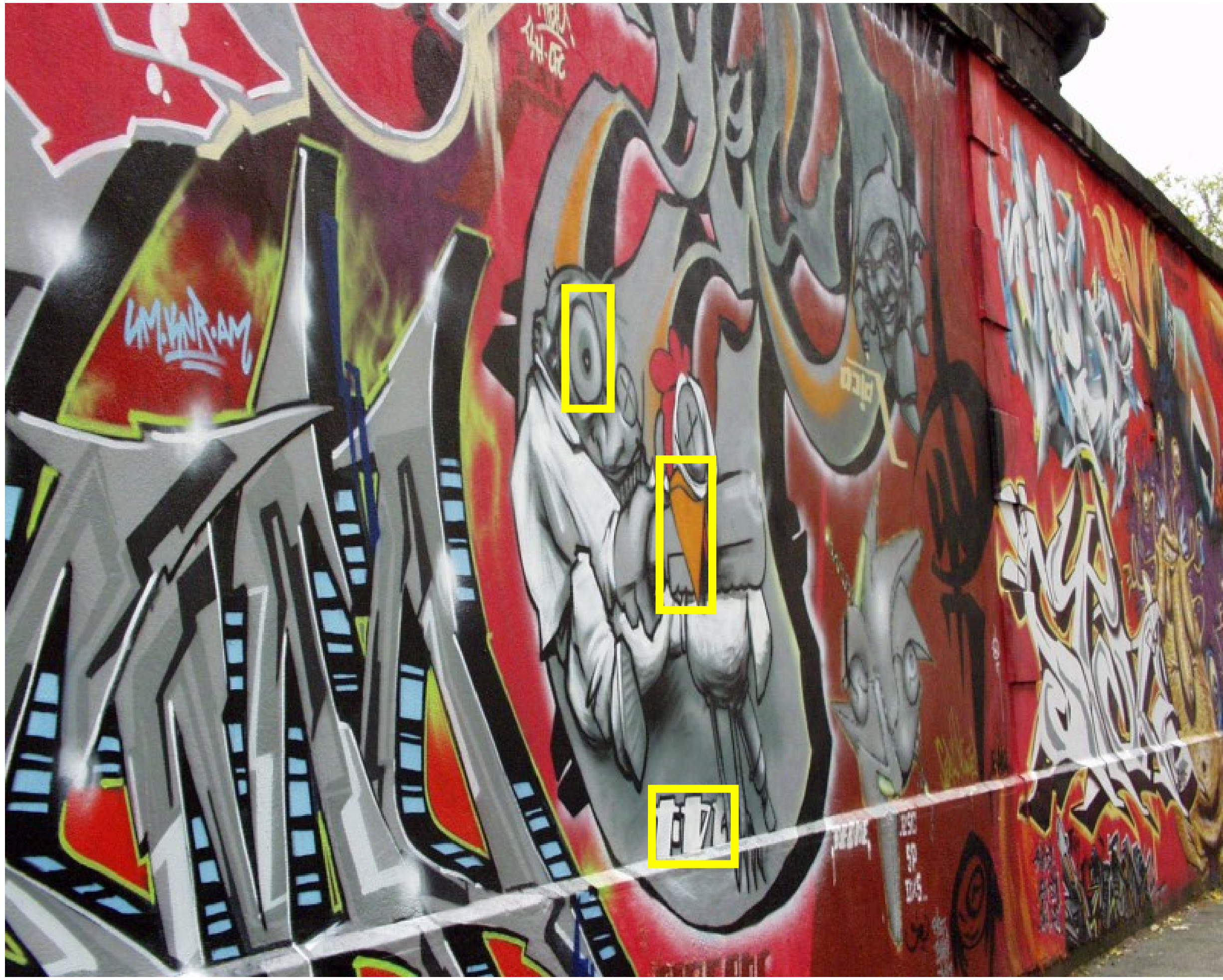




Difficulty of Matching: **Geometric** Transformations

- How do we match the correspondences between two images?



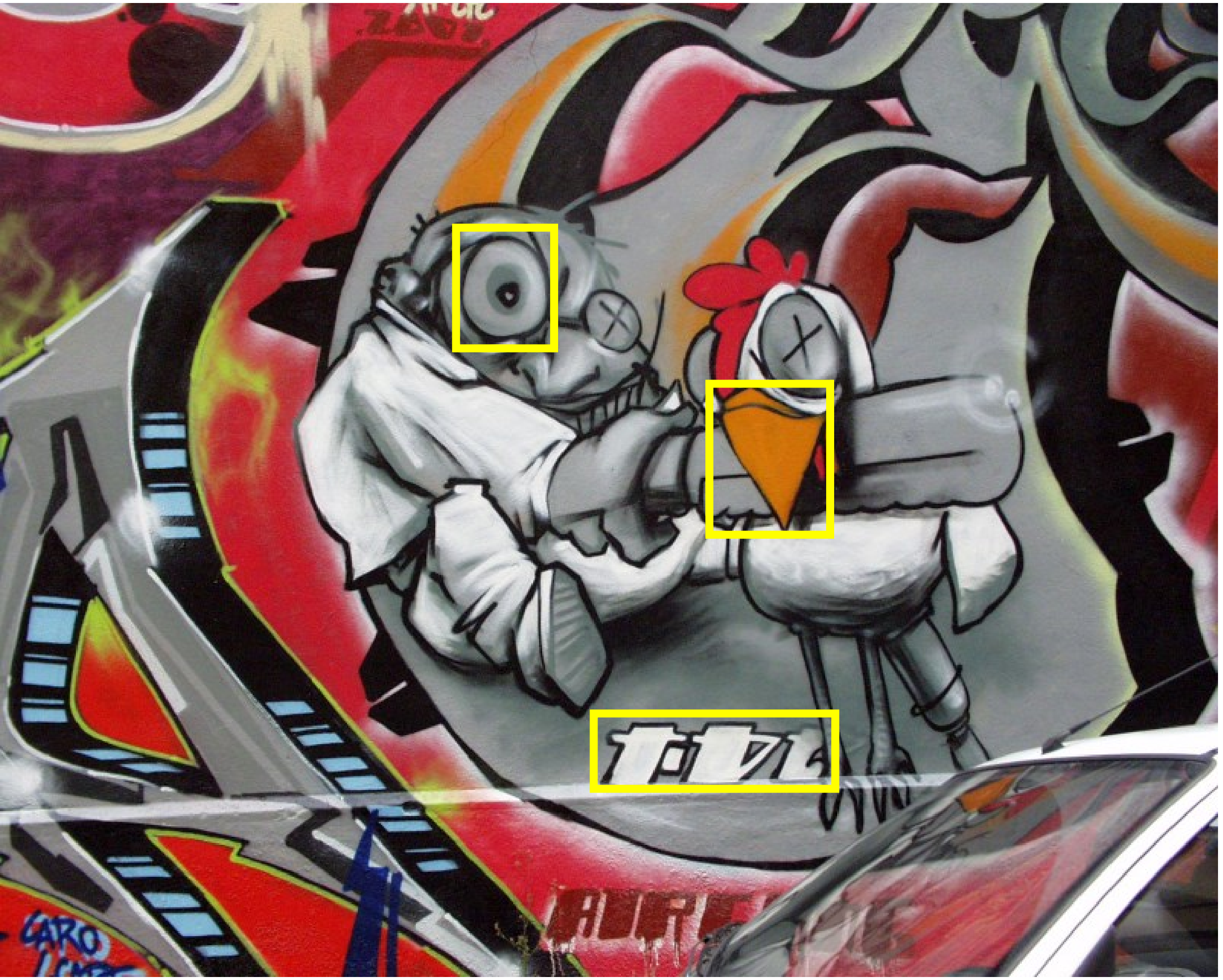


Good Local Features

- How do we know which **corner** goes with which?

Good Local Features

- **Patch** around the local feature is much more **informative**



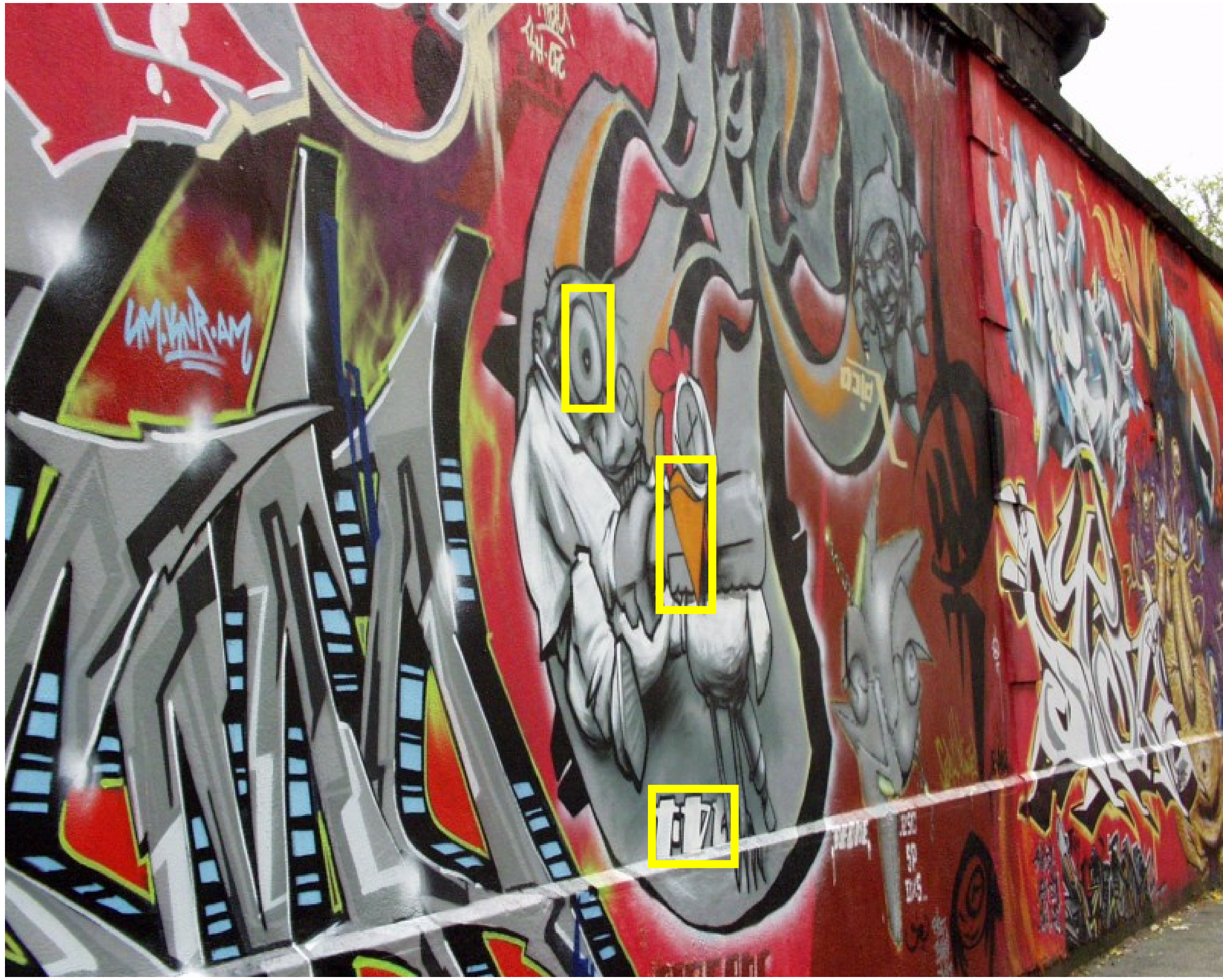
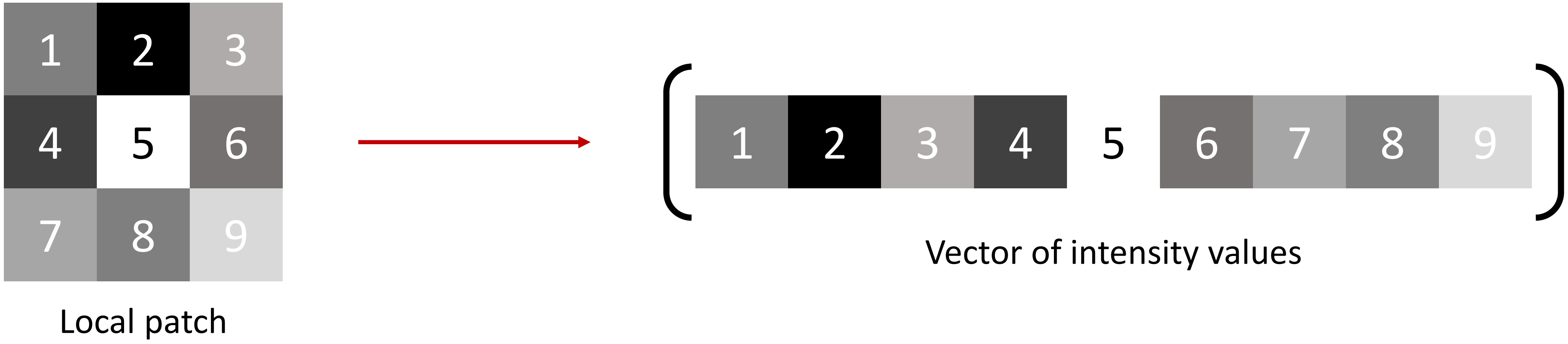


Image **Intensity**

- Just use the **pixel values** of the patch

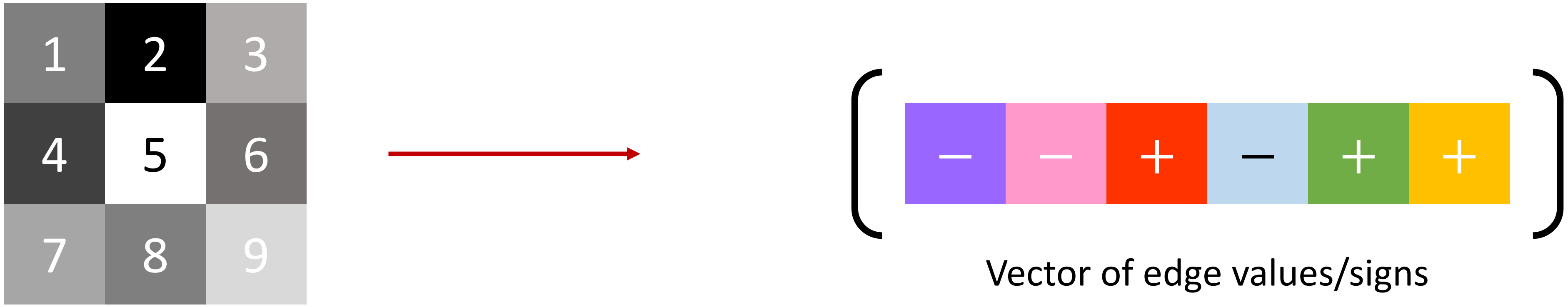


— **Pros:** Perfectly fine if geometry and appearance is unchanged

— **Cons:** Very sensitive to absolute intensity values

Image **Gradients** & **Edges**

- Just use the **Edge values/signs** of the patch



Local patch

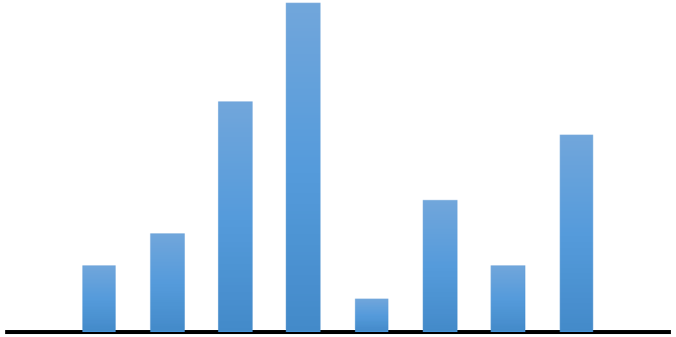
Vector of edge values/signs

— **Pros:** Invariant to absolute intensity values

— **Cons:** Very sensitive to deformations

Color Histogram

- Count the colors in the image using a histogram



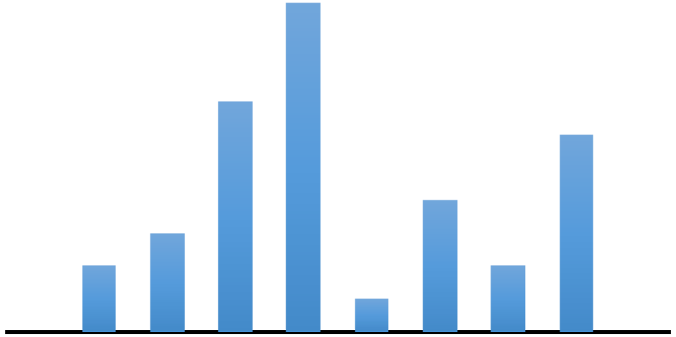
Color histogram

— **Pros:** Invariant to changes in scale and rotation

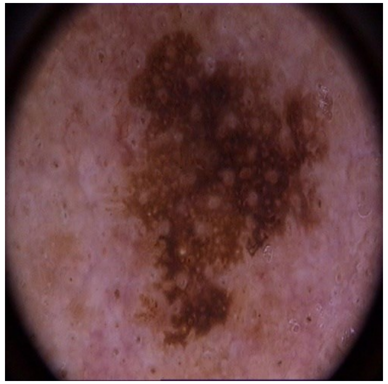
— **Cons:** Ignoring its shape and texture (i.e., spatial layout)

Color Histogram

- Count the colors in the image using a histogram



Color histogram

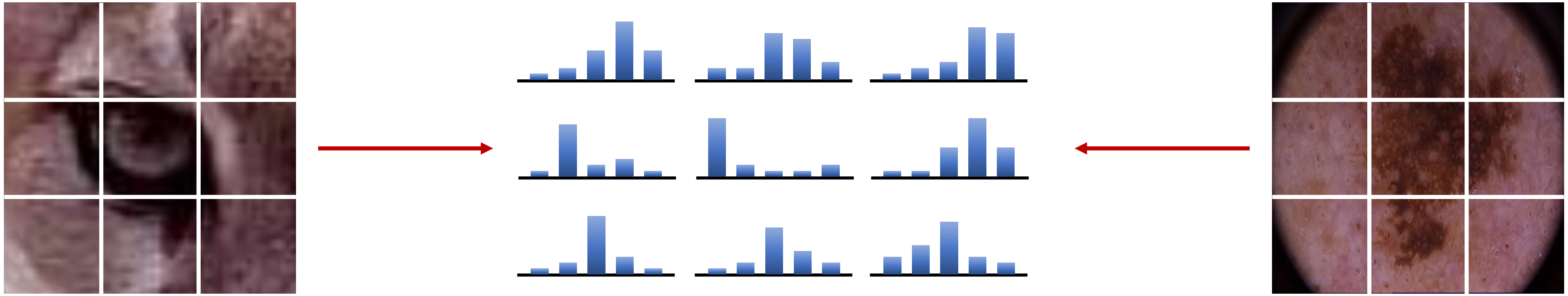


— **Pros:** Invariant to changes in scale and rotation

— **Cons:** Ignoring its shape and texture (i.e., spatial layout)

Spatial Histograms

- Count the colors **in each local region** using a histogram



Spatial histograms

— **Pros:** Invariant to some deformations and retains rough spatial layout

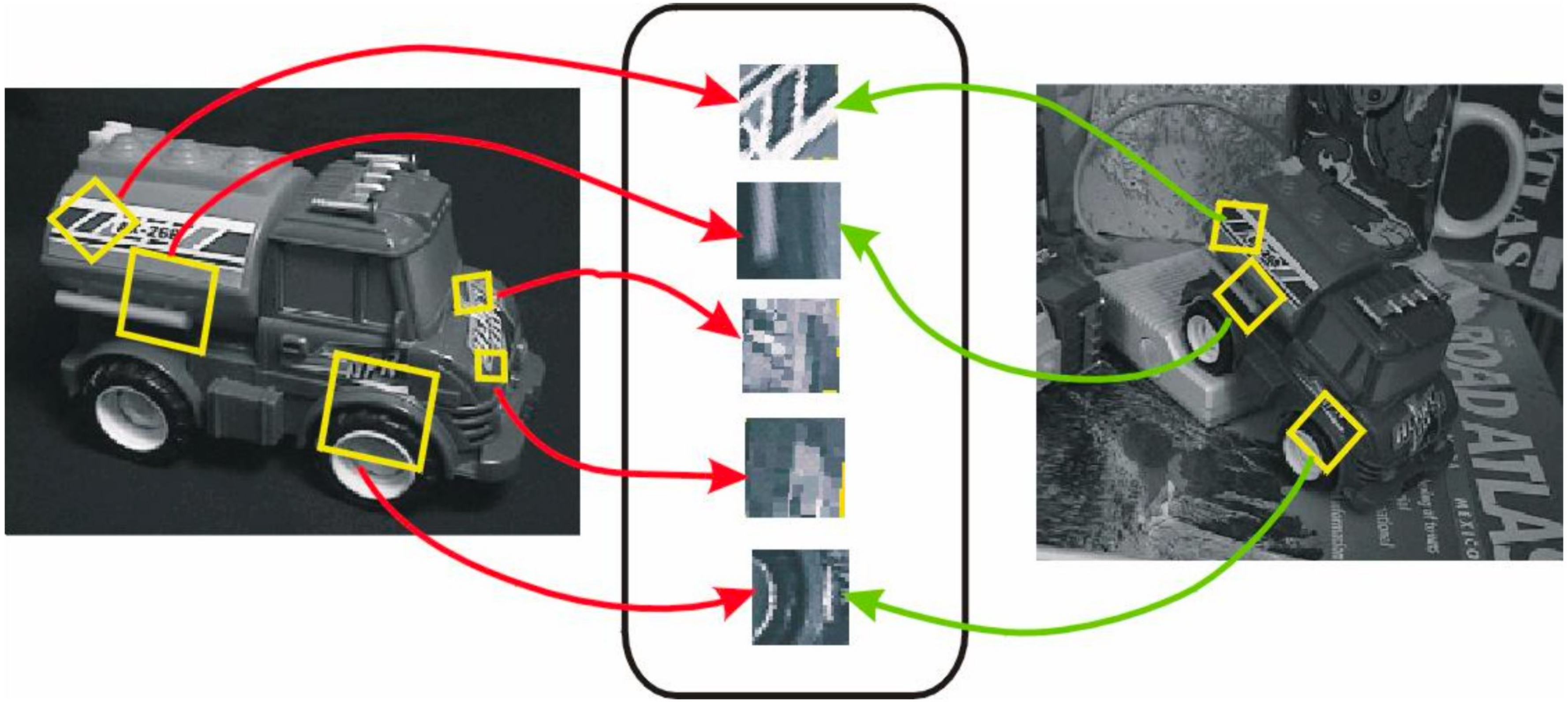
— **Cons:** Sensitive to rotation

Summary of Features

Representation	Result	Approach	Technique
Intensity	Dense (2D)	Template Matching	Normalized correlation, Sum of squared difference (SSD)
Edge	Relatively sparse (1D)	Derivatives	$\nabla^2 G$, Canny edge
Corner / Blob	Sparse (0D)	Locally distinct features	Harris corner, SIFT

Invariant Local Features

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



SIFT Features

Advantages of Invariant Local Features

— **Locality:** Features are local, so robust to occlusion and clutter

— **Distinctiveness:** Individual features can be matched to a large-scale database of objects

— **Quantity:** Many features can be generated for even small objects

— **Efficiency:** Close to real-time performance

Scale Invariant Feature Transform (SIFT)

- The scale-invariant feature transform (SIFT) is an algorithm to **detect**, **describe**, and **match** local features in images
- SIFT describes both a **detector** and **descriptor**

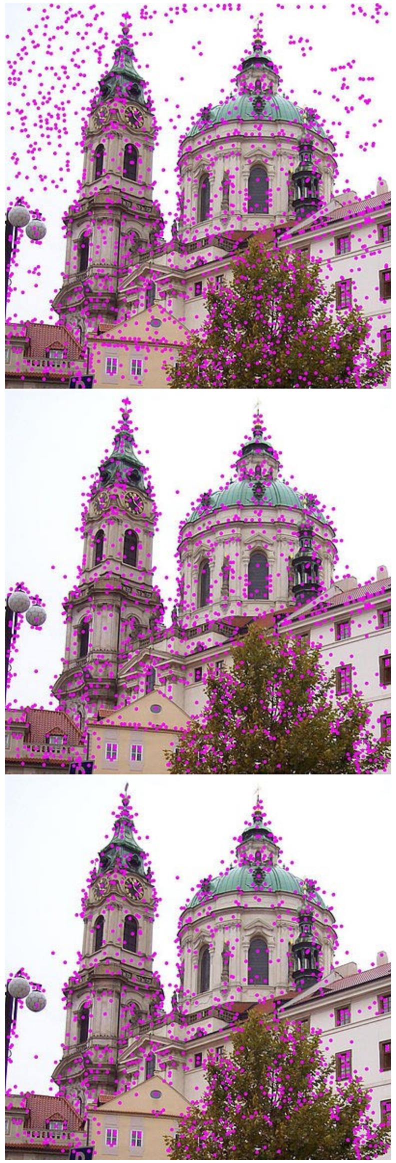
— Applications: object recognition, robotic mapping and navigation, image stitching, 3D modeling, gesture recognition, video tracking, *etc*.

① Multi-scale extrema detection

② Keypoint localization

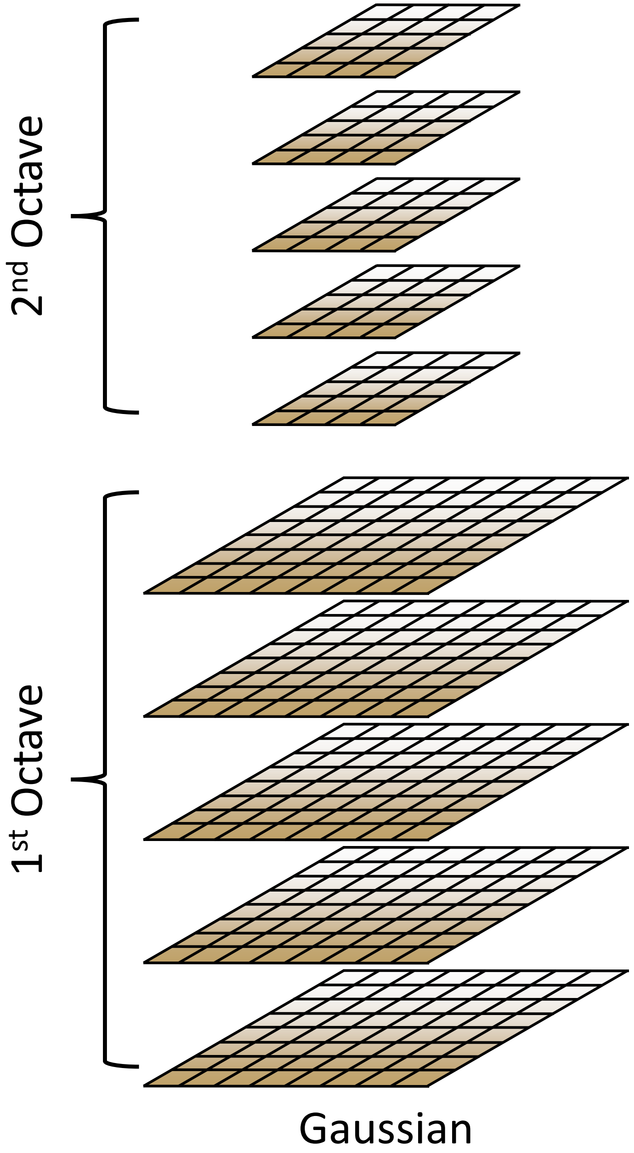
③ Orientation assignment

④ Keypoint descriptor



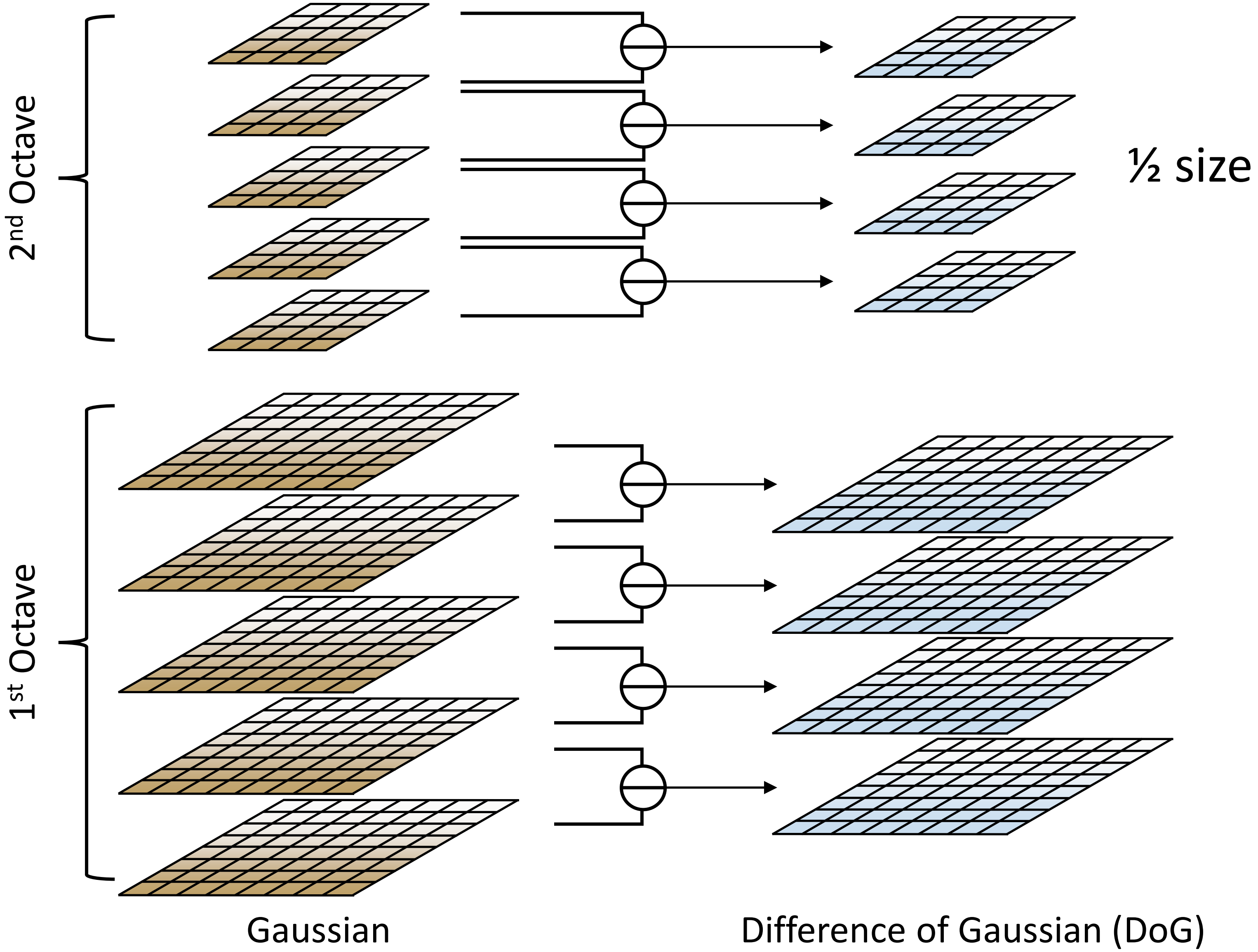
SIFT: **Multi-Scale Extrema Detection**

- Begin by detecting points of interest (i.e., keypoints)
  — The image is convolved with Gaussian filters at different scales



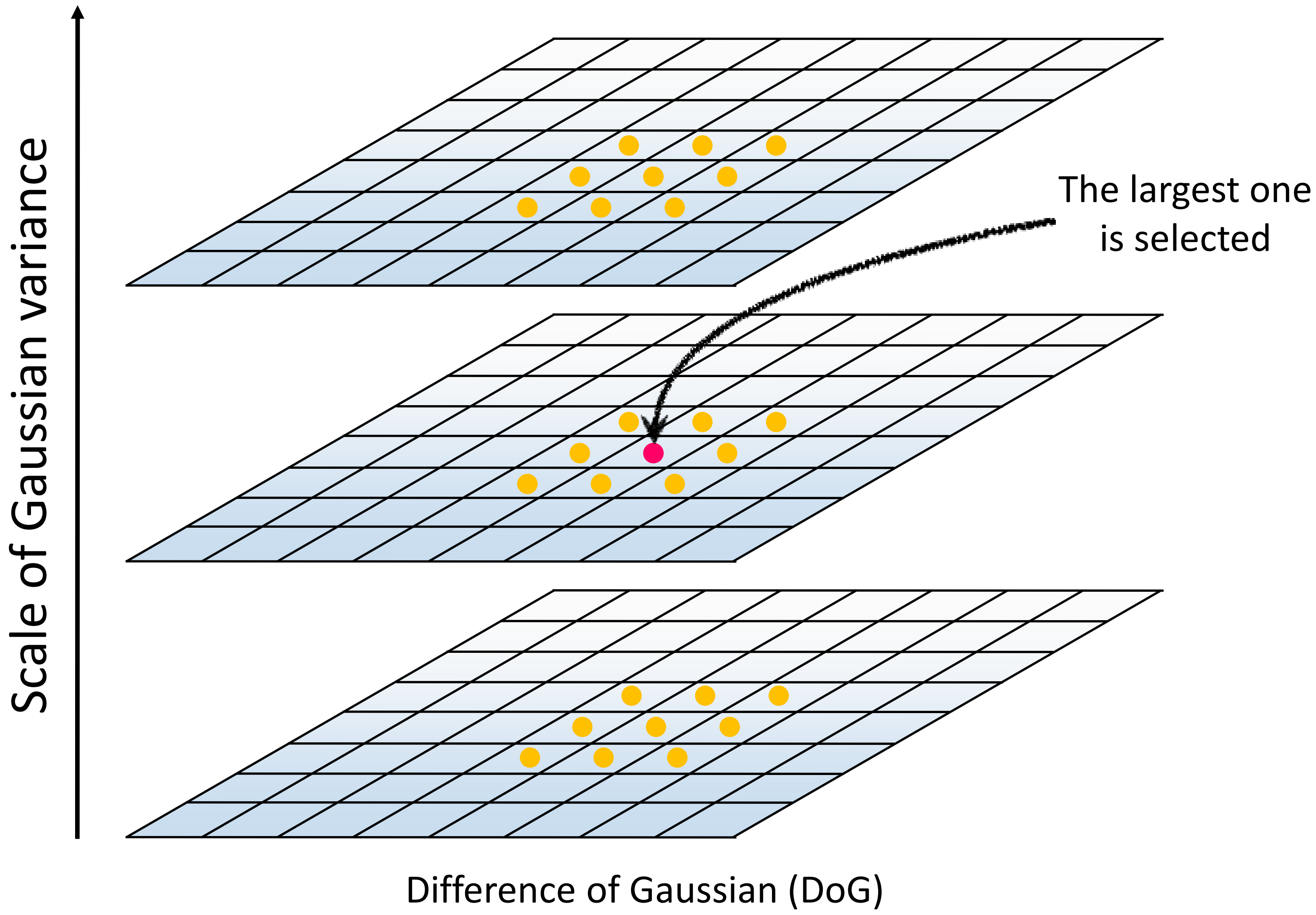
SIFT: **Multi-Scale Extrema Detection**

- Begin by detecting points of interest (i.e., keypoints)
  - The image is convolved with Gaussian filters at different scales
  - Then the difference of successive Gaussian-blurred images are taken



SIFT: **Multi-Scale Extrema Detection**

- Begin by detecting points of interest (i.e., keypoints)
  - The image is convolved with Gaussian filters at different scales
  - Then the difference of successive Gaussian-blurred images are taken
  - Keypoints are taken as maxima/minima of the DoG at multiple scales



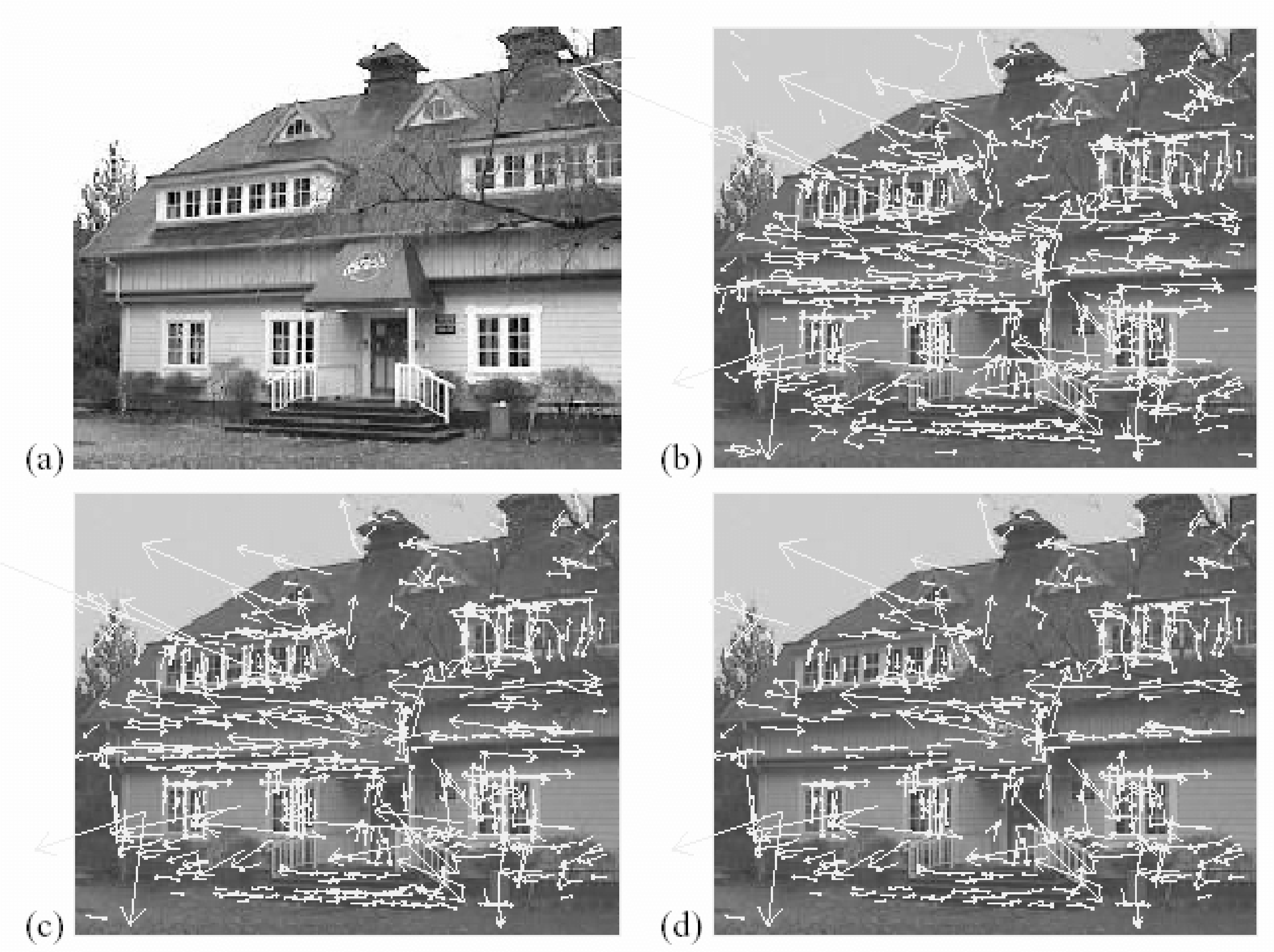
SIFT: **Keypoint Localization**

- Scale-space extrema detection produces too many keypoint candidates, some of which are **unstable**
- In keypoint localization, we reject points which are **low contrast** (and are therefore **sensitive to noise**) or **poorly localized** along an edge

- How do we decide whether a keypoint is poorly localized or well-localized?

— In SIFT, compute the ratio of the eigenvalues of covariance matrix and check if it is greater than a threshold

SIFT: **Keypoint Localization**



(a) 233 x 189 image

(b) 832 DoG extrema in (a)

(c) 729 left after peak value threshold

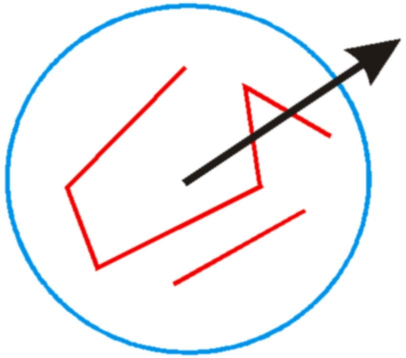
(d) 536 left after testing ratio of principal curvatures

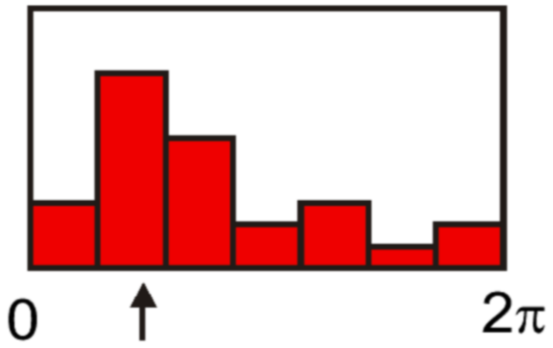
SIFT: **Orientation Assignment**

- Each keypoint is assigned one or more orientations based on local image gradient directions
- This is the key step in achieving invariance to rotation on the Gaussian-smoothed image, L

$$m(x,y) = \sqrt{\big(L(x+1,y) - L(x-1,y)\big)^2 + \big(L(x,y+1) - L(x,y-1)\big)^2}$$

$$\theta(x,y) = \tan^{-1}\Big[\big(L(x+1,y) - L(x-1,y)\big)/\big(L(x,y+1) - L(x,y-1)\big)\Big]$$

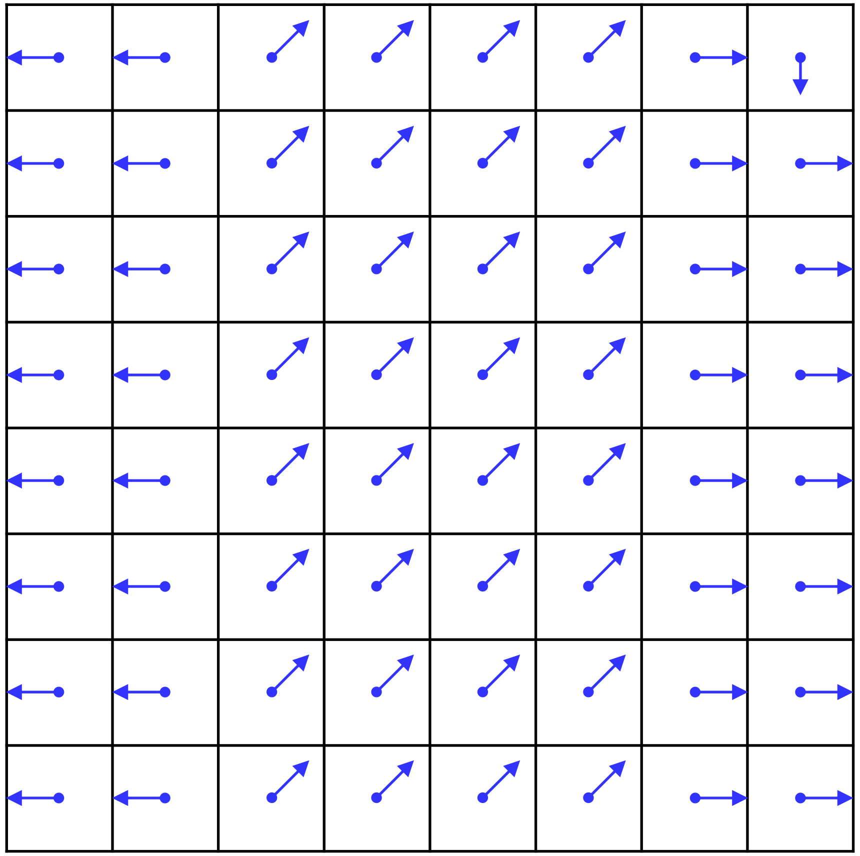


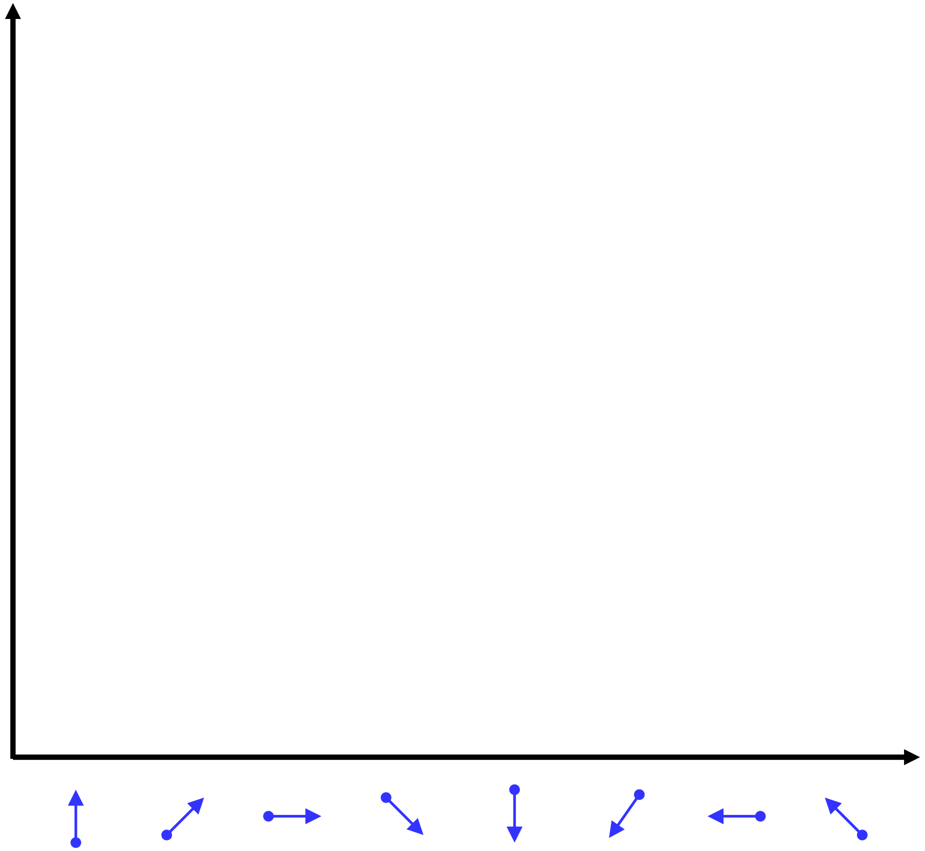


SIFT: **Orientation Assignment**

— **Gradient orientation:** Direction of arrows

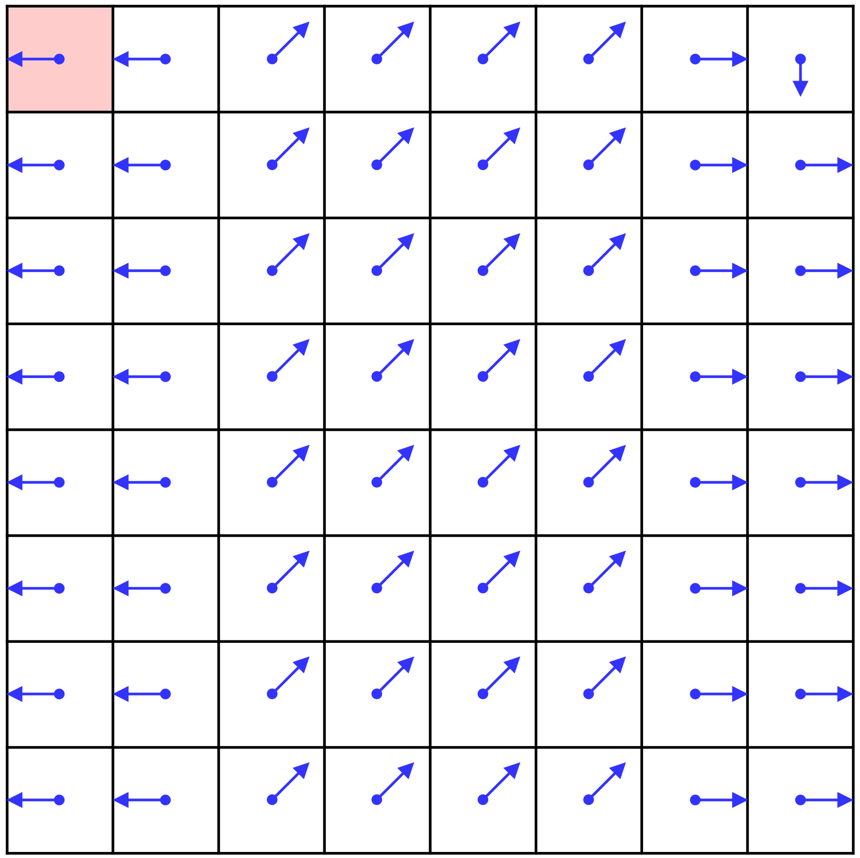
— **Gradient magnitude:** Length of arrows

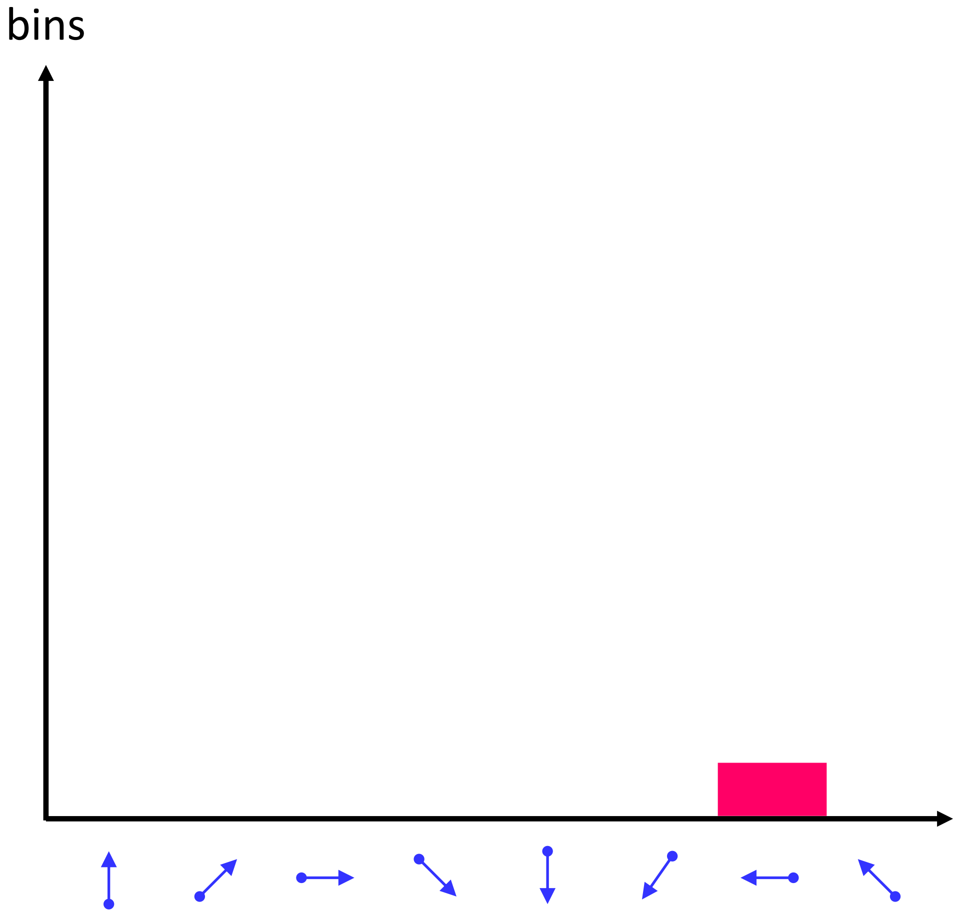




SIFT: **Orientation Assignment**

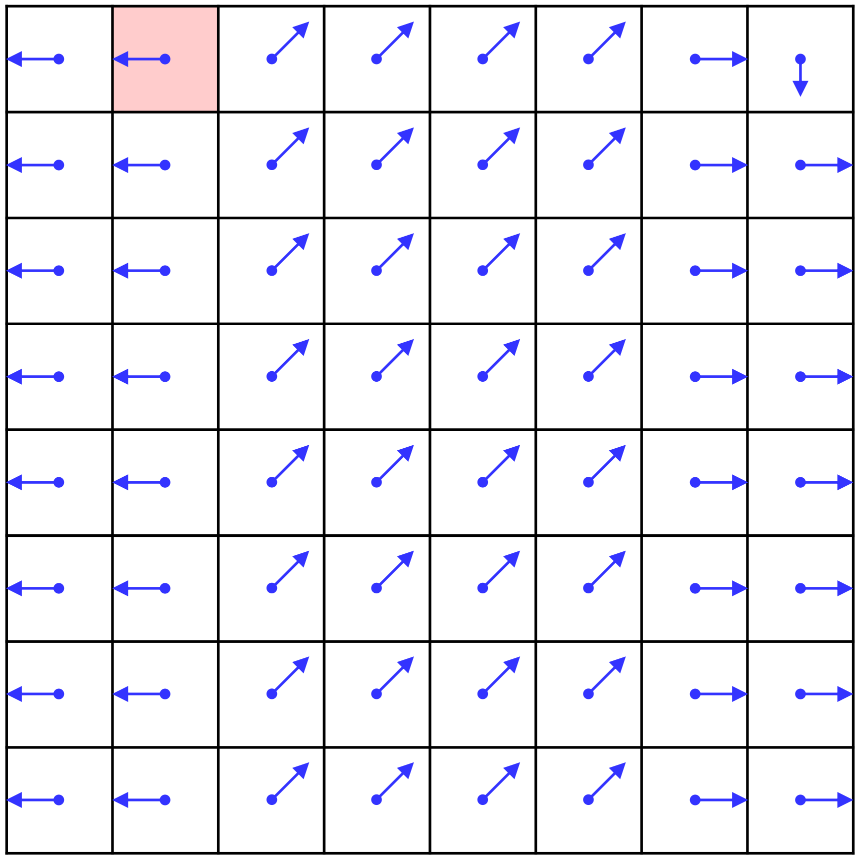
- **Gradient orientation:** Direction of arrows
- **Gradient magnitude:** Length of arrows

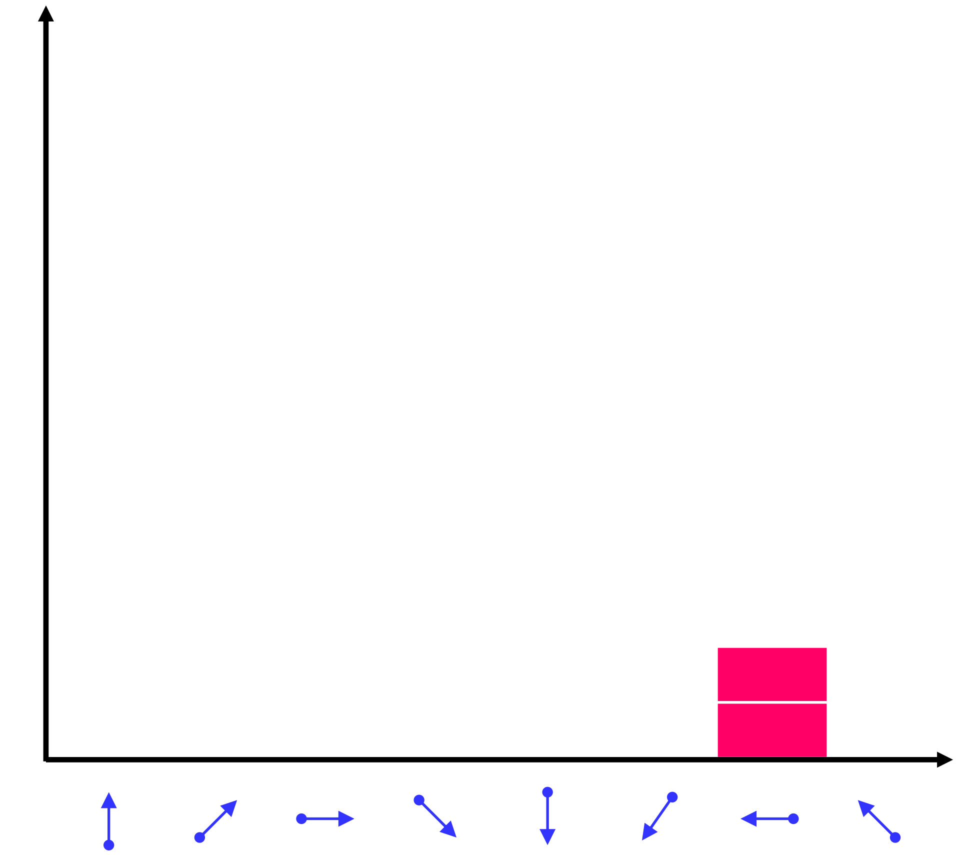




SIFT: **Orientation Assignment**

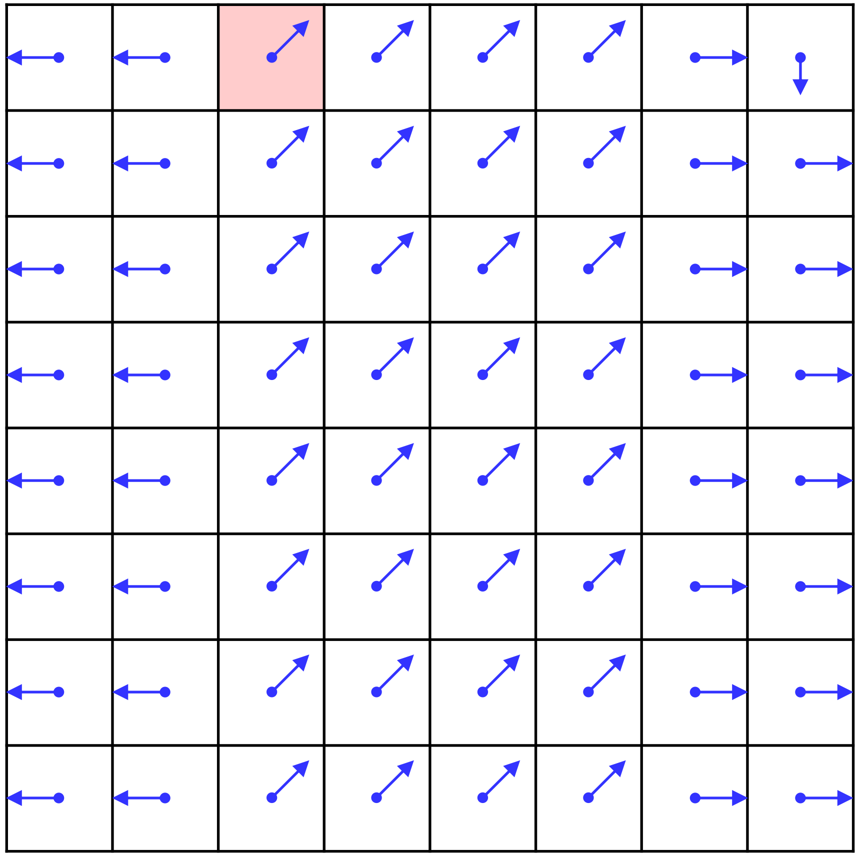
- **Gradient orientation:** Direction of arrows
- **Gradient magnitude:** Length of arrows

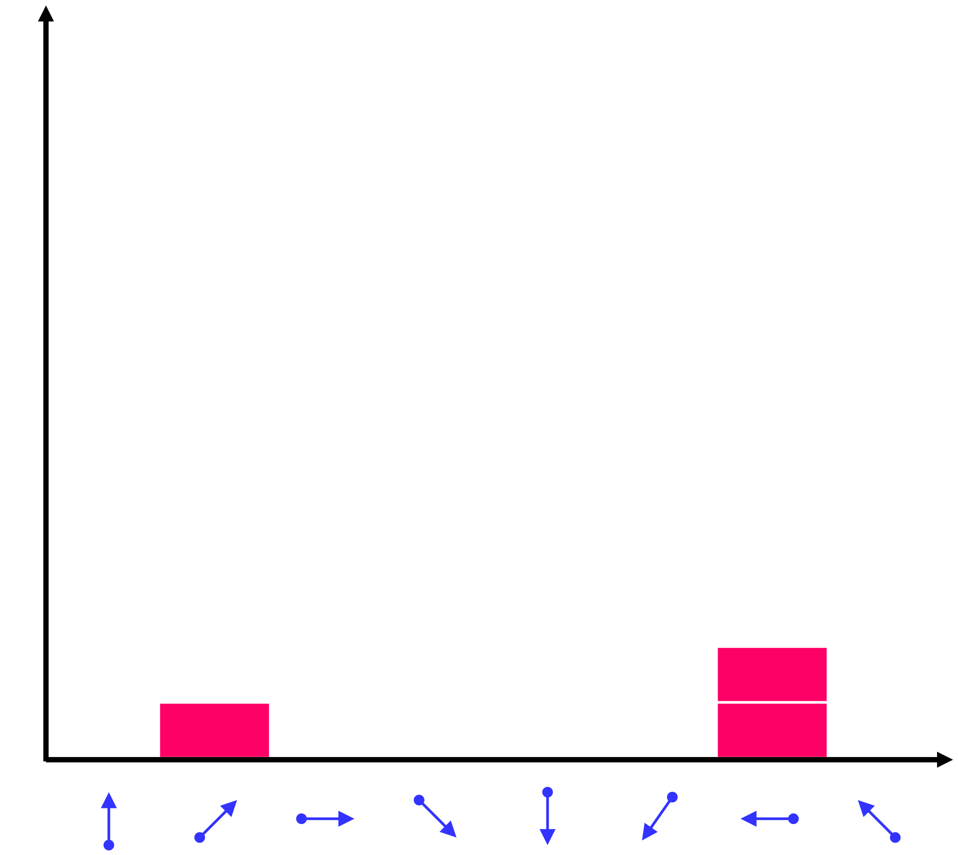




SIFT: **Orientation Assignment**

- **Gradient orientation:** Direction of arrows
- **Gradient magnitude:** Length of arrows

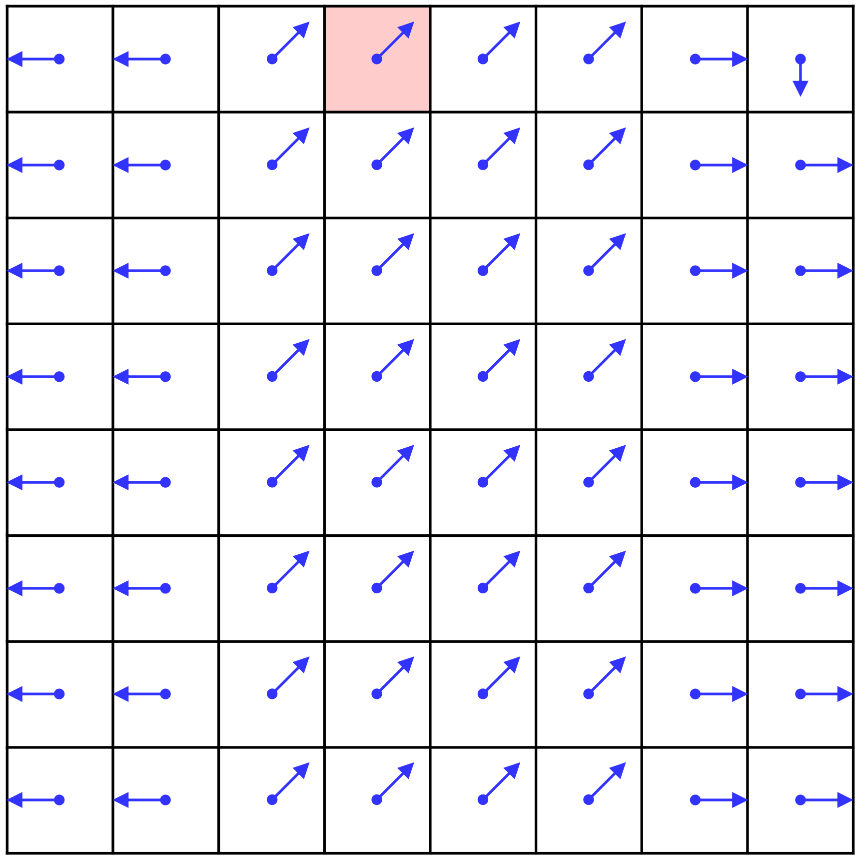


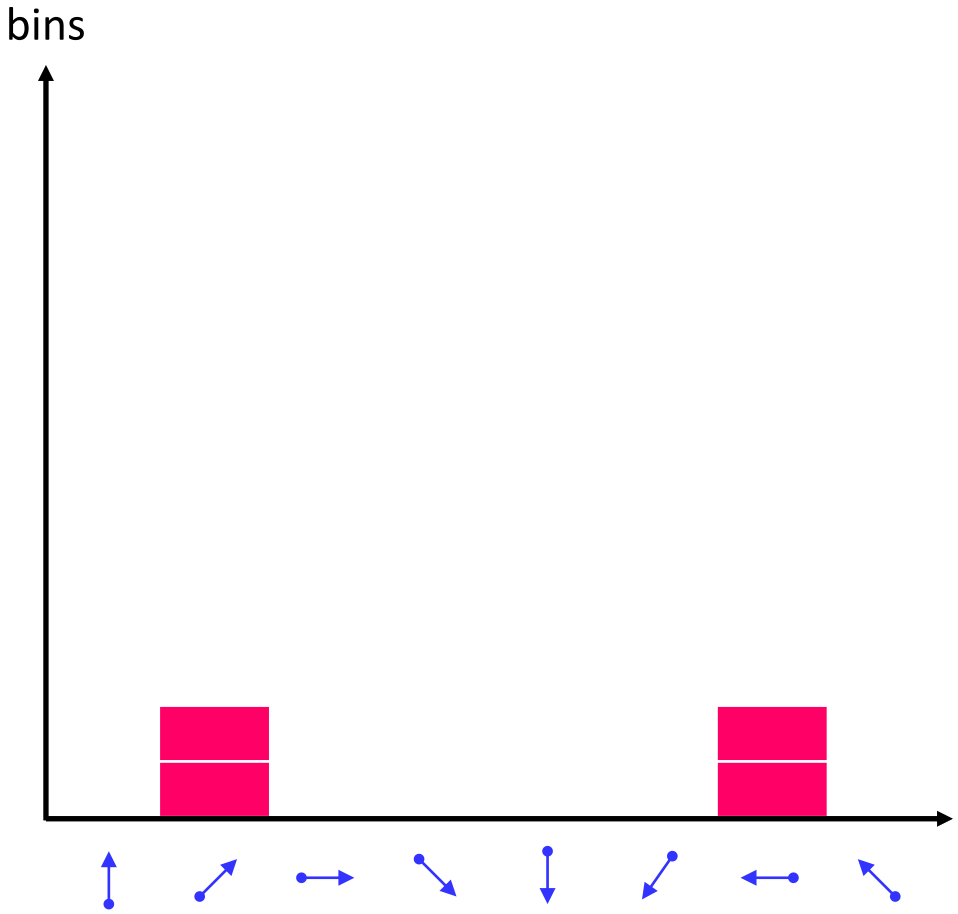


SIFT: **Orientation Assignment**

— **Gradient orientation:** Direction of arrows

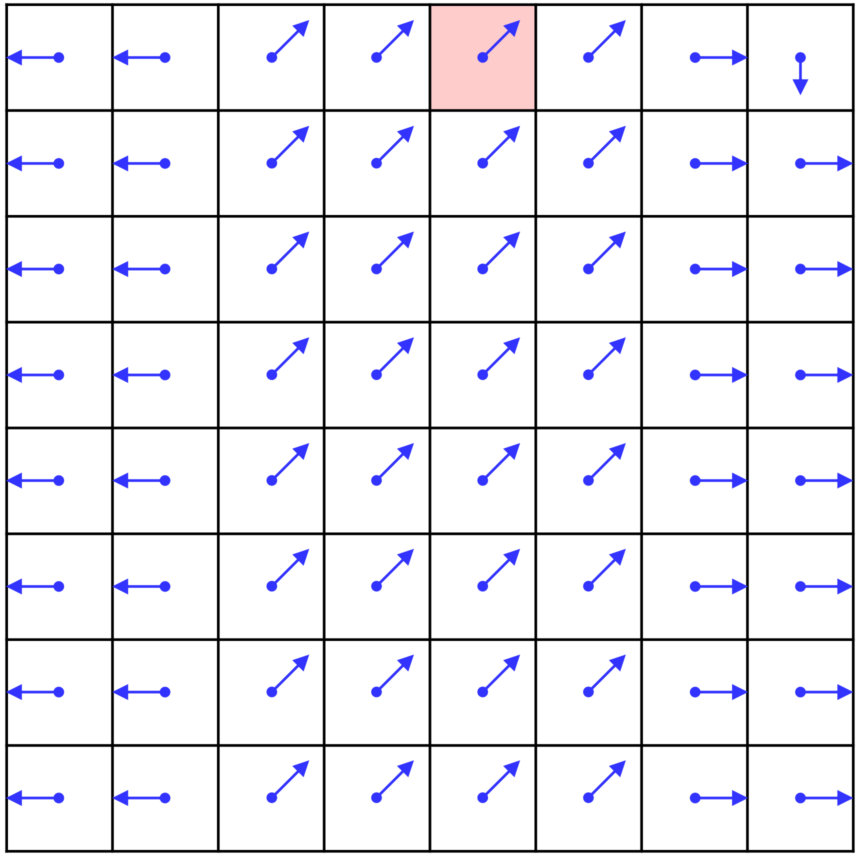
— **Gradient magnitude:** Length of arrows

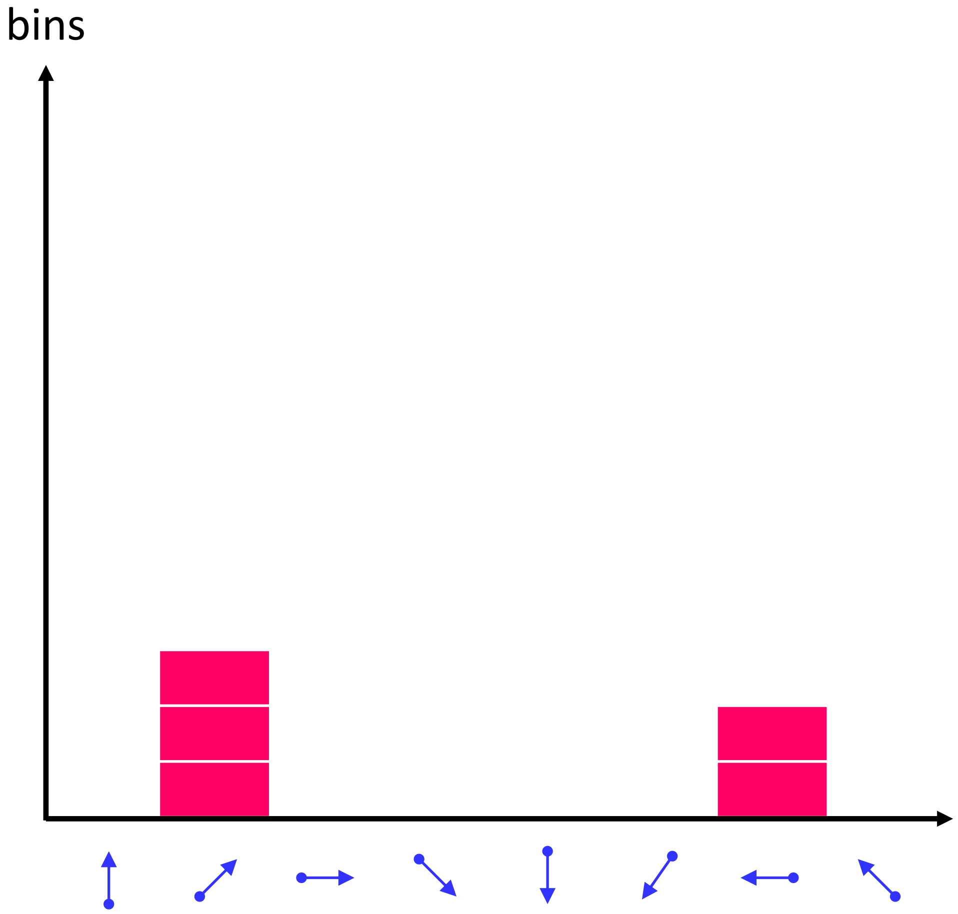




SIFT: **Orientation Assignment**

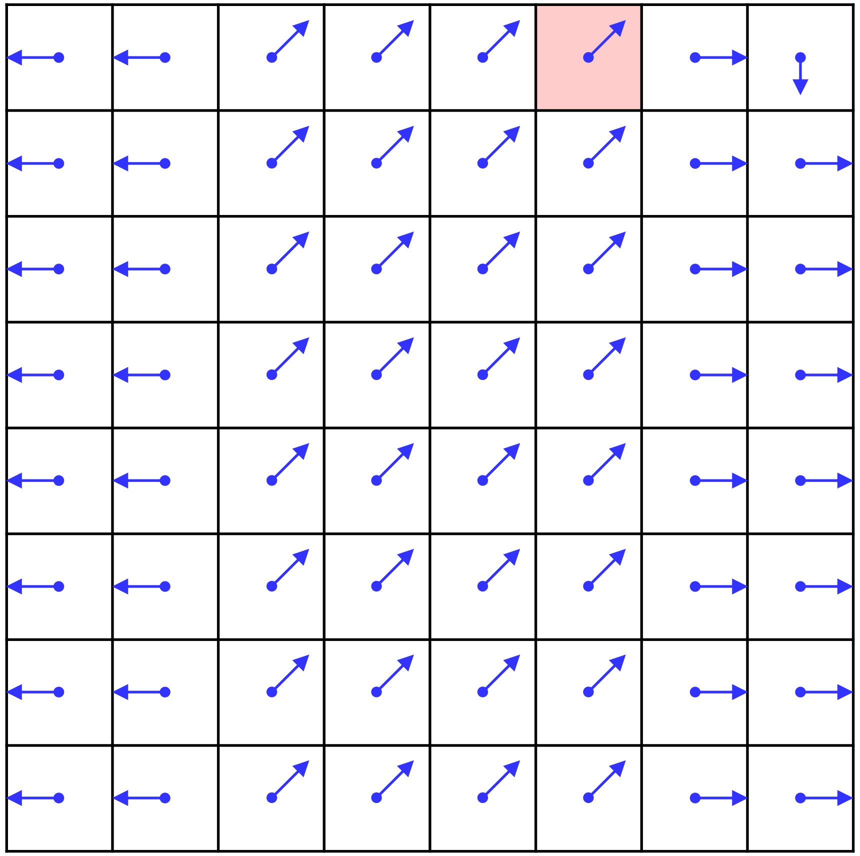
- **Gradient orientation:** Direction of arrows
- **Gradient magnitude:** Length of arrows

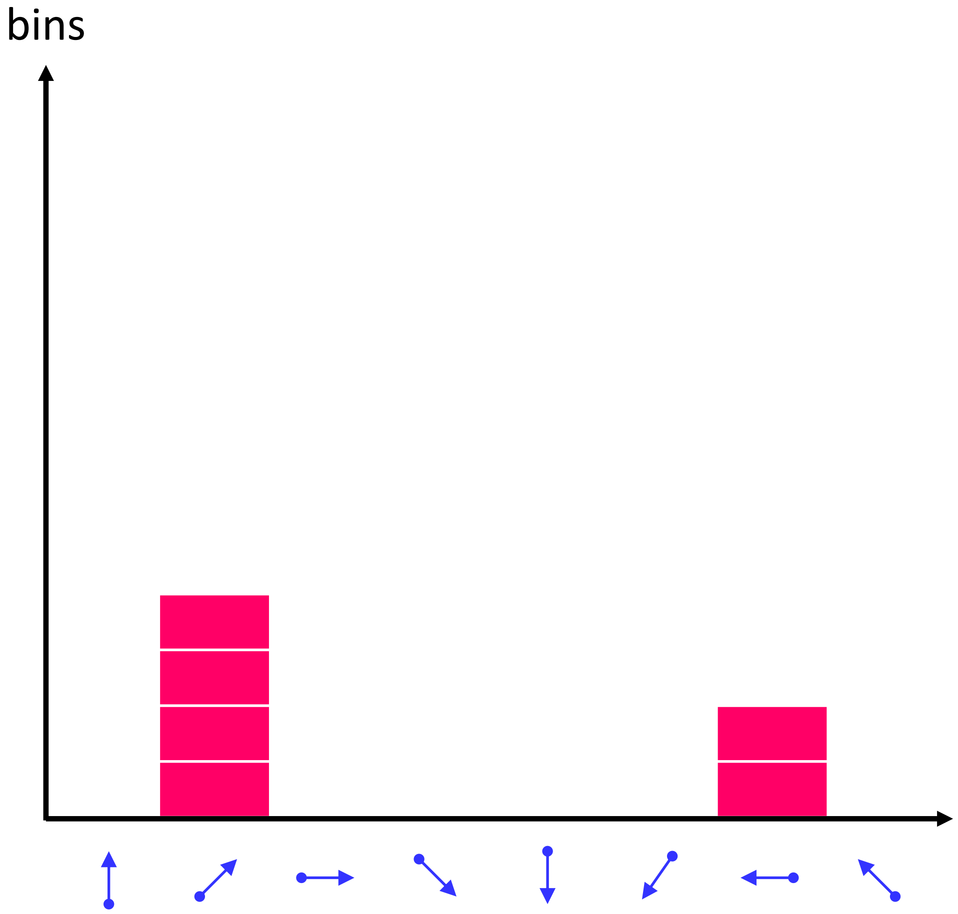




SIFT: **Orientation Assignment**

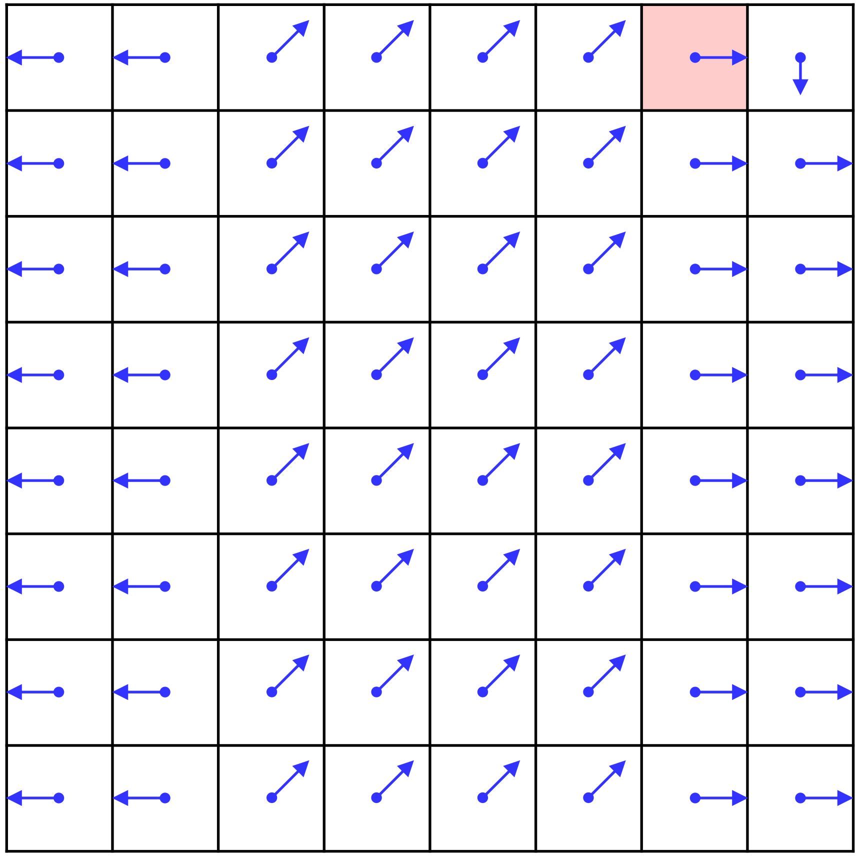
- **Gradient orientation:** Direction of arrows
- **Gradient magnitude:** Length of arrows

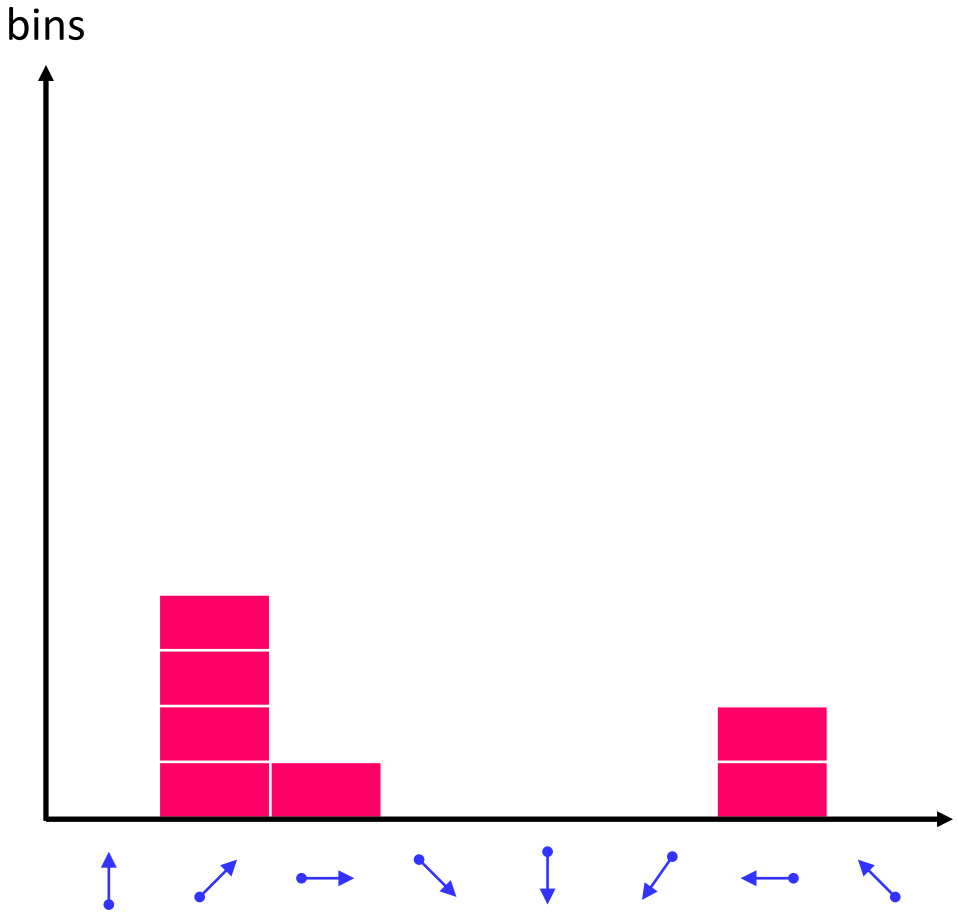




SIFT: **Orientation Assignment**

- **Gradient orientation:** Direction of arrows
- **Gradient magnitude:** Length of arrows

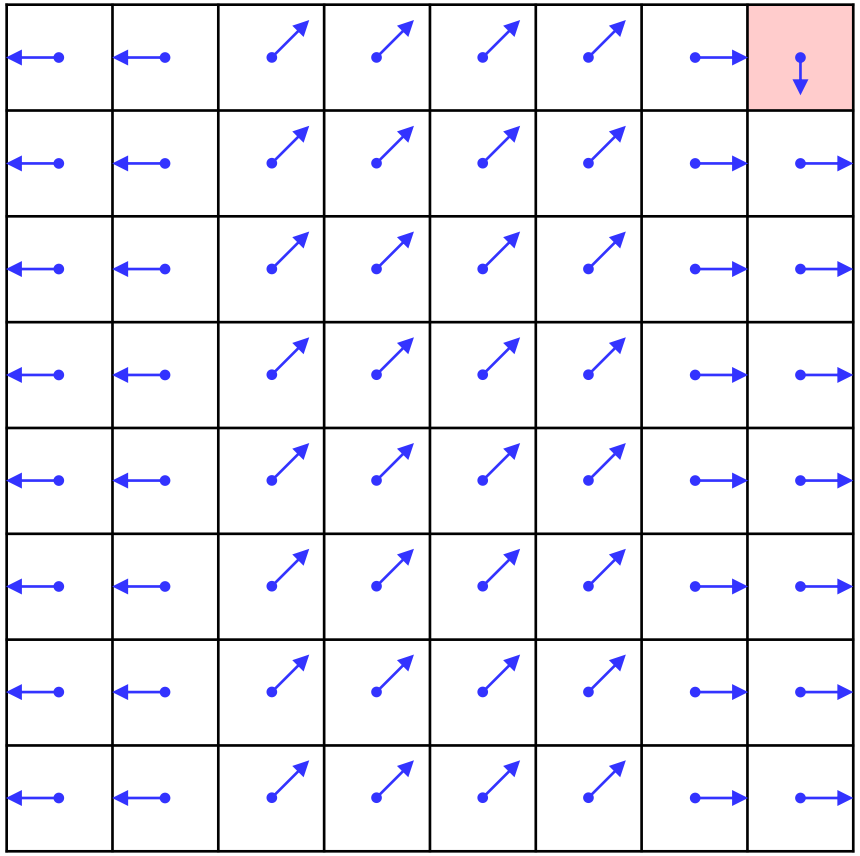


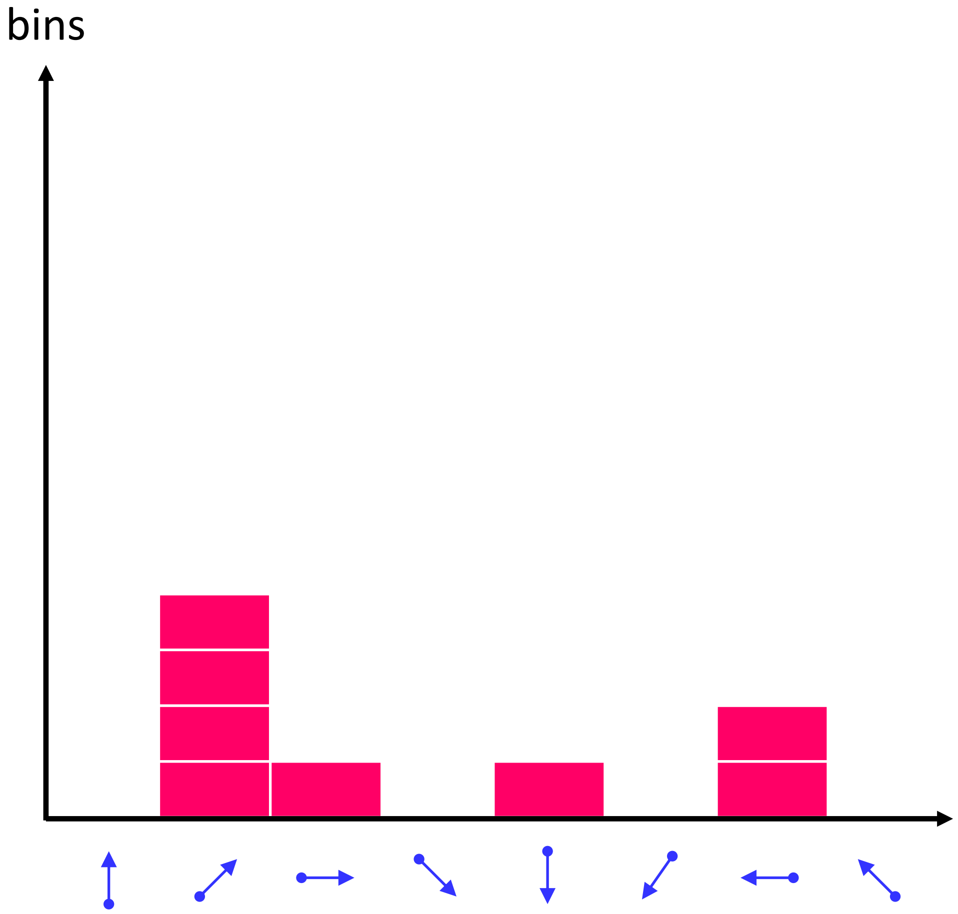


SIFT: **Orientation Assignment**

— **Gradient orientation:** Direction of arrows

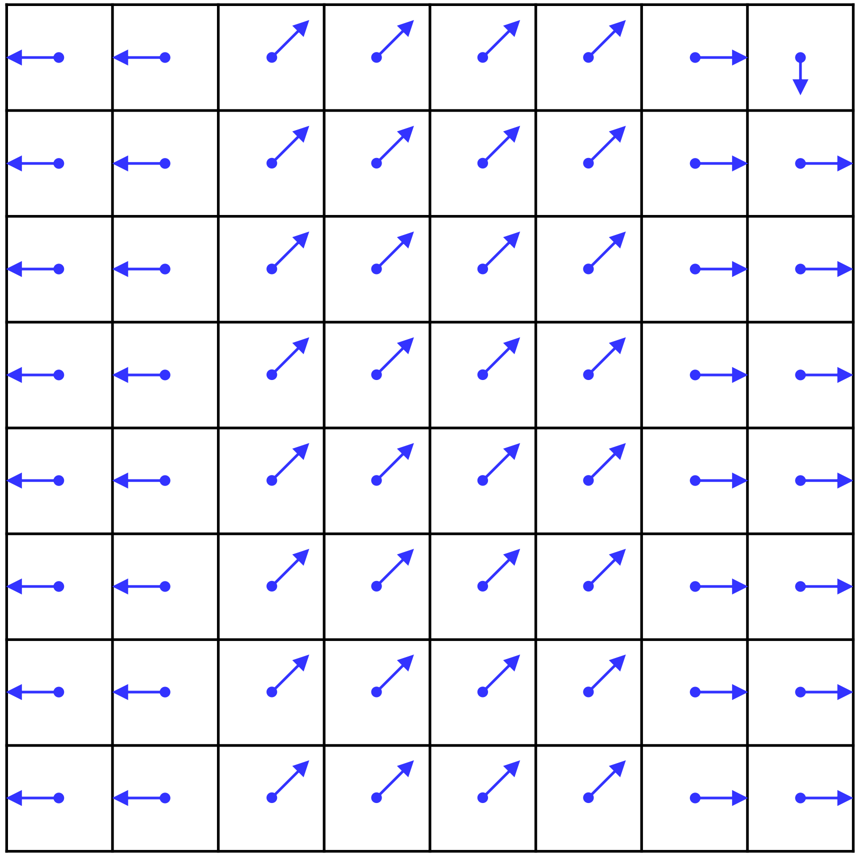
— **Gradient magnitude:** Length of arrows

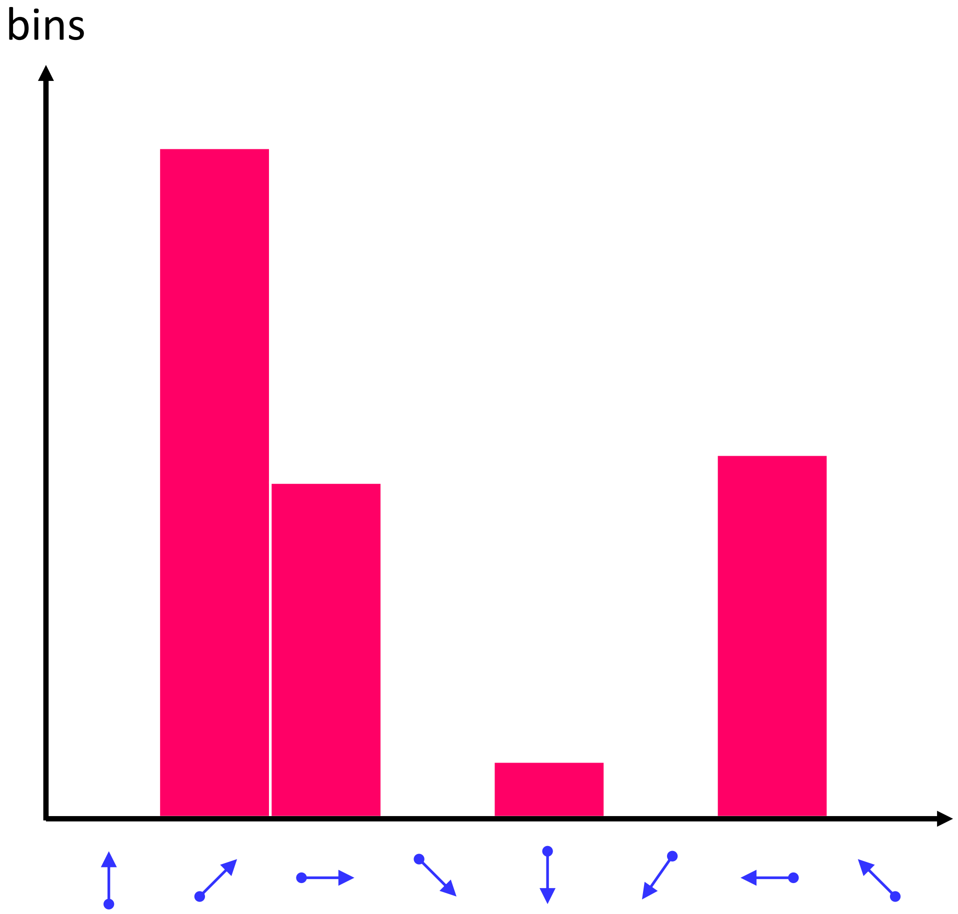




SIFT: **Orientation Assignment**

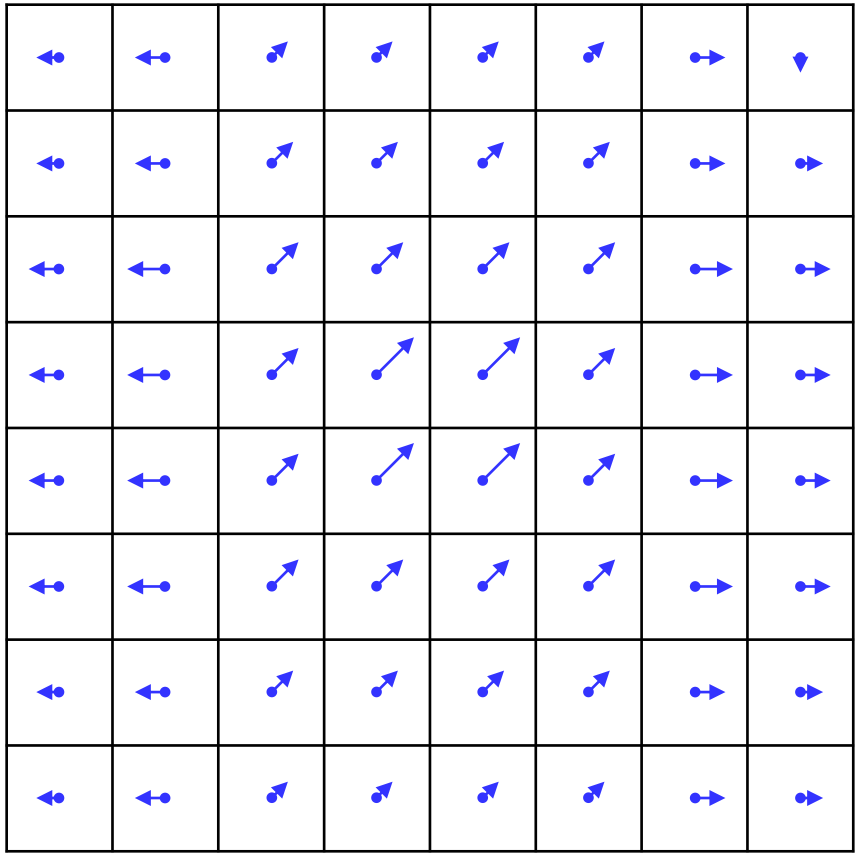
- **Gradient orientation:** Direction of arrows
- **Gradient magnitude:** Length of arrows

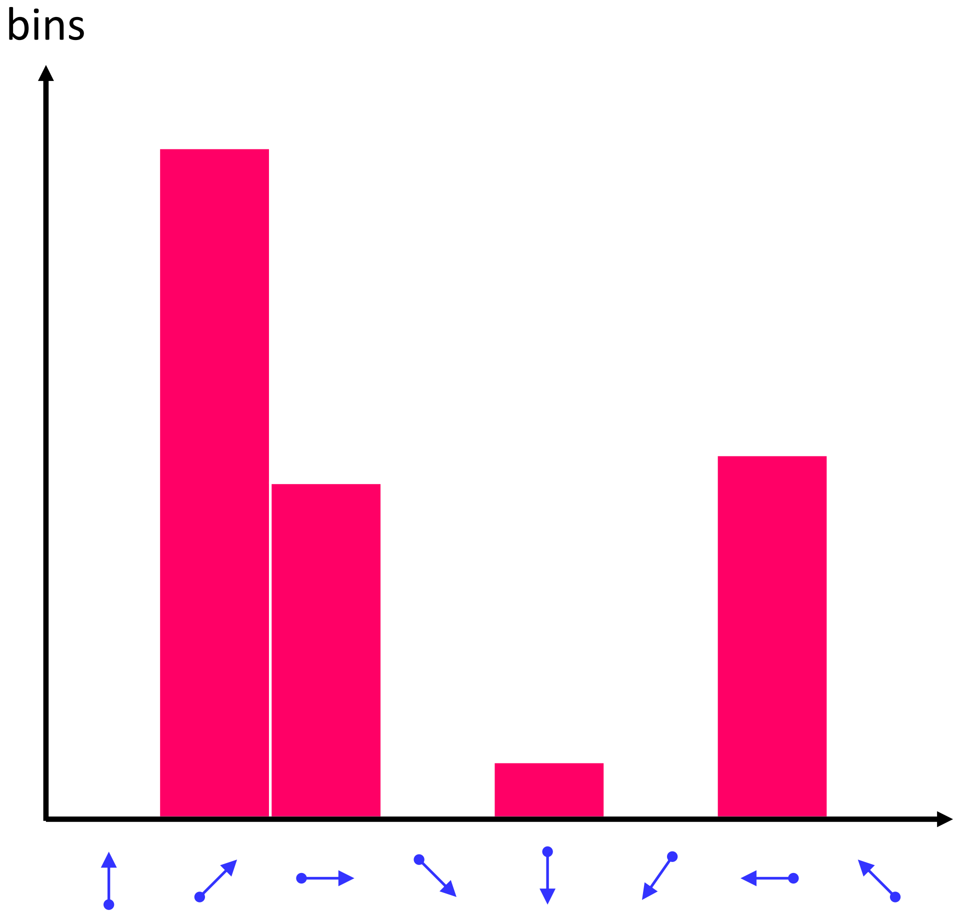




SIFT: **Orientation Assignment**

- Gradient magnitude **by a Gaussian kernel**





SIFT: **Orientation Assignment**

- Orientation Assignment method:

— Create **histogram of local gradient directions** computed at selected scale

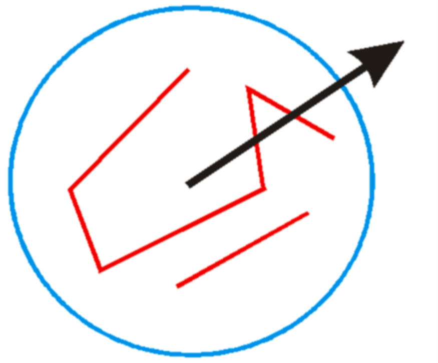
- Histogram of 36 bins (10 degree increments)

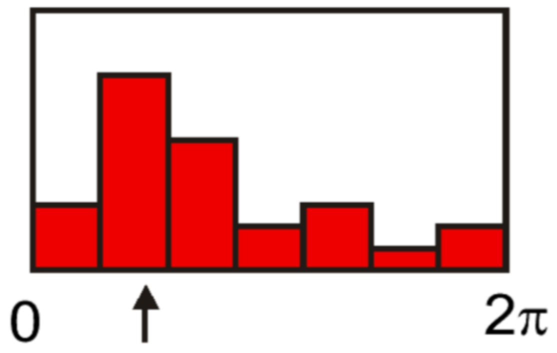
— Assign **canonical orientation** at peak of smoothed histogram

- Gaussian-weighted voting

— Each key specifies stable 2D coordinates: $(x, y, scale, orientation)$

- Highest peak and peaks above 80% of highest also considered for calculating dominant orientations





SIFT: **Keypoint Descriptor**

- **Keypoint detection**

— How to assign a location, scale and orientation to each keypoint

— This ensured invariance to image location, scale and rotation

- **Keypoint description**

— The descriptor is highly distinctive and partially invariant to the remaining variations such as illumination, 3D viewpoint, *etc*.

SIFT: **Keypoint Descriptor**

- Thresholded image gradients are sampled over 16 x 16 array of locations in scale space (weighted by a Gaussian with sigma half the size of the window)
- Create array of orientation histograms
- 8 Orientations x (4 x 4) histogram array

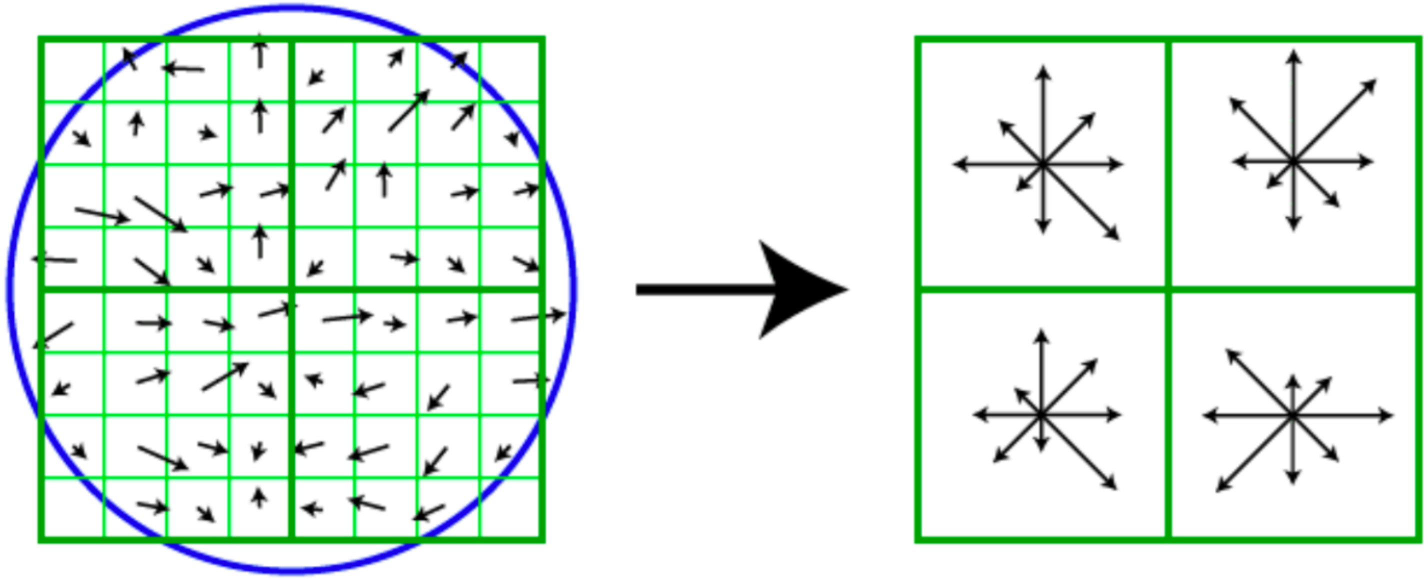


Image gradients

Keypoint descriptor

SIFT: **Keypoint Descriptor**

- Descriptor is normalized to unit length (i.e., magnitude of 1) to reduce the effects of illumination change

— If brightness values are scaled (multiplied) by a constant, the gradients are scaled by the same constant, and the normalization cancels the change

— If brightness values are increased/decreased by a constant, the gradients do not change

Summary: SIFT

① **Scale-space representation and local extreme detection**
- Use DoG/LoG Pyramid
- 3 scales/octave, down-sample by factor of 2 each octave

② **Keypoint localization**
- Select stable keypoints by thresholding on the magnitude of extrema and ratio of principal curvatures

③ **Keypoint orientation assignment**
- Based on histogram of local image gradient directions

④ **Keypoint descriptor**
- Histogram of local gradient directions – Vector with $8 \times (4 \times 4) = 128$ dimensions
- Vector normalization