

The SIFT (Scale Invariant Feature Transform) Detector and Descriptor

developed by David Lowe
University of British Columbia
Initial paper ICCV 1999
Newer journal paper IJCV 2004

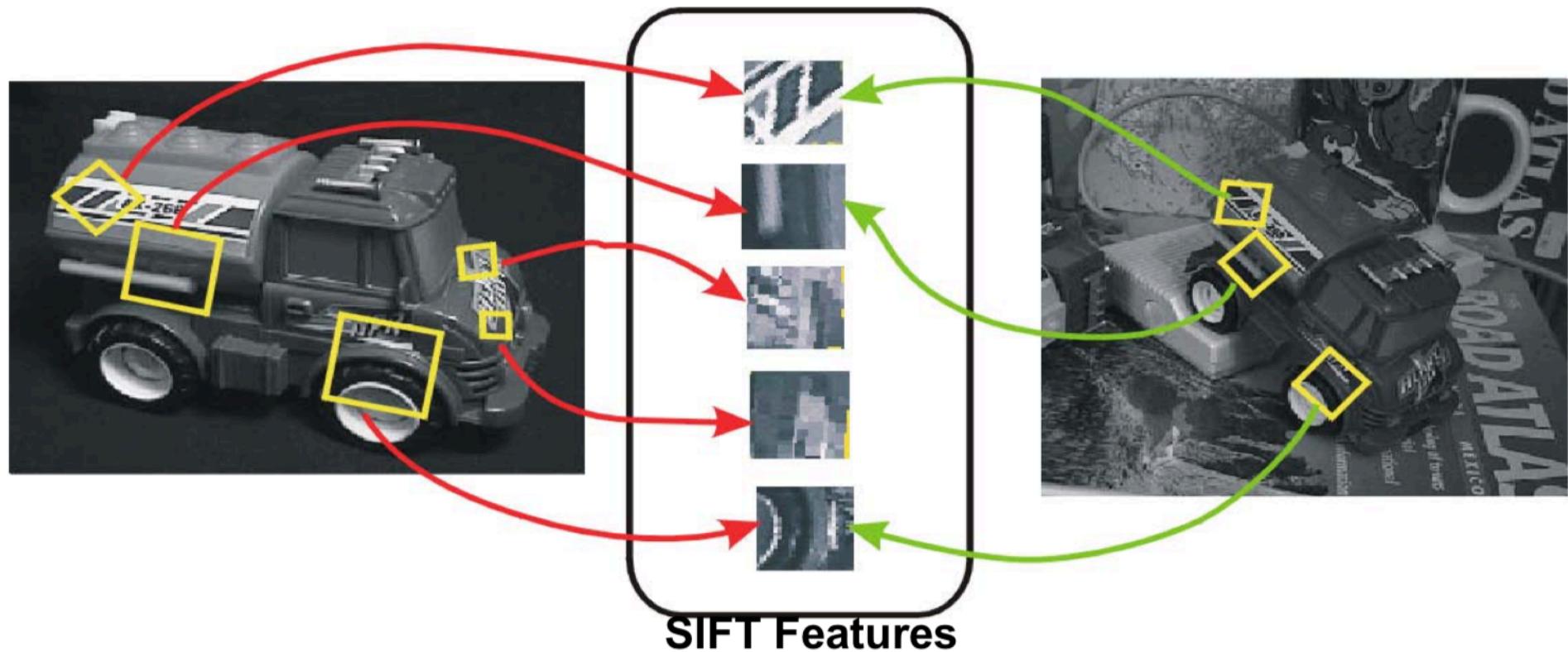
SIFT: Motivation

- The Harris operator is not invariant to scale and correlation is not invariant to rotation¹.
- For better image matching, Lowe's goal was to develop an interest operator that is invariant to scale and rotation.
- Also, Lowe aimed to create a **descriptor** that was robust to the variations corresponding to typical viewing conditions. **The descriptor is the most-used part of SIFT.**

¹But Schmid and Mohr developed a rotation invariant descriptor for it in 1997.

Idea of SIFT

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



Overall Procedure at a High Level

1. Scale-space extrema detection

Search over multiple scales and image locations.

2. Keypoint localization

Fit a model to determine location and scale.

Select keypoints based on a measure of stability.

3. Orientation assignment

Compute best orientation(s) for each keypoint region.

4. Keypoint description

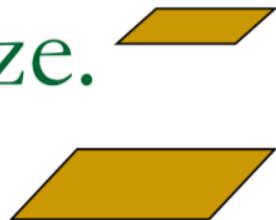
Use local image gradients at selected scale and rotation to describe each keypoint region.

1. Scale-space extrema detection

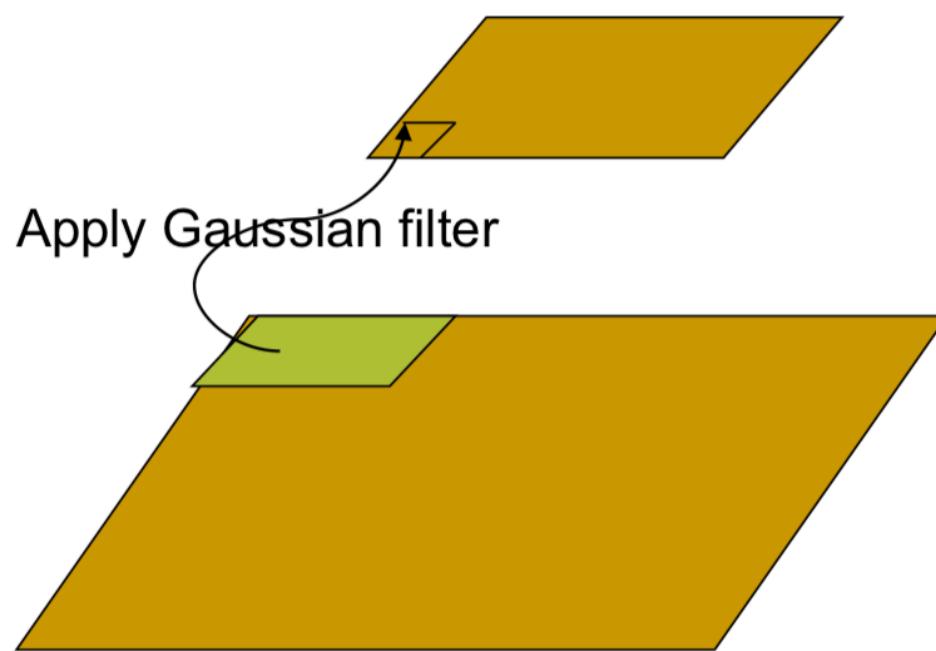
- **Goal:** Identify locations and scales that can be repeatably assigned under different views of the same scene or object.
- **Method:** search for stable features across multiple scales using a continuous function of scale.
- **Prior work** has shown that under a variety of assumptions, the best function is a **Gaussian function**.
- The scale space of an image is a function $L(x,y,\sigma)$ that is produced from the convolution of a Gaussian kernel (at different scales) with the input image.

Aside: Gaussian Pyramid

At each level, image is smoothed and reduced in size.



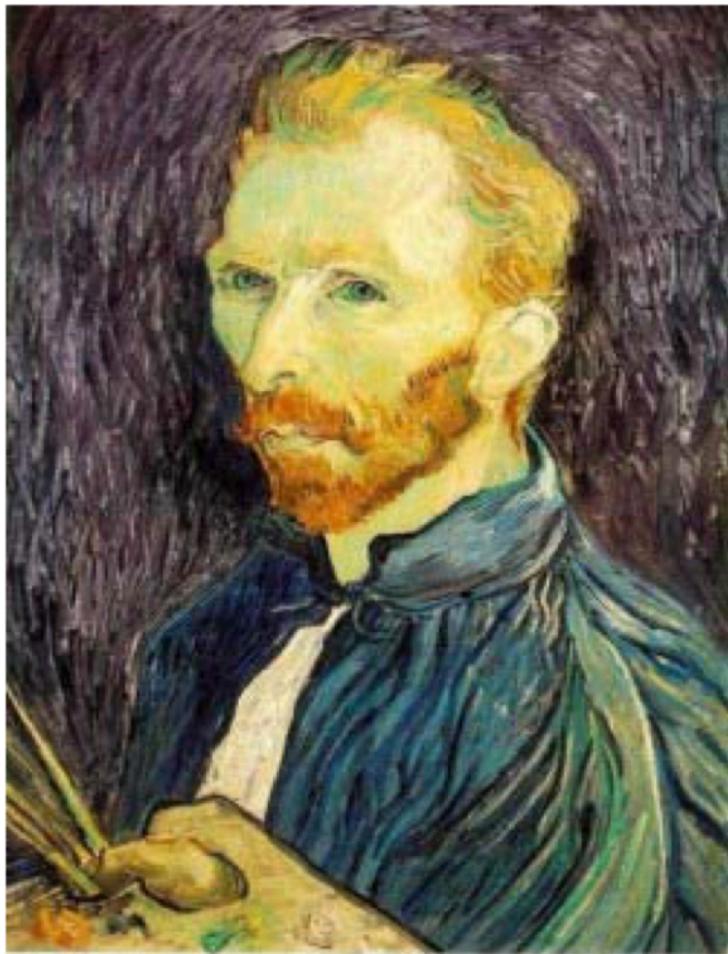
And so on.



At 2nd level, each pixel is the result of applying a Gaussian mask to the first level and then subsampling to reduce the size.

Bottom level is the original image.

Example: Subsampling with Gaussian pre-filtering



Gaussian 1/2



G 1/4



G 1/8

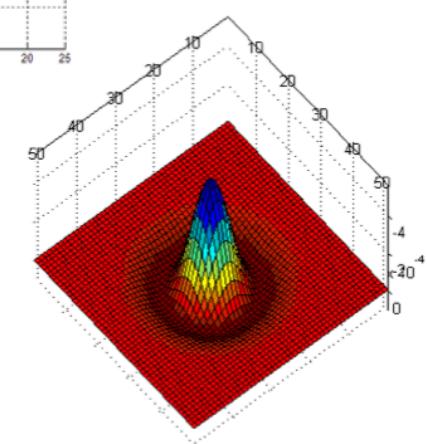
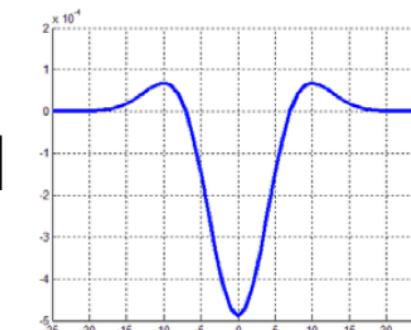
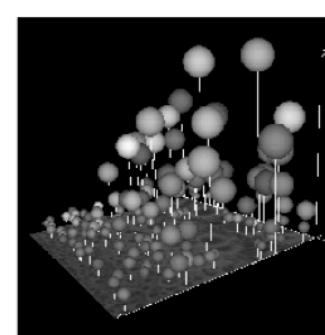
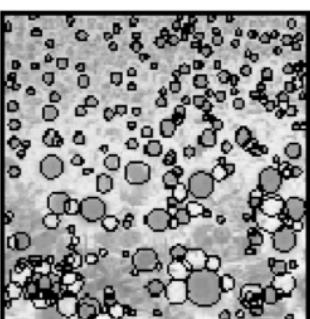
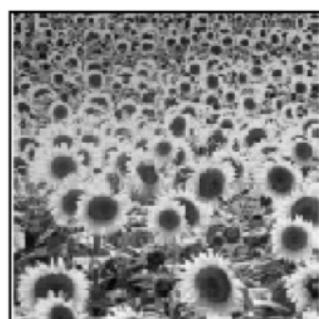
Lowe's Scale-space Interest Points

■ Laplacian of Gaussian kernel

- Scale normalised (x by scale 2)
- Proposed by Lindeberg

■ Scale-space detection

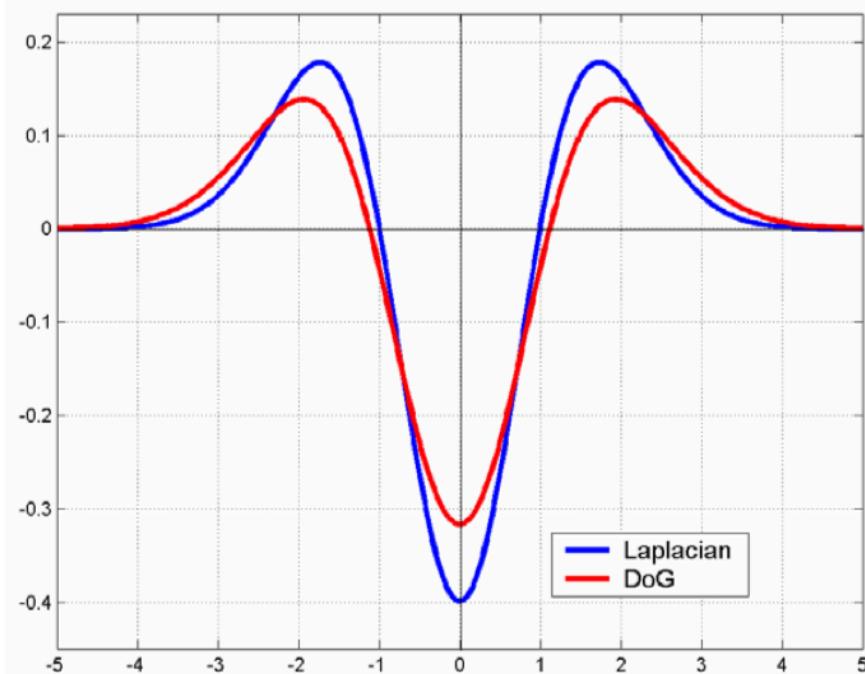
- Find local maxima across scale/space
- A good “blob” detector



$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2} \frac{x^2+y^2}{\sigma^2}}$$

$$\nabla^2 G(x, y, \sigma) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2}$$

Lowe's Scale-space Interest Points: Difference of Gaussians



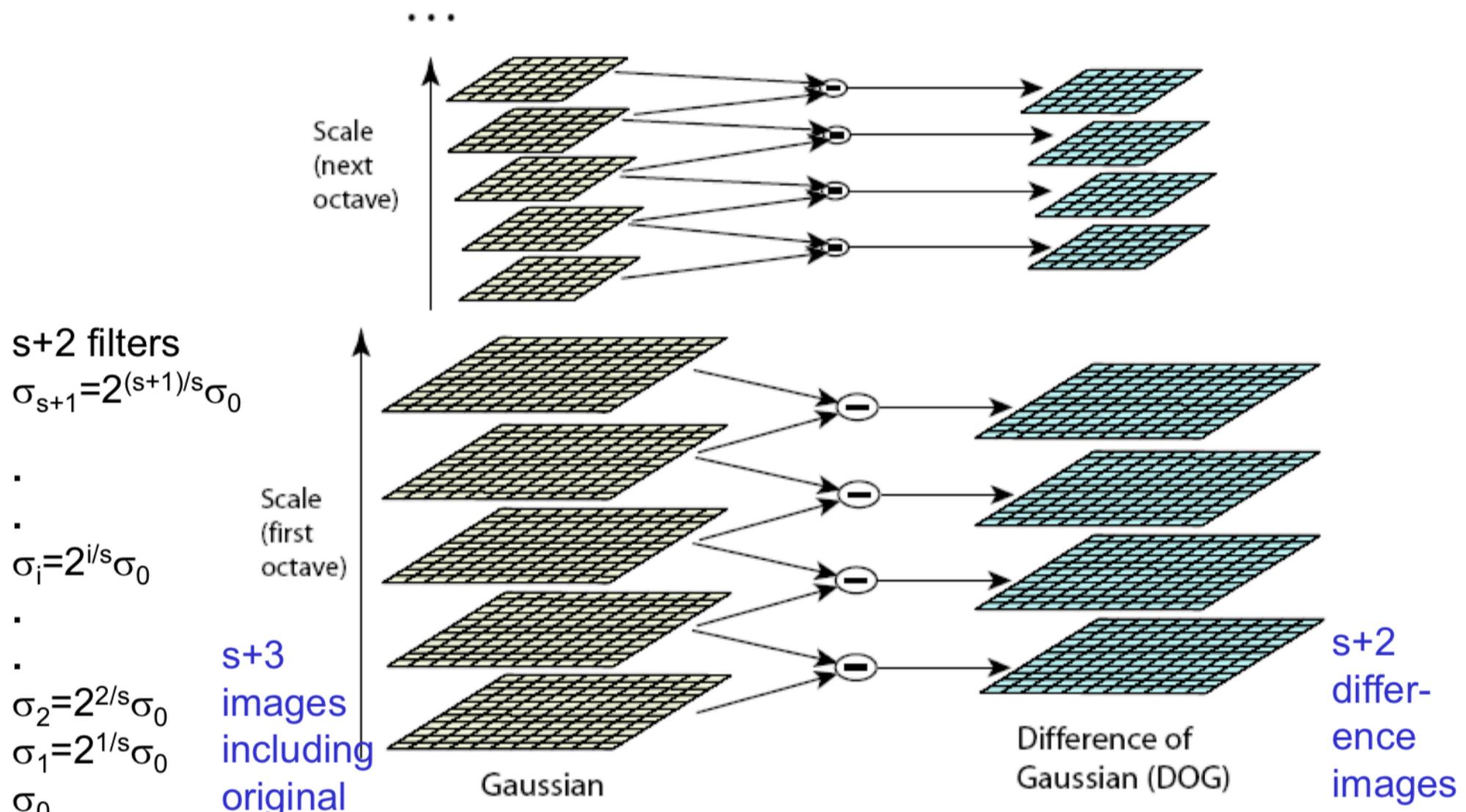
- Gaussian is an ad hoc solution of heat diffusion equation
- Hence
- $G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G.$
- k is not necessarily very small in practice

①
DoG = $\frac{1}{\sqrt{\pi}} G_i - G_{i+1}$
 G_i Gauss
key points

Lowe's Pyramid Scheme

- Scale space is separated into **octaves**:
 - Octave 1 uses scale σ
 - Octave 2 uses scale 2σ
 - etc.
- In each octave, the initial image is repeatedly convolved with Gaussians to produce a set of scale space images.
- Adjacent Gaussians are subtracted to produce the DOG
- After each octave, the Gaussian image is down-sampled by a factor of 2 to produce an image $\frac{1}{4}$ the size to start the next level.

Lowe's Pyramid Scheme



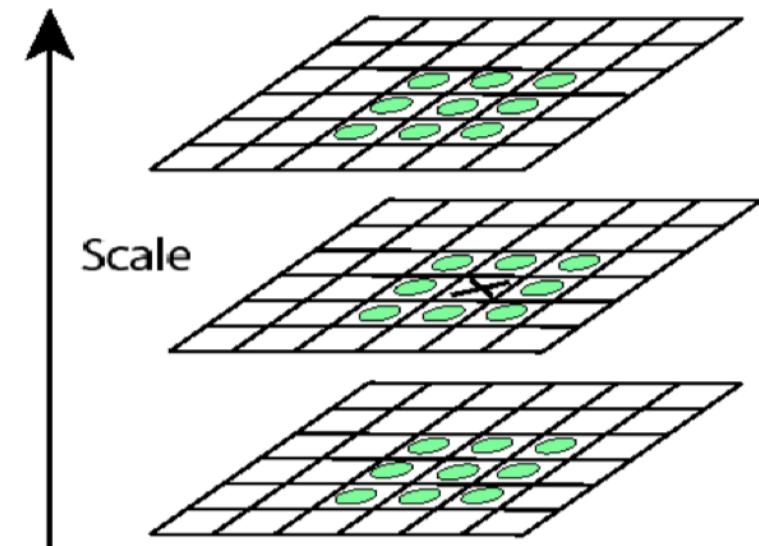
The parameter **s** determines the number of images per octave.

Key point localization

如果这个点是 max/min
pixel { it is max/min by

s+2 difference images.
top and bottom ignored.
s planes searched.

- Detect maxima and minima of difference-of-Gaussian in scale space
- Each point is compared to its 8 neighbors in the current image and 9 neighbors each in the scales above and below

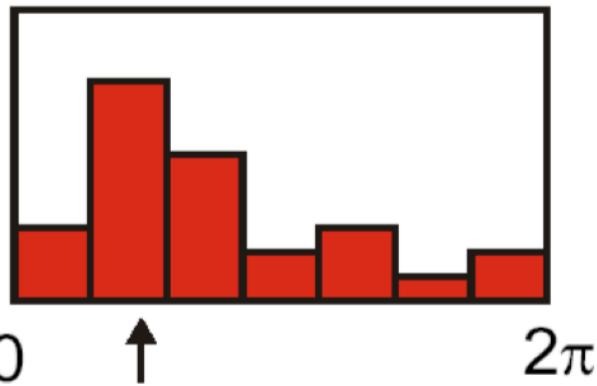
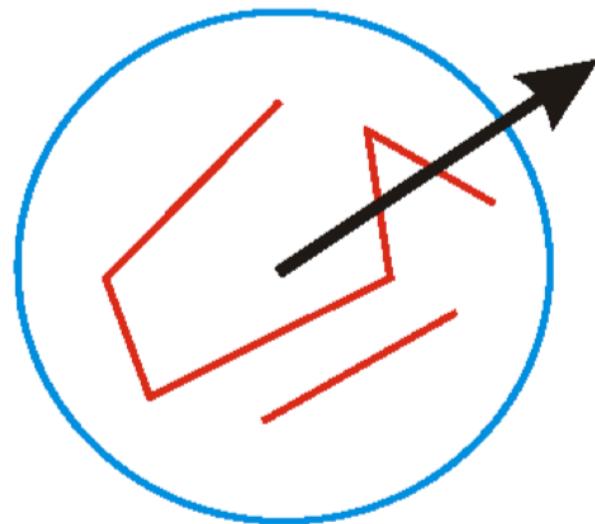


For each max or min found,
output is the **location** and
the **scale**.

Keypoint localization

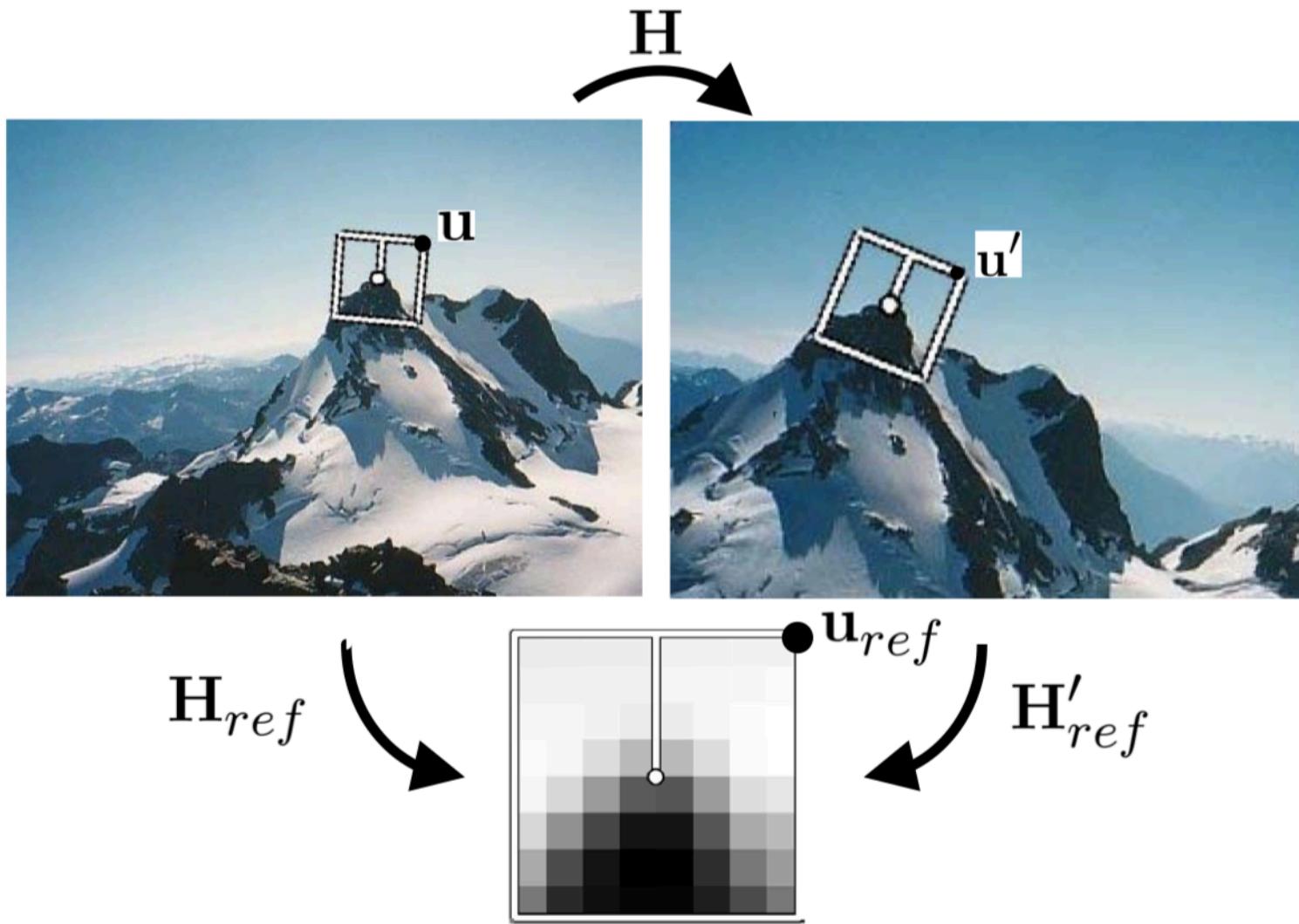
- Once a keypoint candidate is found, perform a detailed fit to nearby data to determine
 - location, scale, and ratio of principal curvatures
- In initial work keypoints were found at location and scale of a central sample point.
- In newer work, they fit a 3D quadratic function to improve interpolation accuracy.
- The Hessian matrix was used to eliminate edge responses.

3. Orientation assignment

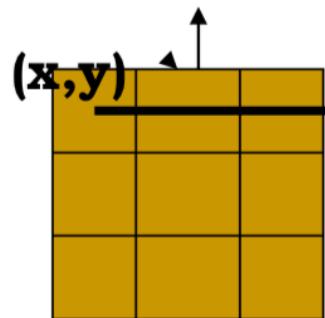


- Create histogram of local gradient directions at selected scale
- Assign canonical orientation at peak of smoothed histogram
- Each key specifies stable 2D coordinates ($x, y, \text{scale}, \text{orientation}$)

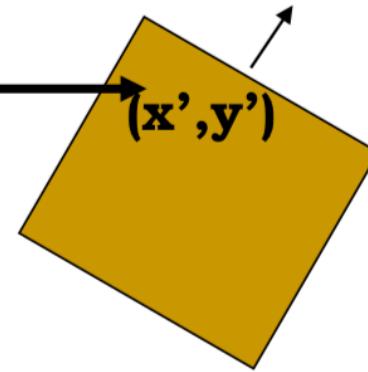
If 2 major orientations, use both.



Rotating a Patch



empty canonical patch



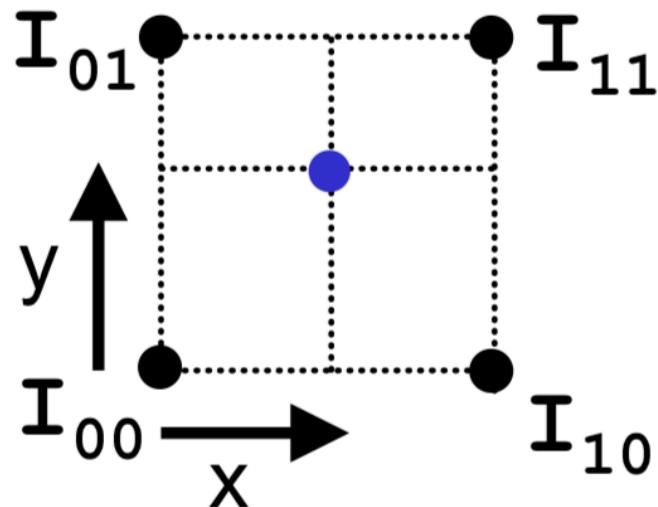
patch detected in the image

$$T \quad \begin{aligned} x' &= x \cos\theta - y \sin\theta \\ y' &= x \sin\theta + y \cos\theta \end{aligned}$$

counterclockwise rotation

Using Bilinear Interpolation

- Use all 4 adjacent samples



Keypoint localization with orientation

233x189



729

keypoints after
gradient threshold



832

initial keypoints



536

keypoints after
ratio threshold



4. Keypoint Descriptors

- At this point, each keypoint has
 - location
 - scale
 - orientation
- Next is to compute a descriptor for the local image region about each keypoint that is
 - highly distinctive
 - invariant as possible to variations such as changes in viewpoint and illumination

SIFT Descriptor

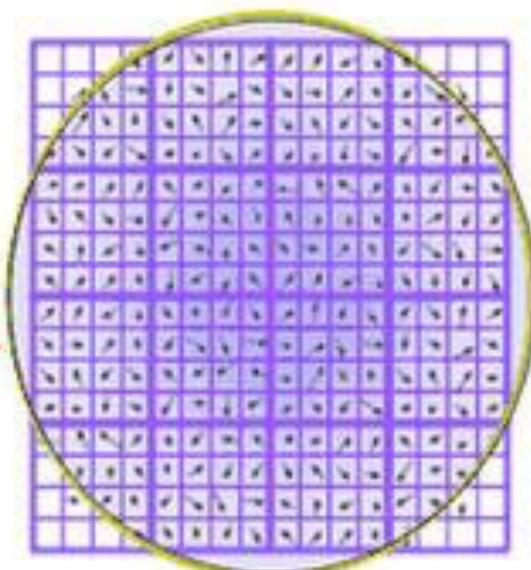
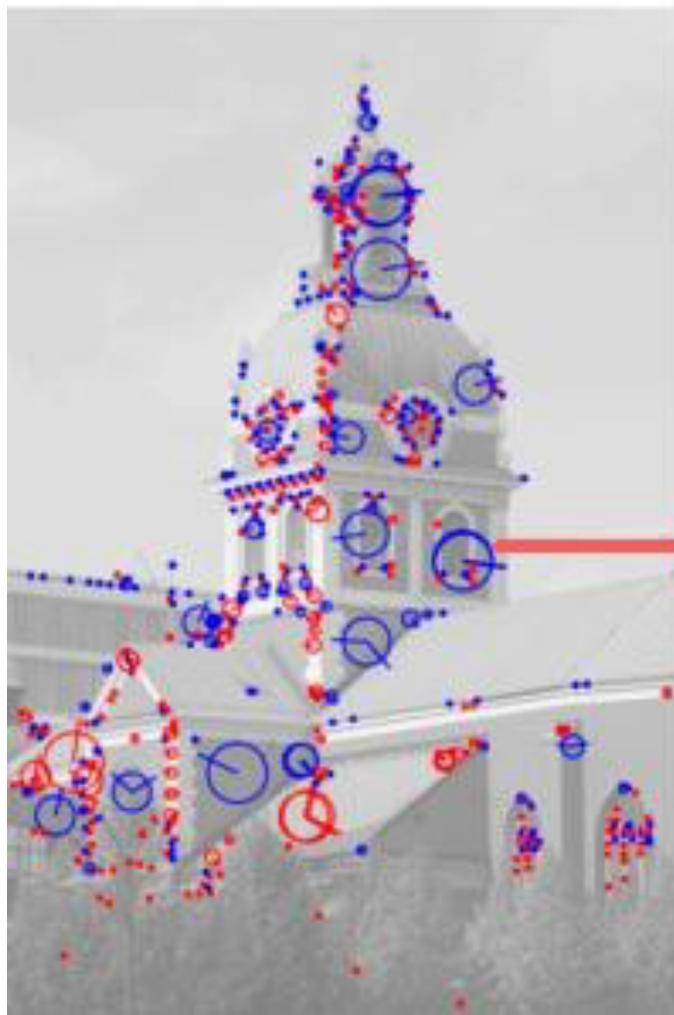
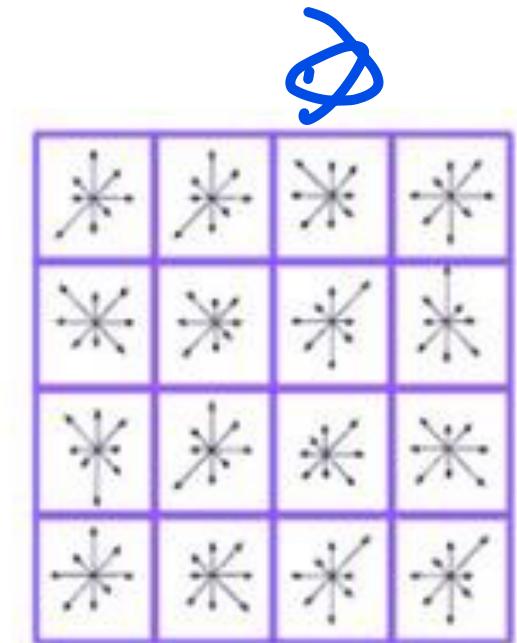


Image gradients



Keypoint descriptor

找到 keypoints
提取特征

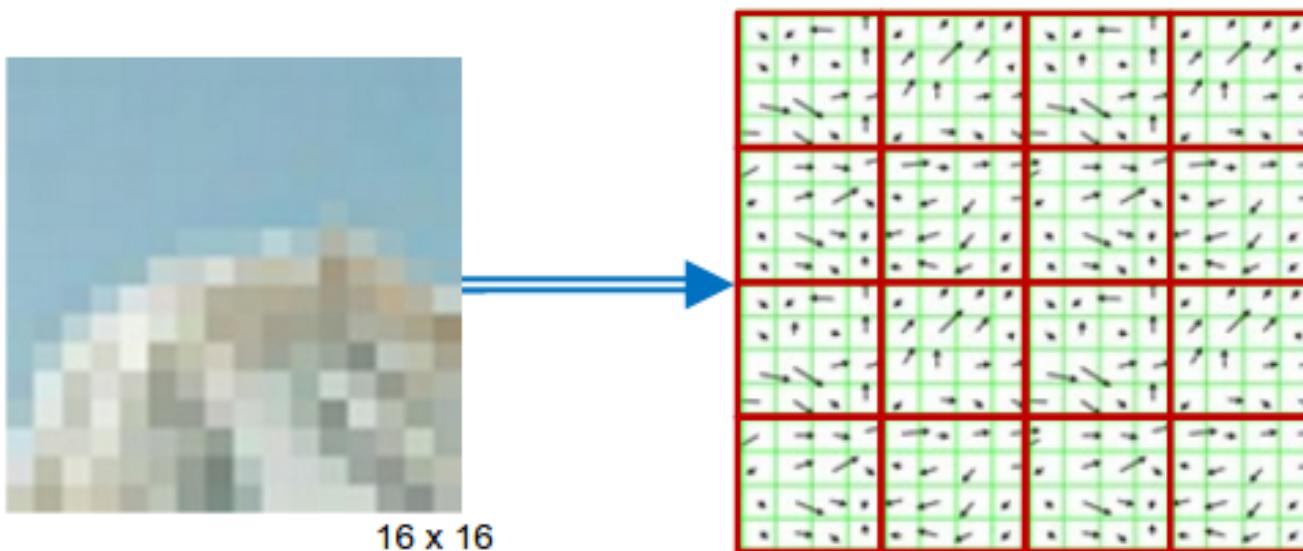
How to compute SIFT descriptor

Input: an image and a location to compute the descriptor



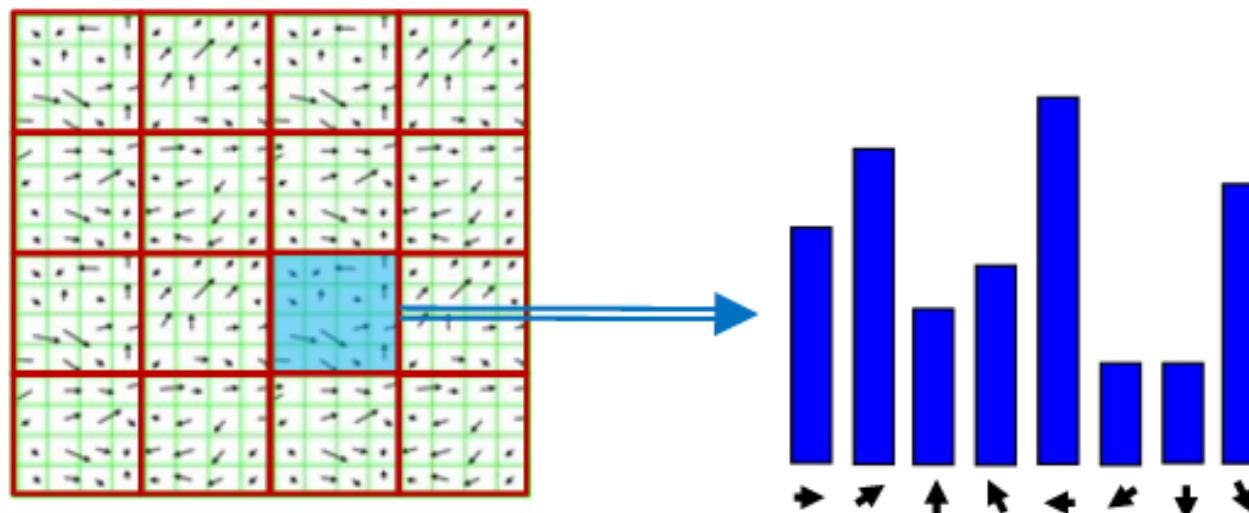
Step 1: Warp the image to the correct orientation and scale, and than extract the feature as 16×16 pixels

Step 2: Compute the gradient for each pixel (direction and magnitude)



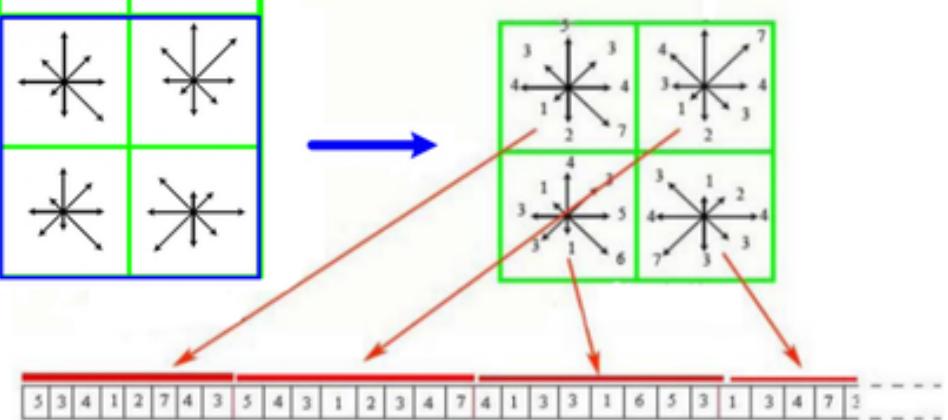
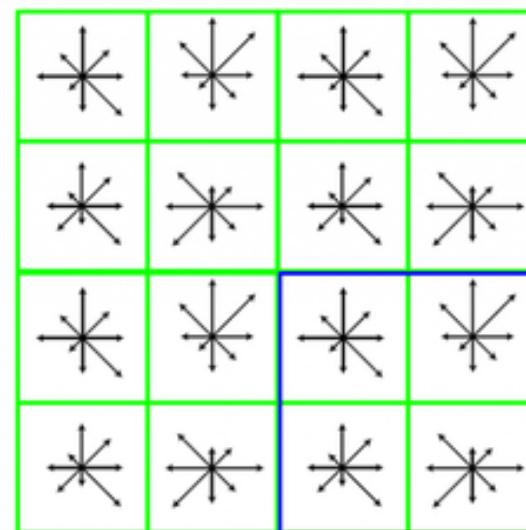
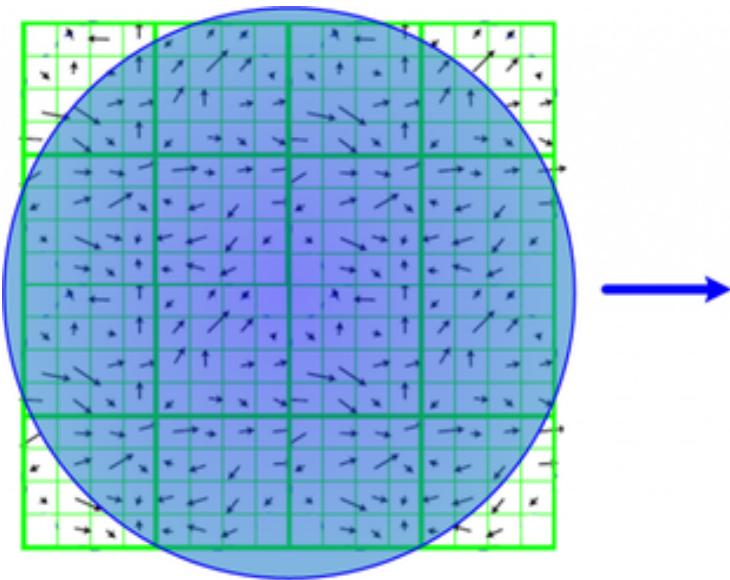
Step 3: Divide the pixels into 16, 4x4 squares

Step 4: For each square, compute gradient direction histogram over 8 directions.



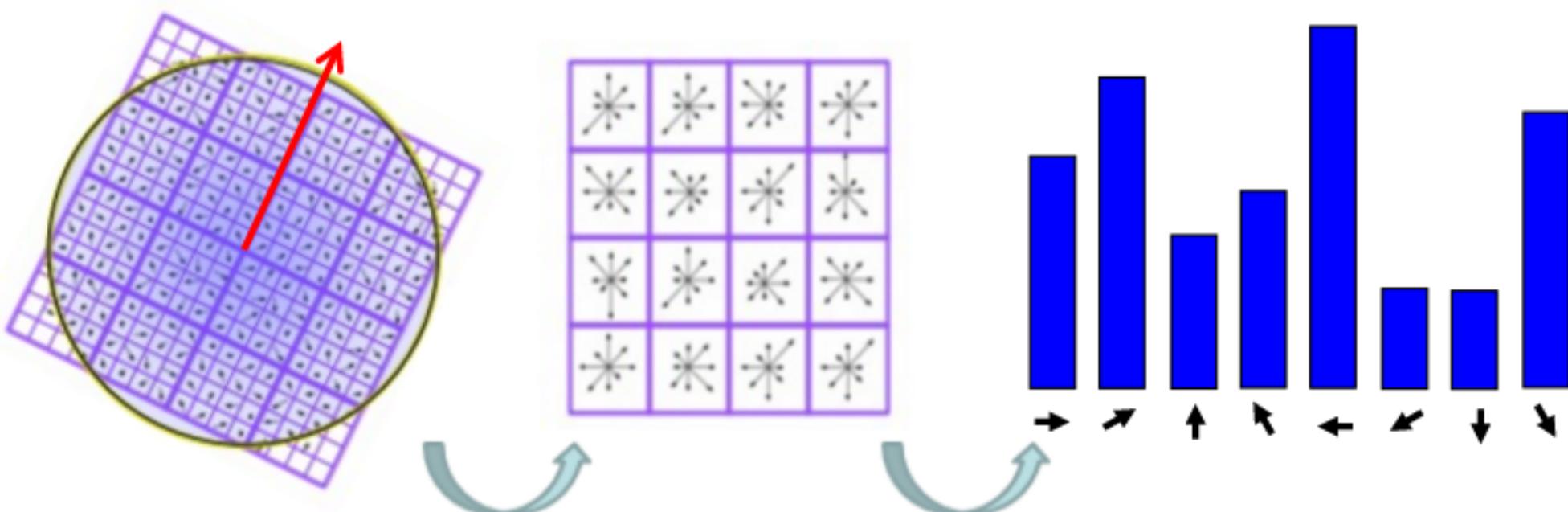
The result: 128 dimensions feature vector.





Recap

- Warp the feature into 16x16 square.
- Divide into 16, 4x4 squares.
- For each square, compute an histogram of the gradient directions.

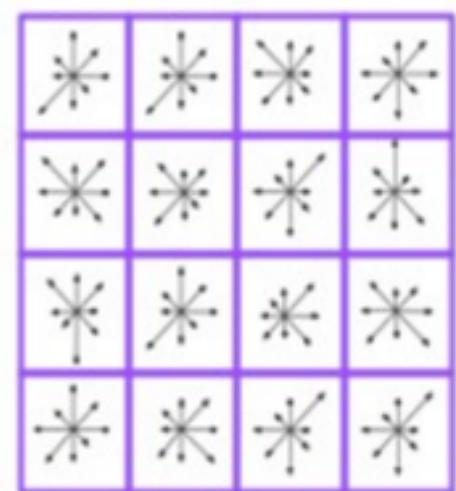


Invariance to illumination

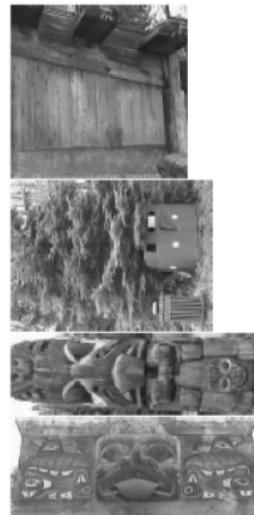
- Gradient are invariant to Light intensity shift (i.e. add a scalar to all the pixels)
- Normalization to unit length add invariance to light intensity change (i.e. multiply all the pixels by a scalar)

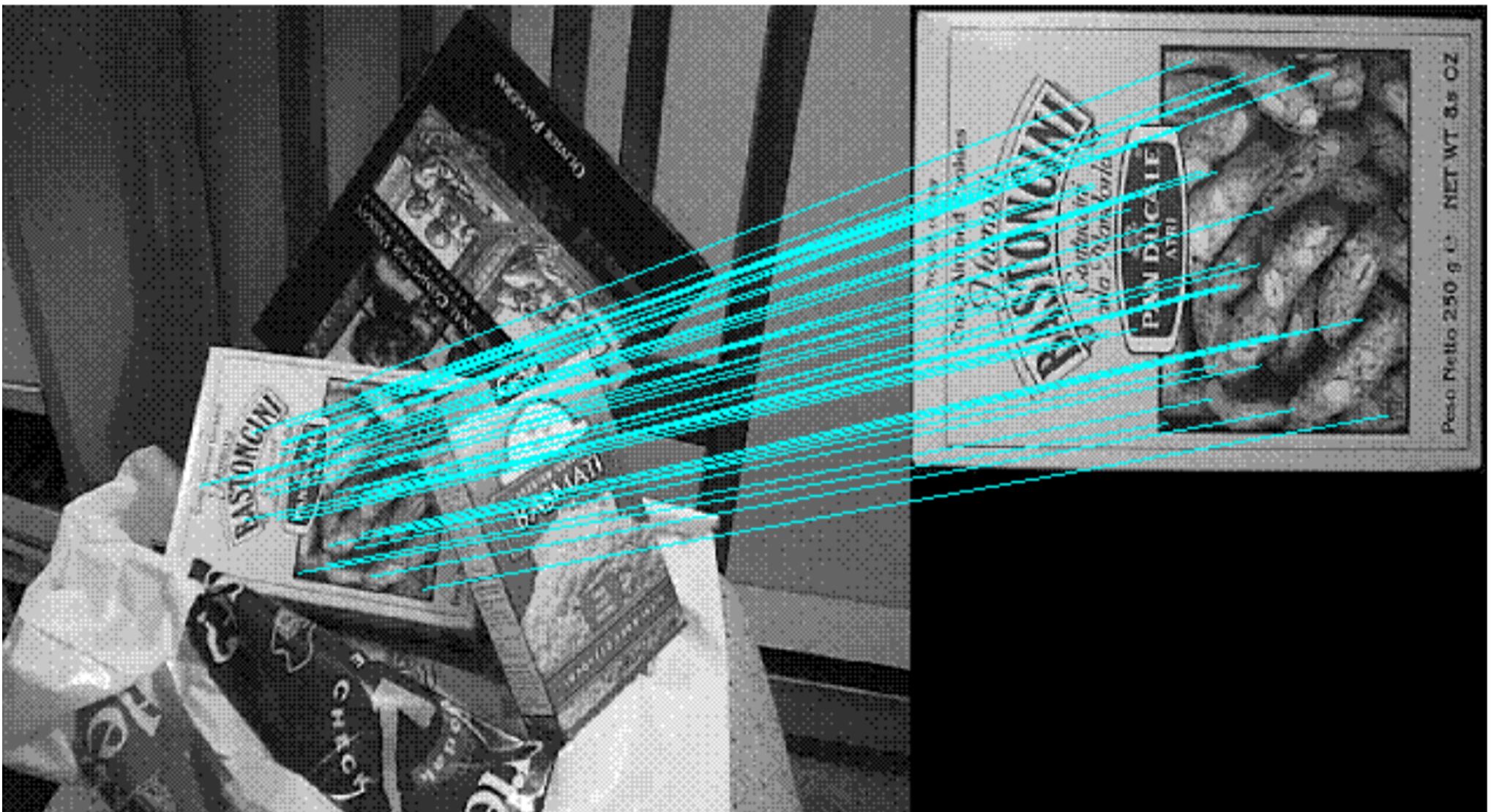
Invariance to shift and rotation

- Histograms does not contains any geometric information
- Using 16 histograms allows to preserve geometric information.



Using SIFT for Matching “Objects”





Peso Netto 250 g • NET WT 8.8 OZ

Uses for SIFT

- Feature points are used also for:
 - Image alignment (homography, fundamental matrix)
 - 3D reconstruction (e.g. Photo Tourism)
 - Motion tracking
 - Object recognition
 - Indexing and database retrieval
 - Robot navigation
 - ... many others

