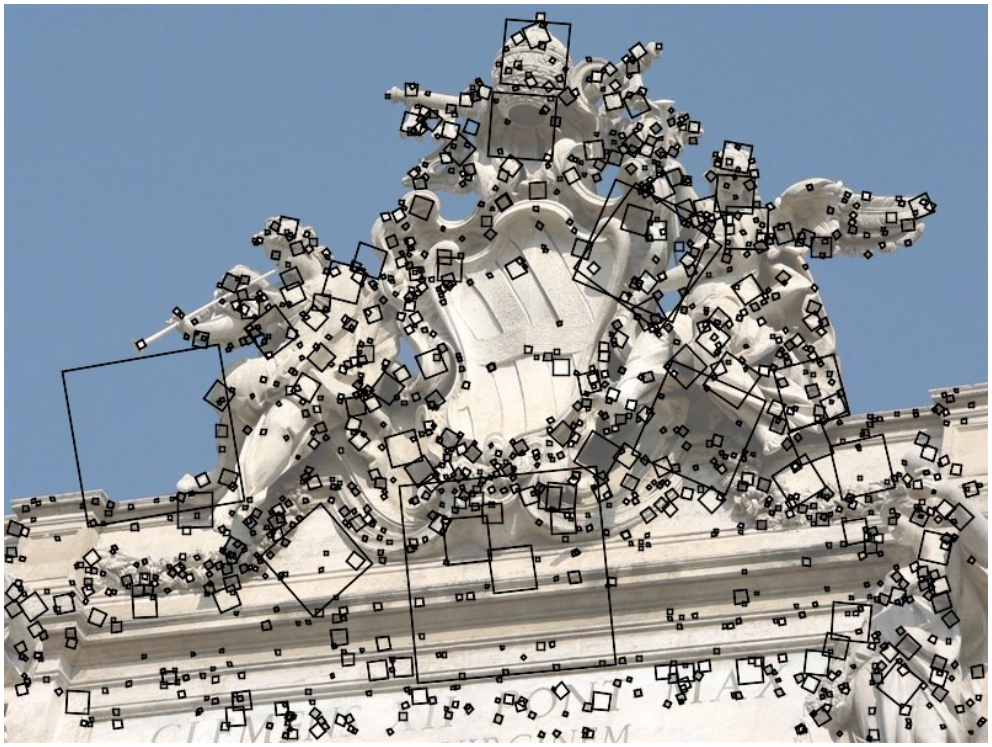
SIFT keypoint detection



D. Lowe, Distinctive image features from scale-invariant keypoints,
IJCV 60 (2), pp. 91-110, 2004

Overview

- Windows
  - Location and scale
- Laplacian of Gaussian filter
- Orientation
- SIFT descriptors
- Learning to find interest points and their descriptions

What we want

We want to find patches that are “worth representing”

- to match from image to image
- to represent textures
- to represent objects

And describe them in a distinctive way

What we have

- Corner detection
  - A corner can be localized
    - It is covariant to translation, rotation
      - Tolerable behavior under scale (this can be fixed, but we won’t)

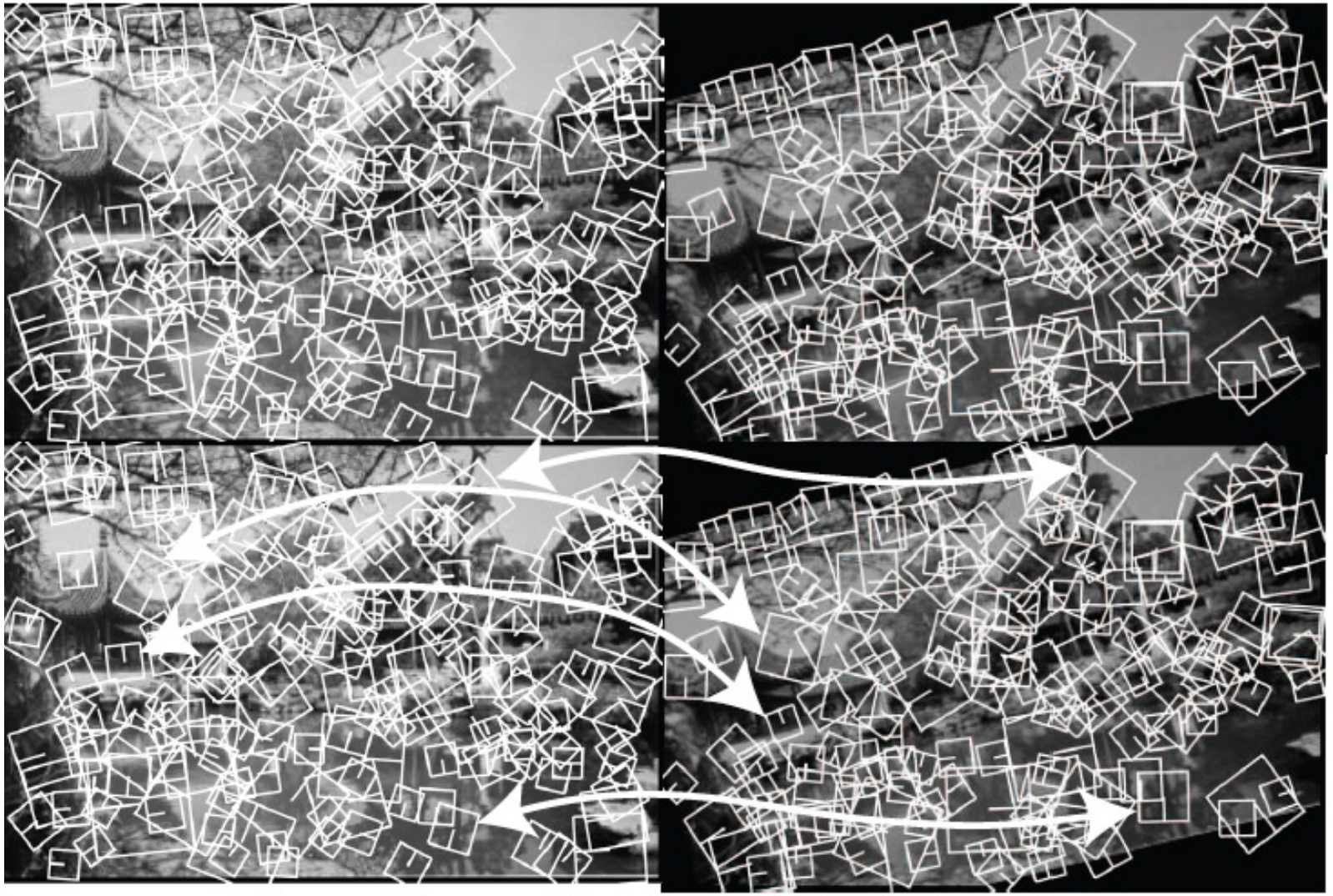
What we need

Build a window around the corner

- Covariant to translation, rotation, scale
  - i.e. if the image is translated, rotated, scaled, so are the neighborhoods
  - important to ensure that the representation of the patch is stable
- Localizable in translation, rotation, scale
  - we can estimate the position, orientation and size of the patch
  - and get the answer about right

Describe that window

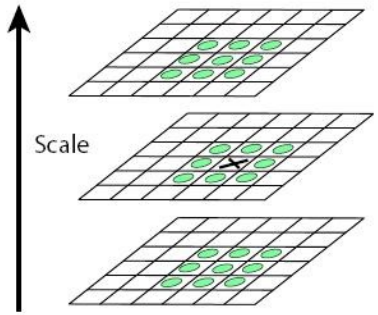
(methods exist for richer sets of requirements, but…)



Corresponding neighborhoods are scaled and rotated appropriately. Arrows identify matches; look for others on your own.

Scale of a window

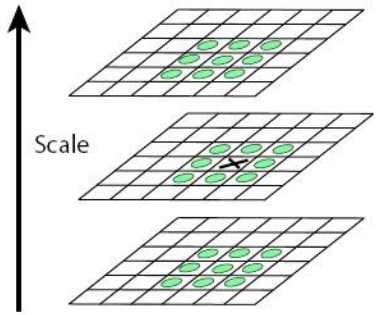
- At each corner, find the *characteristic scale.*
  - Do so using method that is *equivariant* (or *covariant*) w.r.t. to scaling of the image
- Approach: for corner at (x, y) search a range of scales (σ) so that a scale-covariant function reaches a local maximum in scale



"scale space"

Scale of a window

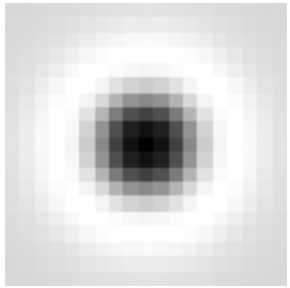
- At each corner, find the *characteristic scale.*
  - Do so using method that is *equivariant* (or *covariant*) w.r.t. to scaling of the image
- Approach: for corner at (x, y) search a range of scales (σ) so that a scale-covariant function reaches a local maximum in scale



"scale space"

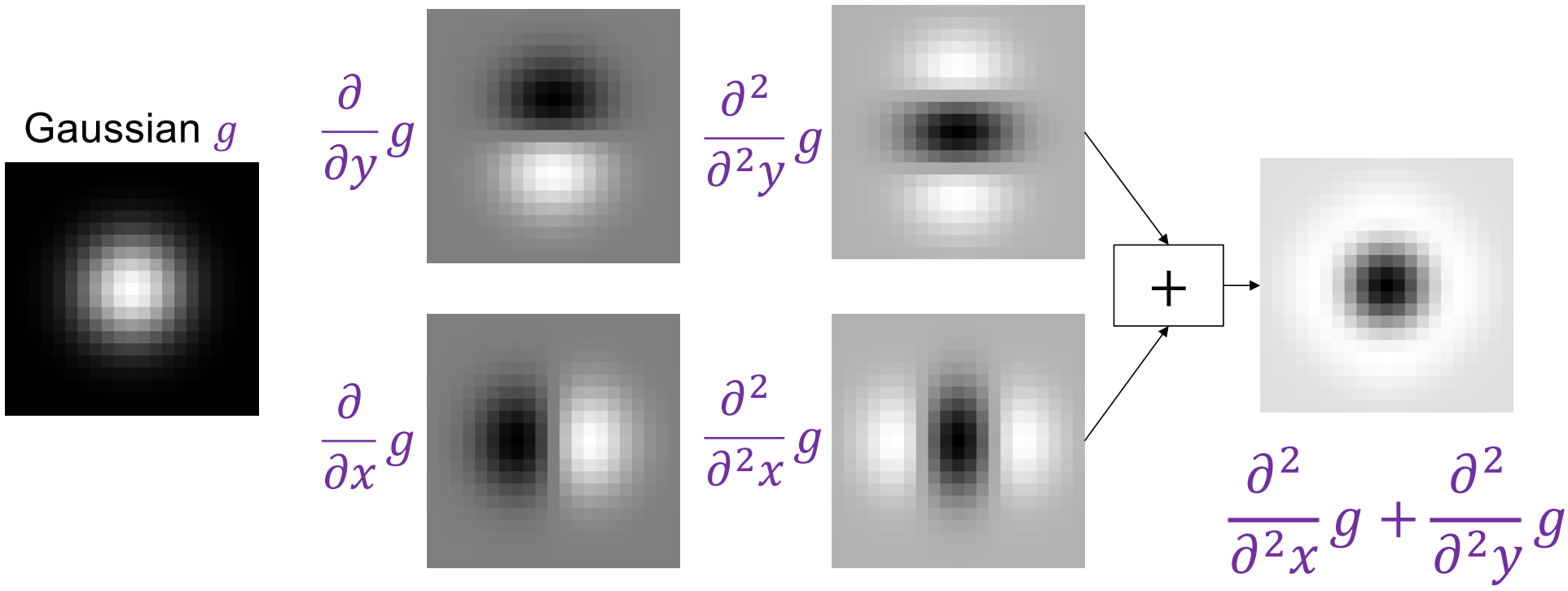
Scale of a window

- At each corner, find the *characteristic scale.*
  - Do so using method that is *equivariant* (or *covariant*) w.r.t. to scaling of the image
- Approach: for corner at (x, y) search a range of scales (σ) so that a scale-covariant function reaches a local maximum in scale
- A particularly convenient response function is given by the *scale-normalized Laplacian of Gaussian (LoG) filter*:



$$\nabla^2_{\text{norm}} = \sigma^2 \left(\frac{\partial^2}{\partial x^2} g + \frac{\partial^2}{\partial^2 y} g \right)$$

Laplacian of Gaussian

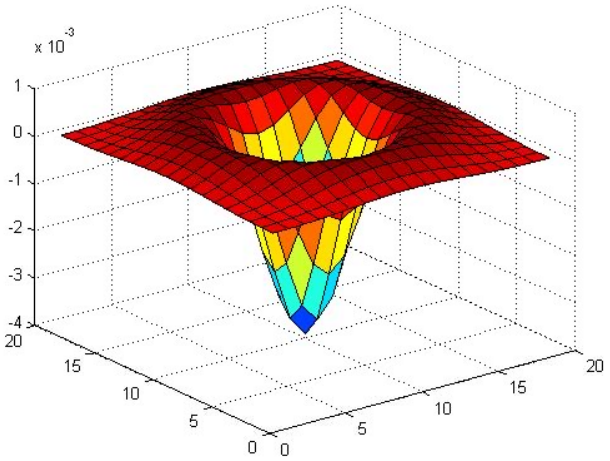


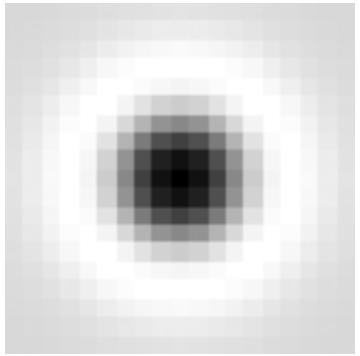
Is this filter separable?

Source: J. Johnson and D. Fouhey

Scale-normalized Laplacian

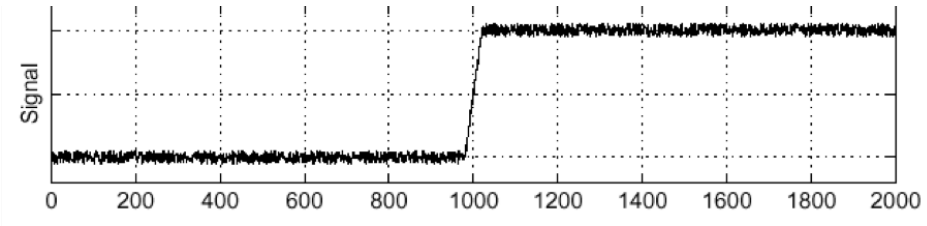
- You need to multiply the LoG by σ^2 to make responses comparable across scales





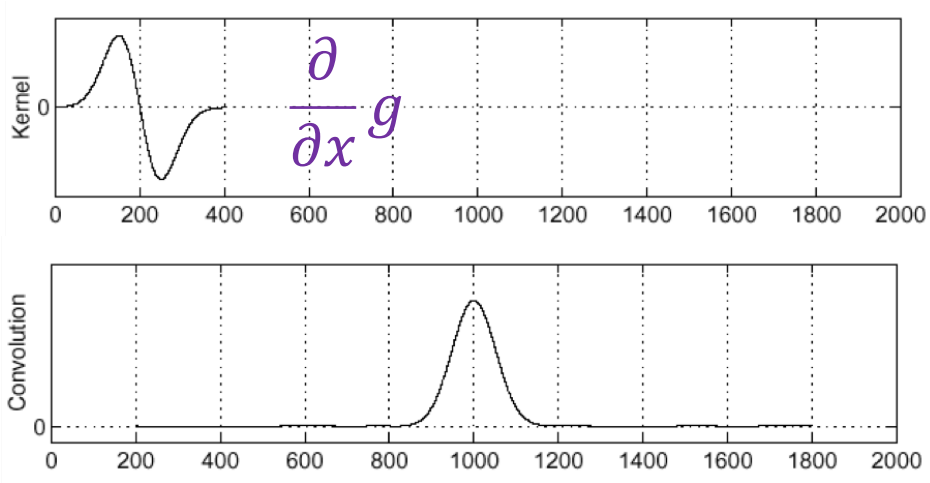
$$\nabla^2_{\text{norm}} = \sigma^2 \left(\frac{\partial^2}{\partial x^2} g + \frac{\partial^2}{\partial^2 y} g \right)$$

Edge detection with LoG



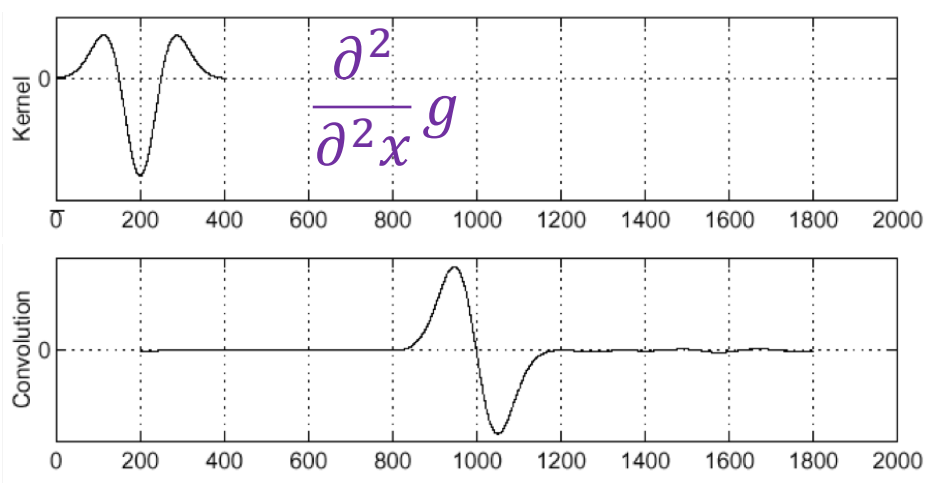
Edge

Derivative of Gaussian



Edge = *local maximum* of derivative

Laplacian of Gaussian



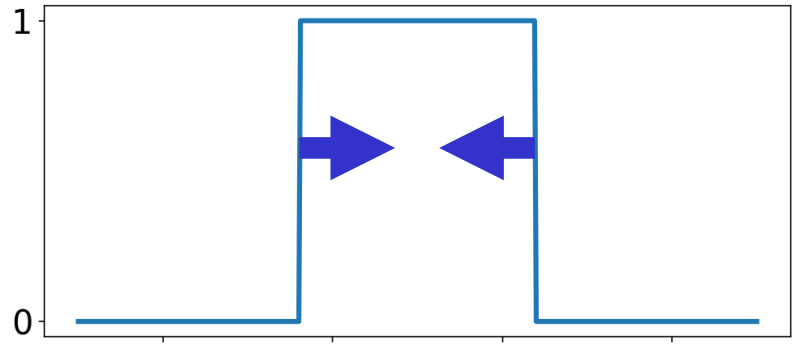
Edge = *zero-crossing* of Laplacian

D. Marr and E. Hildreth, Theory of Edge Detection, Proc. of the Royal Society of London. Series B, Biological Sciences, 207 (1167): 187–217, 1980

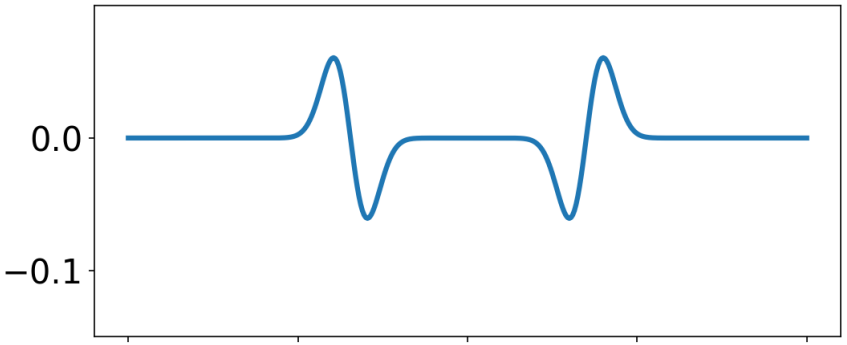
Blob detection with LoG

- Let's convolve a 1D "blob" with the Laplacian:

f



$f * \frac{\partial^2}{\partial^2 x} g$

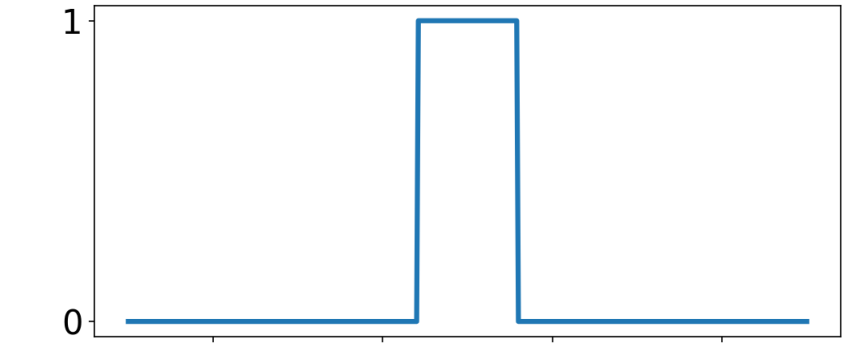


Source: J. Johnson and D. Fouhey

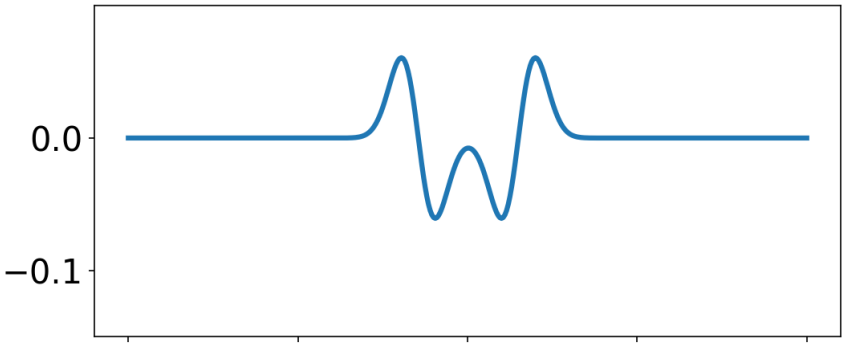
Blob detection with LoG

- Let's convolve a 1D "blob" with the Laplacian:

f



$f * \frac{\partial^2}{\partial^2 x} g$

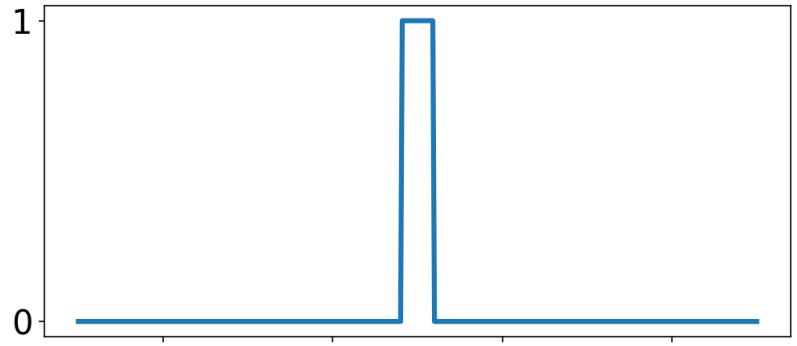


Source: J. Johnson and D. Fouhey

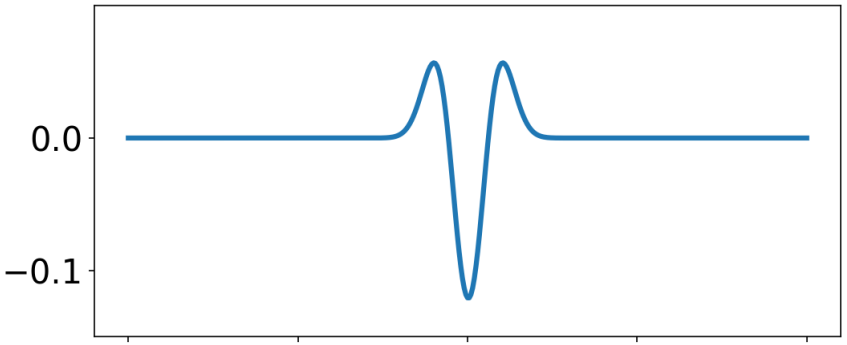
Blob detection with LoG

- Let's convolve a 1D "blob" with the Laplacian:

f



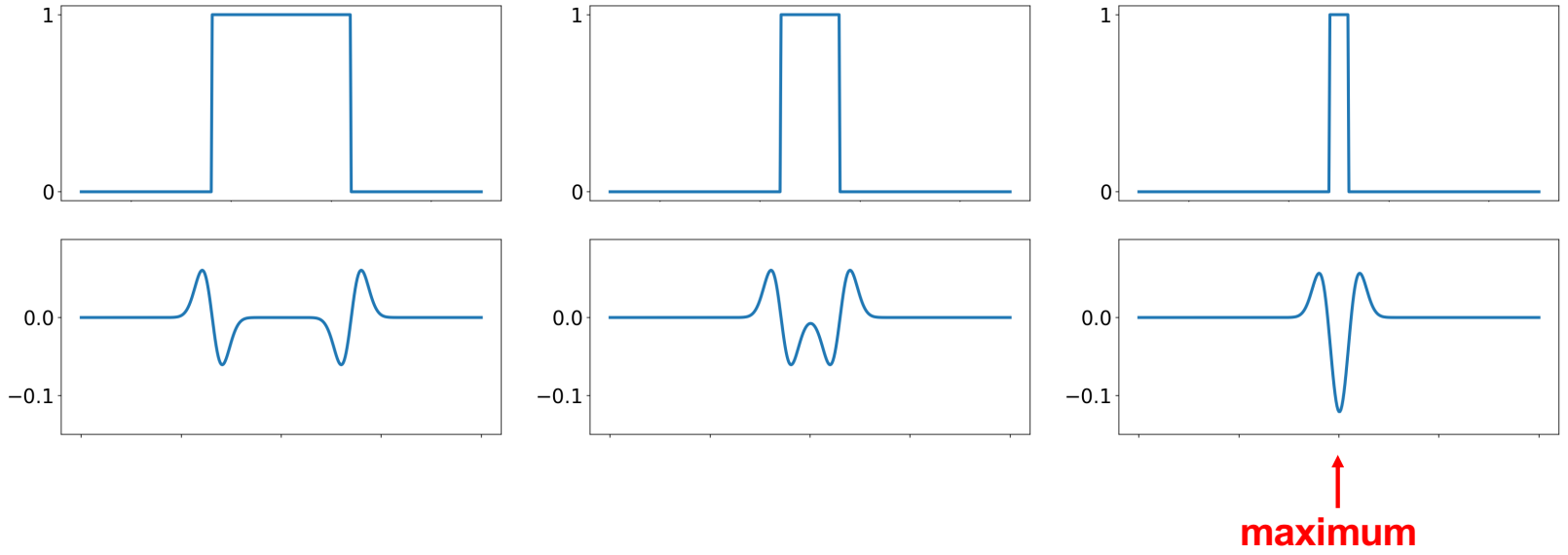
$f * \frac{\partial^2}{\partial^2 x} g$



Source: J. Johnson and D. Fouhey

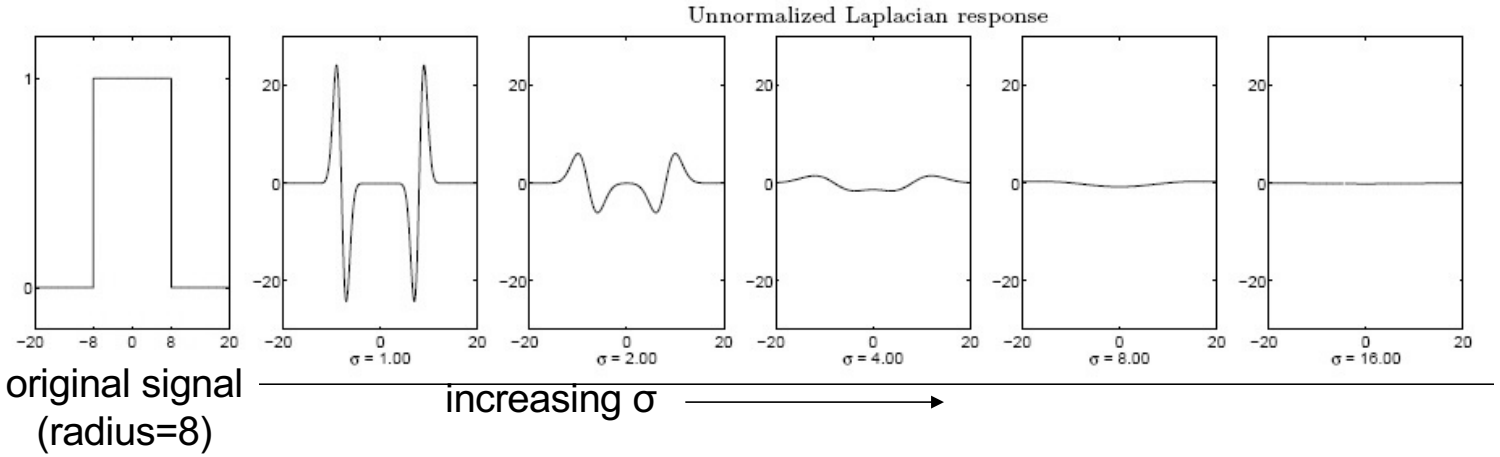
Spatial selection

- The magnitude of the Laplacian response will achieve a maximum at the center of the blob, provided the scale of the Laplacian is “matched” to the scale of the blob



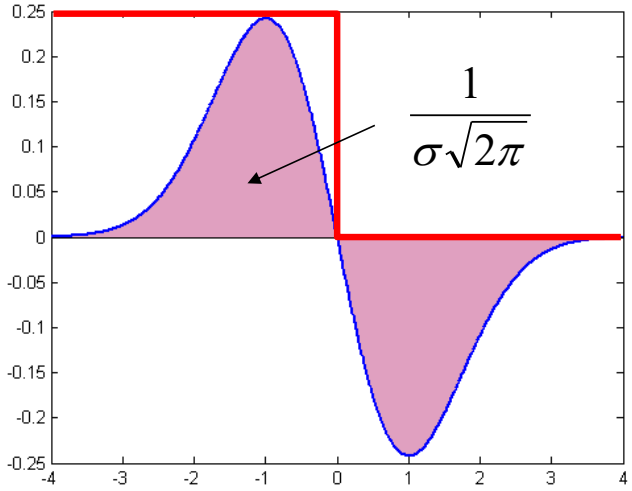
Scale selection

- We want to find the characteristic scale of the blob by convolving it with Laplacians at several scales and looking for the maximum response
- However, Laplacian response decays as scale increases:



Scale normalization

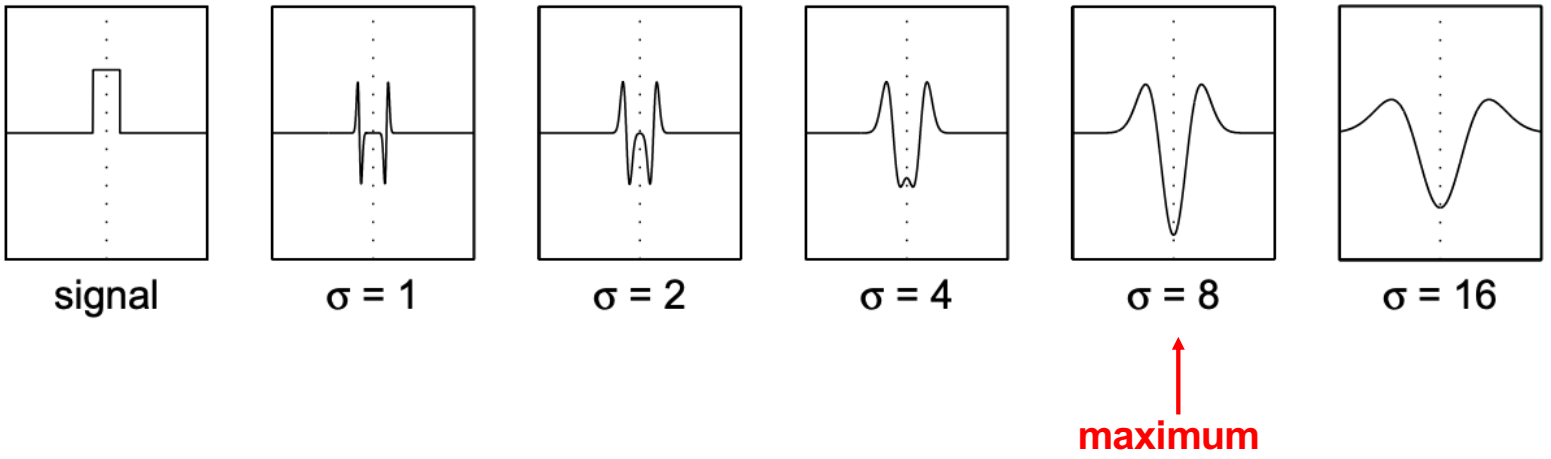
- The response of a derivative of Gaussian filter to a perfect step edge decreases as σ increases:



- To keep response the same (scale-invariant), must multiply Gaussian derivative by σ
- Laplacian is the second Gaussian derivative, so it must be multiplied by σ^2

Scale selection

- We can find the *characteristic scale* of the blob by convolving it with *scale-normalized* Laplacians at several scales (σ) and looking for the maximum response



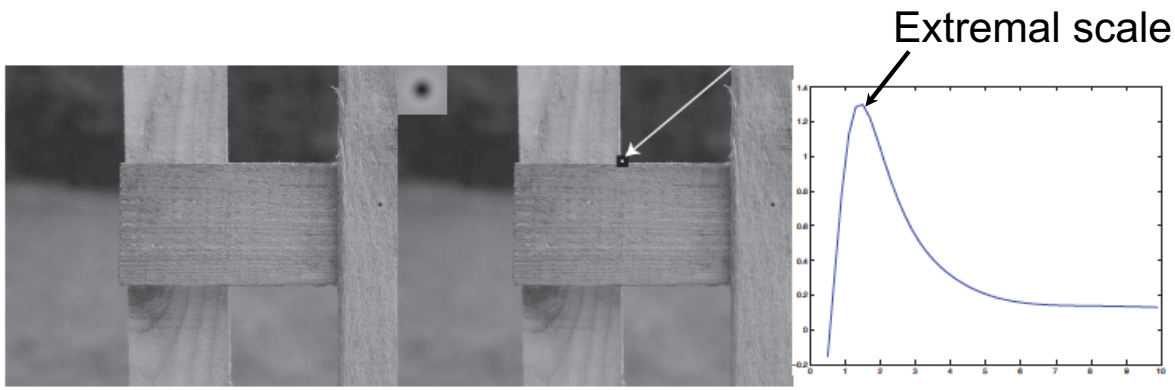


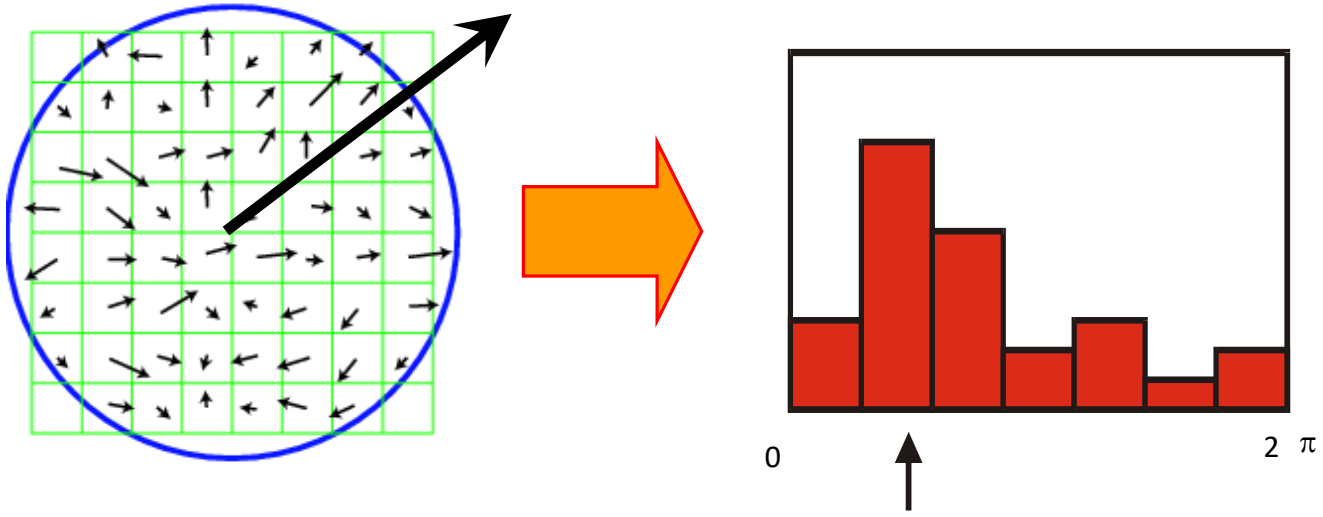
FIGURE 5.12: The scale of a neighborhood around a corner can be estimated by finding a local extremum, in *scale* of the response at that point to a smoothed Laplacian of Gaussian kernel. On the **left**, a detail of a piece of fencing. In the center, a corner identified by an arrow (which points to the corner, given by a white spot surrounded by a black ring). *Overlaid* on this image is a Laplacian of Gaussian kernel, in the **top right** corner; dark values are negative, mid gray is zero, and light values are positive. Notice that, using the reasoning of Section 4.5, this filter will give a strong positive response for a dark blob on a light background, and a strong negative response for a light blob on a dark background, so by searching for the strongest response at this point as a function of scale, we are looking for the size of the best-fitting blob. On the **right**, the response of a Laplacian of Gaussian *at the location of the corner*, as a function of the smoothing parameter (which is plotted in pixels). There is one extremal scale, at approximately 2 pixels. This means that there is one scale at which the image neighborhood looks most like a blob (some corners have more than one scale). © *Dorling Kindersley, used with permission.*

Overview

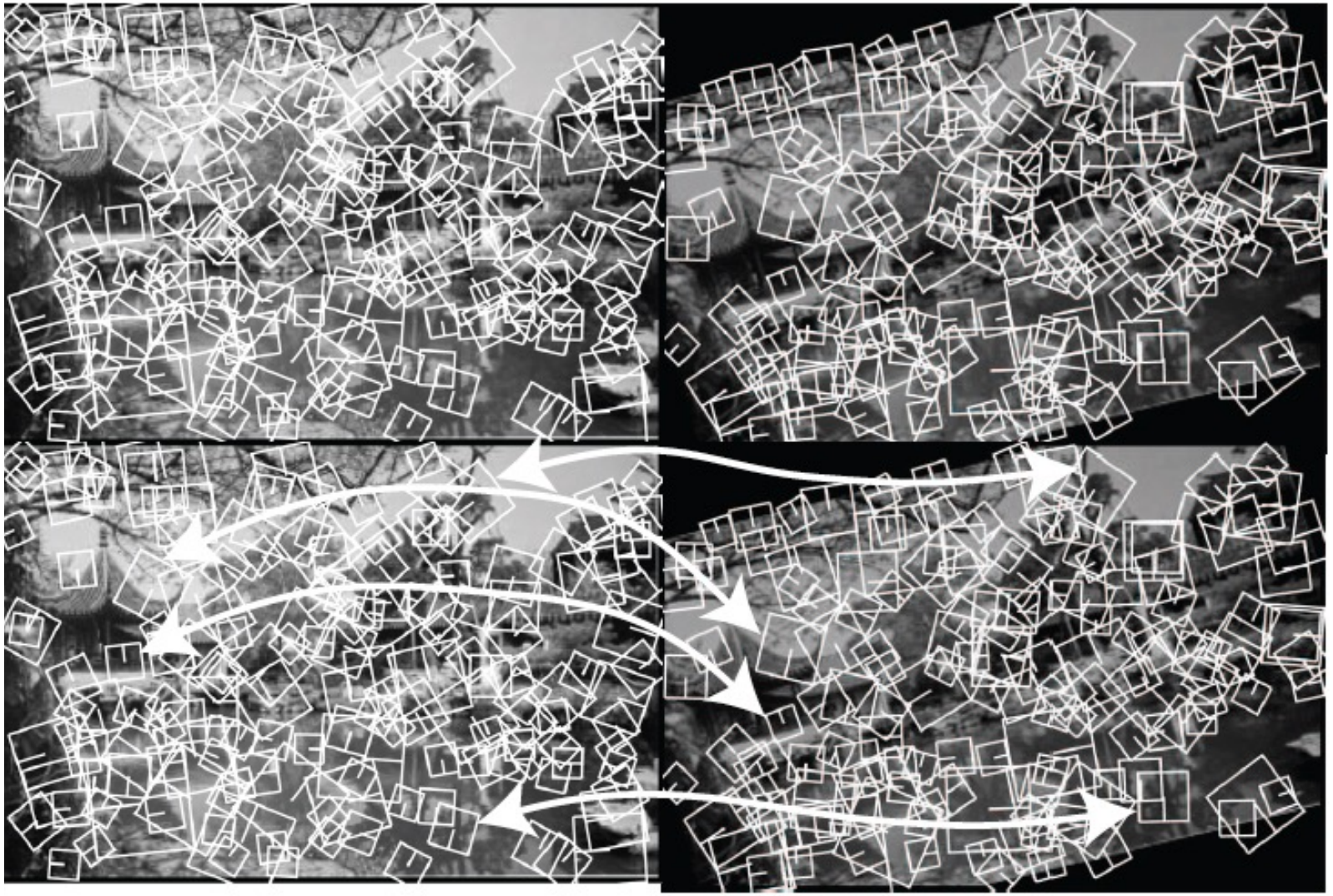
- Windows
  - Location and scale
- Laplacian of Gaussian filter
- Orientation
- SIFT descriptors
- Learning to find interest points and their descriptions

Orientation

- We have
  - Location, scale of window
- Very effective hack for finding a reference orientation:
  - Create histogram of local gradient directions in the patch
  - Assign reference orientation at peak of smoothed histogram



Two peaks? Choose biggest
Same size? Choose randomly



At this point we have Our windows – how do we describe the contents?

Overview

- Windows
  - Location and scale
- Laplacian of Gaussian filter
- Orientation
- SIFT descriptors
- Learning to find interest points and their descriptions

SIFT features

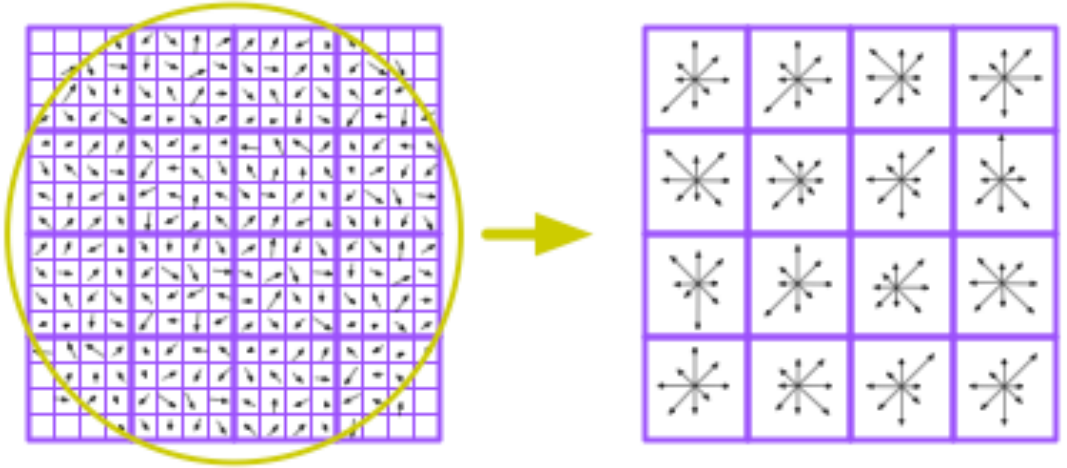
SIFT features

- SIFT=Scale Invariant Feature Transform
- Very strong record of effectiveness in matching applications
- SIFT features behave very well using nearest neighbors matching
  - i.e. the nearest neighbor to a query patch is usually a matching patch

Describing a window's contents

We want description to be:

- Invariant to changes in image brightness: - use orientations
- Robust to noise: - ignore orientations with small magnitude
- Distinctive: - use lots of local orientations
- Invariant to small errors in location:
  - "bucket" the orientations in image
  - Use a histogram for each bucket



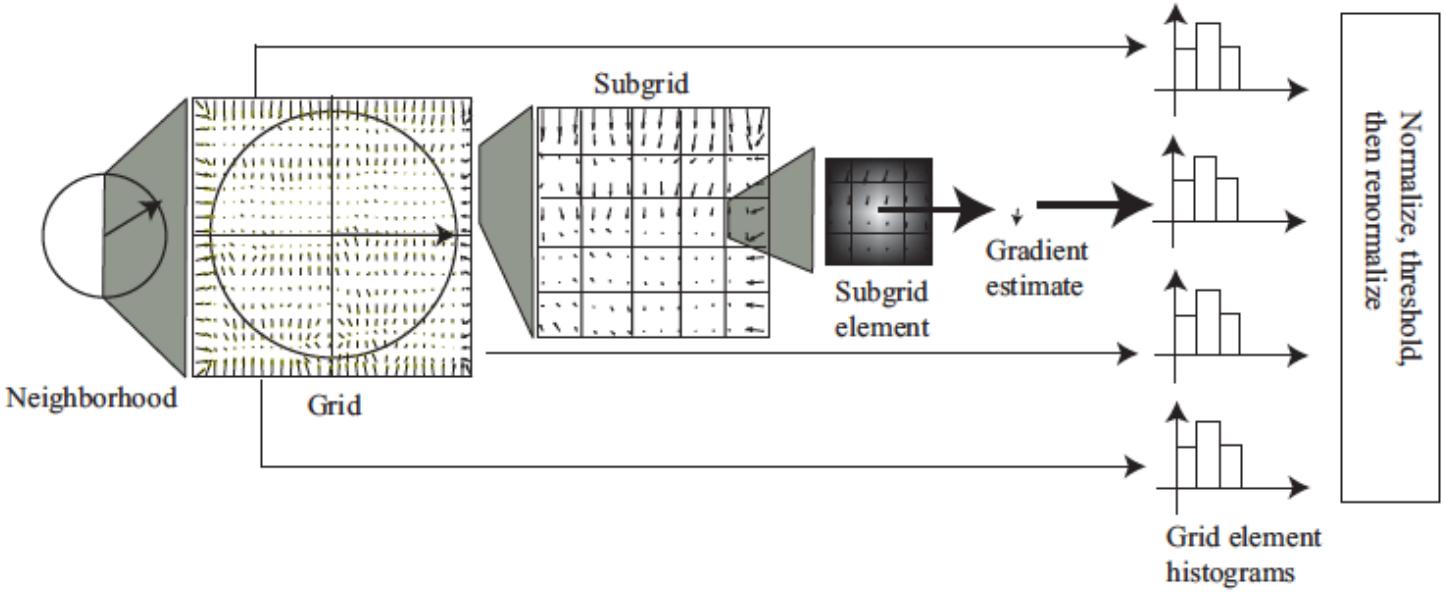
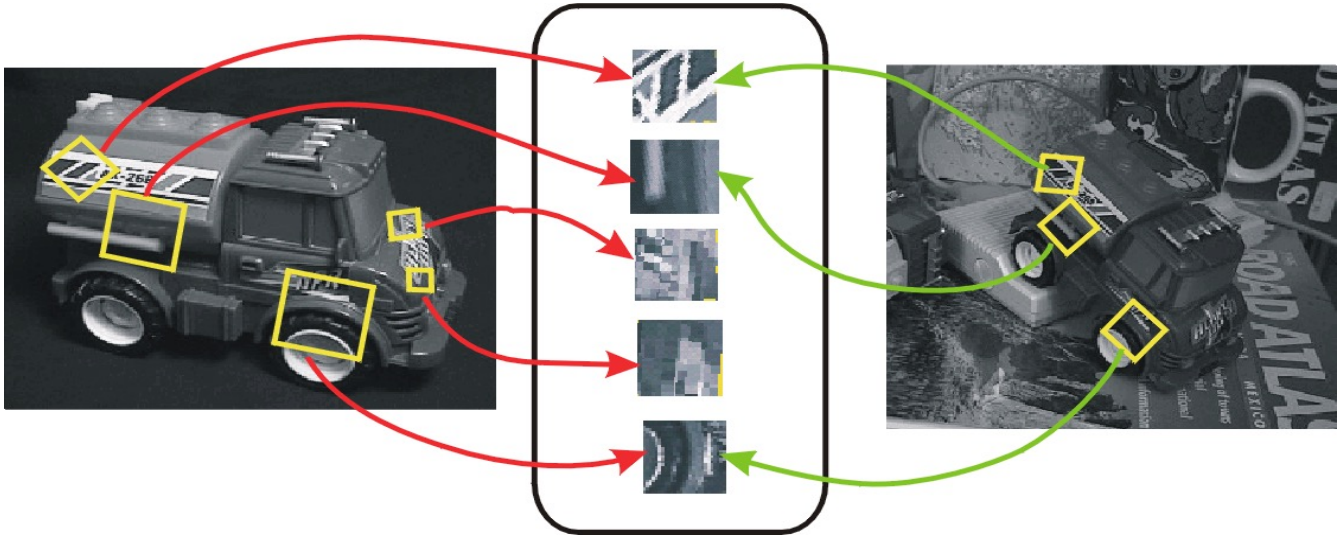


FIGURE 5.14: To construct a SIFT descriptor for a neighborhood, we place a grid over the rectified neighborhood. Each grid is divided into a subgrid, and a gradient estimate is computed at the center of each subgrid element. This gradient estimate is a weighted average of nearby gradients, with weights chosen so that gradients outside the subgrid cell contribute. The gradient estimates in each subgrid element are accumulated into an orientation histogram. Each gradient votes for its orientation, with a vote weighted by its magnitude and by its distance to the center of the neighborhood. The resulting orientation histograms are stacked to give a single feature vector. This is normalized to have unit norm; then terms in the normalized feature vector are thresholded, and the vector is normalized again.

SIFT for matching

- The main goal of SIFT is to enable image matching in the presence of significant transformations
  - To recognize the same keypoint in multiple images, we need to match appearance descriptors or “signatures” in their neighborhoods
  - Descriptors that are *locally* invariant w.r.t. scale and rotation can handle a wide range of *global* transformations



SIFT: Scale-invariant feature transform

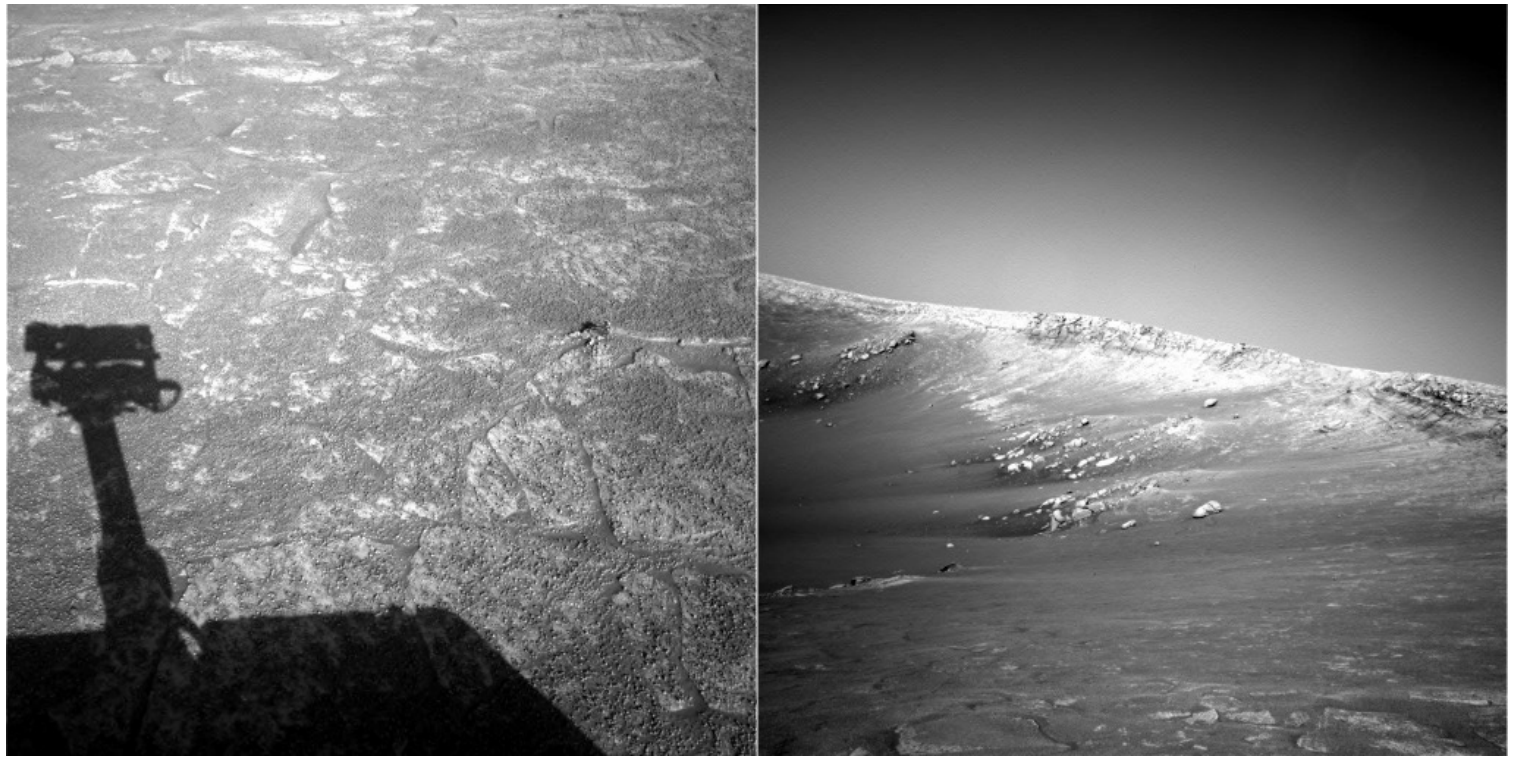
D. Lowe. Object recognition from local scale-invariant features. ICCV 1999

D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV* 60 (2), pp. 91-110, 2004

SIFT for matching

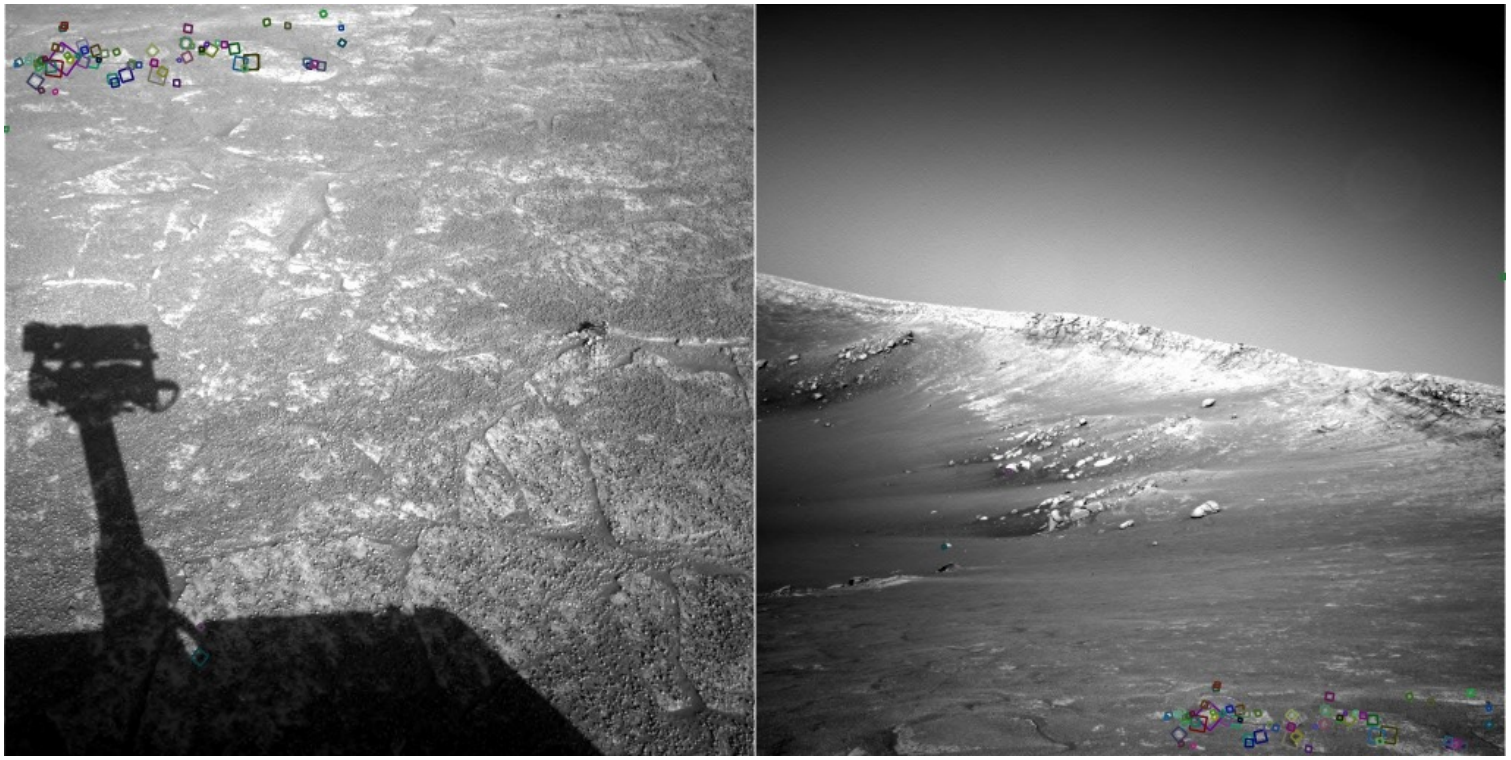
- Extraordinarily robust detection and description technique
  - Can handle changes in viewpoint
    - Up to about 60 degree out-of-plane rotation
  - Can handle significant changes in illumination
    - Sometimes even day vs. night
  - Fast and efficient—can run in real time
  - Lots of code available

A hard matching problem



NASA Mars Rover images

Answer below (look for tiny colored squares…)



NASA Mars Rover images
with SIFT feature matches
Figure by Noah Snavely

Neighborhoods and SIFT - Key Points

- Algorithms to find neighborhoods
  - Represented by location, scale and orientation
  - Neighborhood is covariant
    - If image is translated, scaled, rotated
    - Neighborhood is translated, scaled, rotated
    - Important property for matching
  - Affine covariant constructions are available
- Once found, describe with SIFT features
  - A representation of local orientation histograms, comparable to HOG
  - Normalized differently

Learning to detect and describe keypoints

- You should be able to learn all this
  - keypoints are stable under rotation, translation, scale (homographies)
  - descriptions are stable under rotation, translation, scale (homographies)

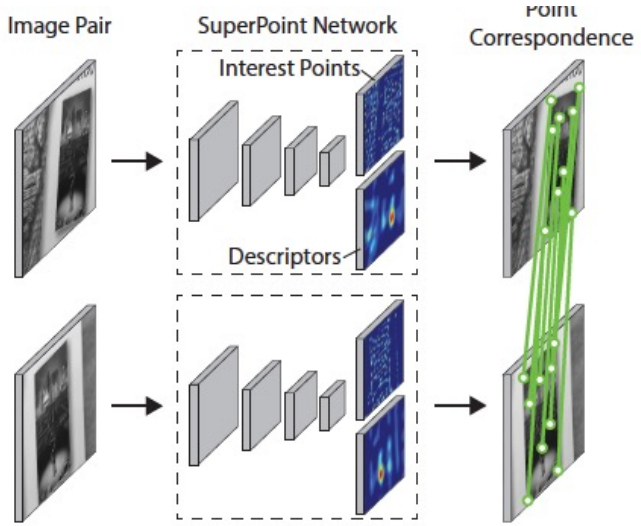


Figure 1. **SuperPoint for Geometric Correspondences.** We present a fully-convolutional neural network that computes SIFT-like 2D interest point locations and descriptors in a single forward pass and runs at 70 FPS on 480×640 images with a Titan X GPU.

DeTone et al, 18

SuperPoint

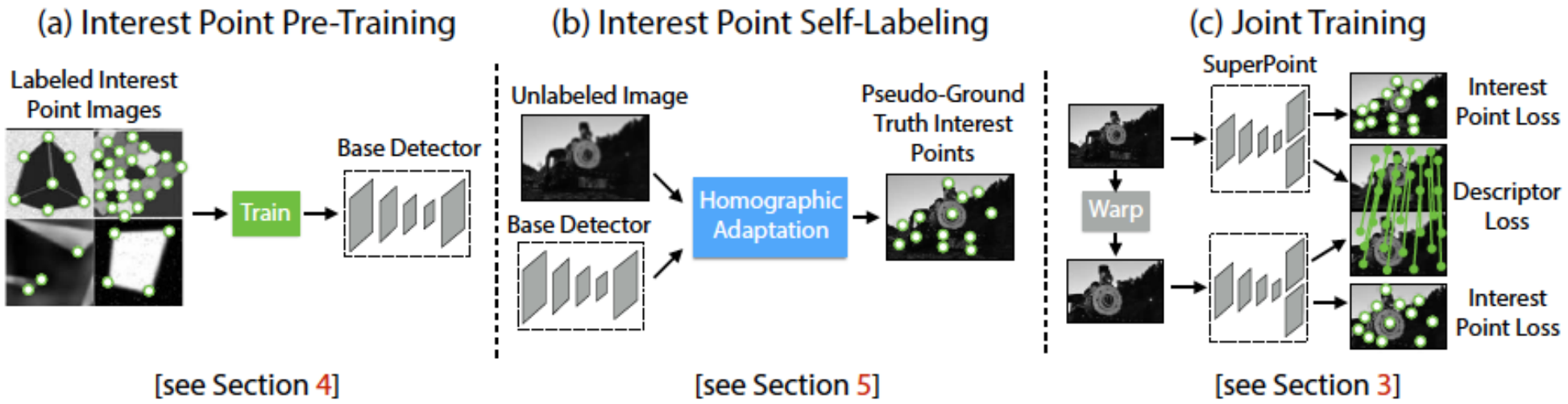


Figure 2. **Self-Supervised Training Overview.** In our self-supervised approach, we (a) pre-train an initial interest point detector on synthetic data and (b) apply a novel Homographic Adaptation procedure to automatically label images from a target, unlabeled domain. The generated labels are used to (c) train a fully-convolutional network that jointly extracts interest points and descriptors from an image.

DeTone et al, 18

SuperPoint

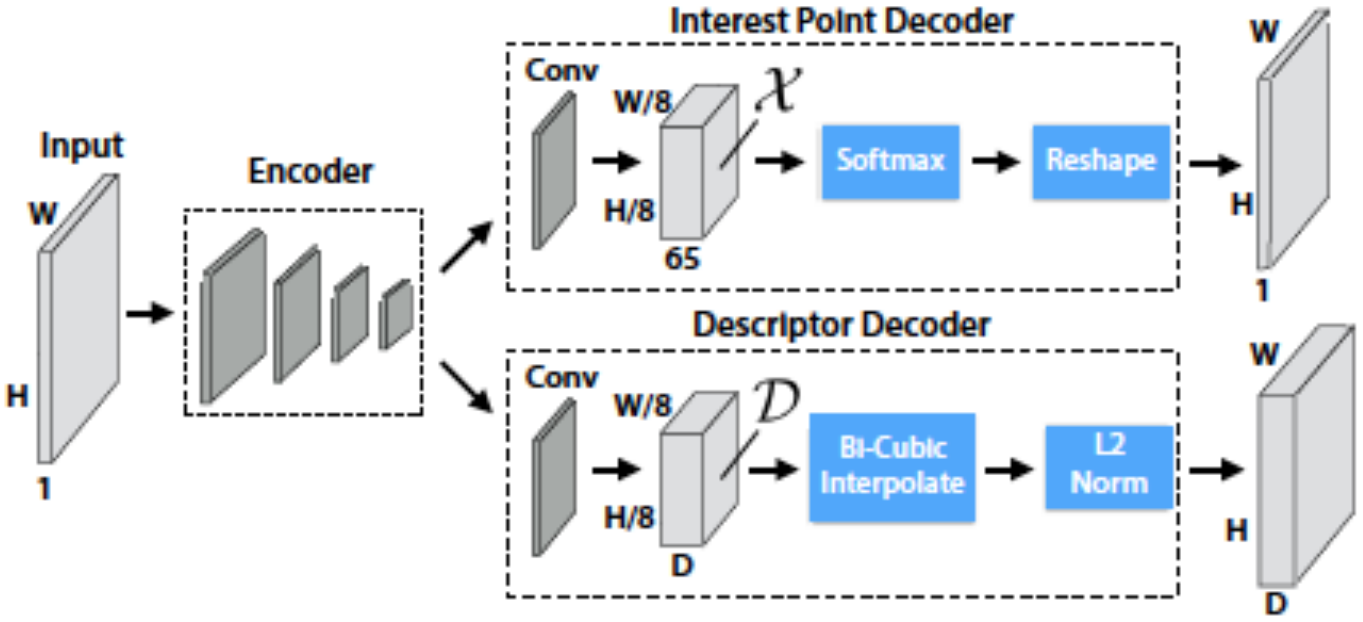


Figure 3. **SuperPoint Decoders.** Both decoders operate on a shared and spatially reduced representation of the input. To keep the model fast and easy to train, both decoders use non-learned upsampling to bring the representation back to $\mathbb{R}^{H \times W}$.

DeTone et al, 18

SuperPoint

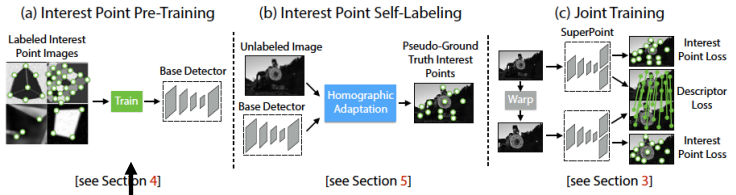


Figure 2. **Self-Supervised Training Overview.** In our self-supervised approach, we (a) pre-train an initial interest point detector on synthetic data and (b) apply a novel Homographic Adaptation procedure to automatically label images from a target, unlabeled domain. The generated labels are used to (c) train a fully-convolutional network that jointly extracts interest points and descriptors from an image.

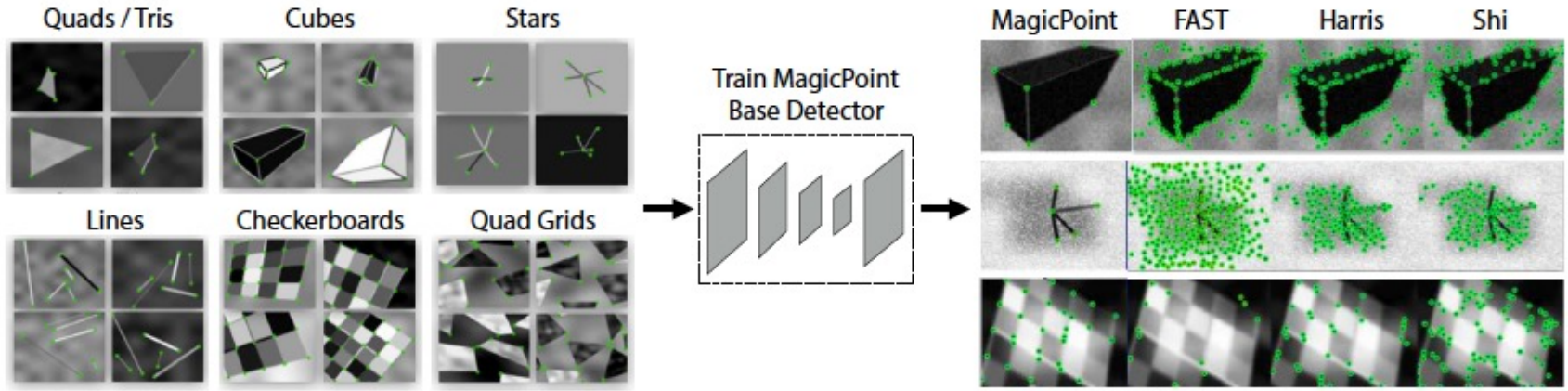


Figure 4. **Synthetic Pre-Training.** We use our Synthetic Shapes dataset consisting of rendered triangles, quadrilaterals, lines, cubes, checkerboards, and stars each with ground truth corner locations. The dataset is used to train the MagicPoint convolutional neural network, which is more robust to noise when compared to classical detectors.

DeTone et al, 18

SuperPoint

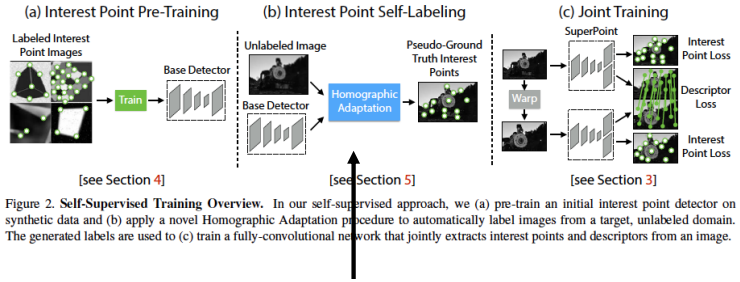


Figure 2. **Self-Supervised Training Overview.** In our self-supervised approach, we (a) pre-train an initial interest point detector on synthetic data and (b) apply a novel Homographic Adaptation procedure to automatically label images from a target, unlabeled domain. The generated labels are used to (c) train a fully-convolutional network that jointly extracts interest points and descriptors from an image.

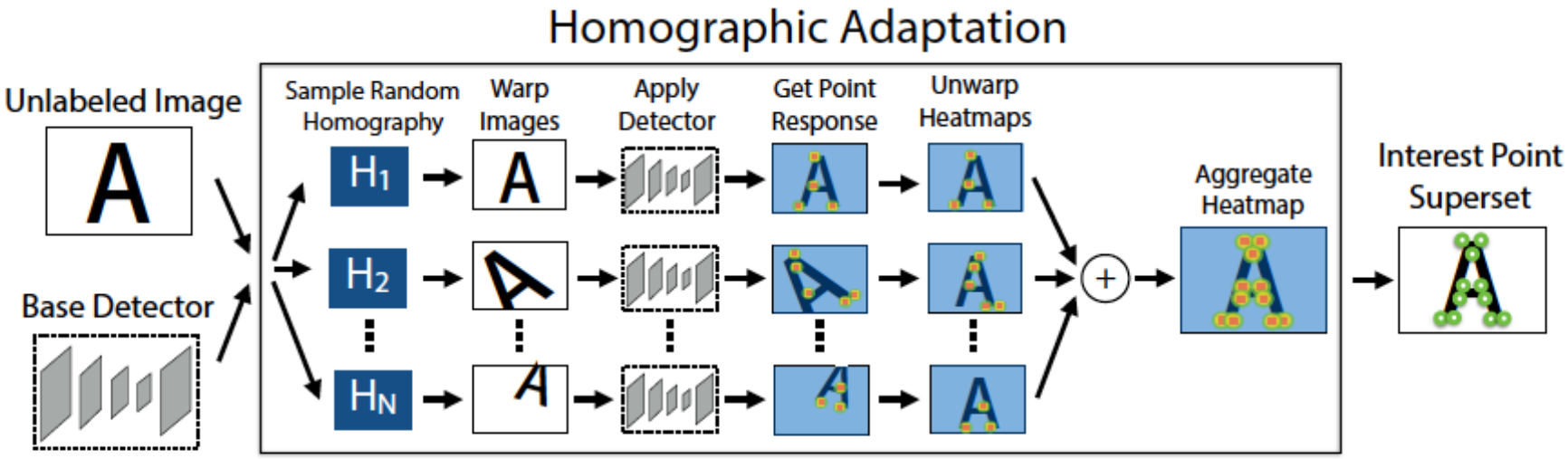


Figure 5. **Homographic Adaptation.** Homographic Adaptation is a form of self-supervision for boosting the geometric consistency of an interest point detector trained with convolutional neural networks. The entire procedure is mathematically defined in Equation 10.

DeTone et al, 18

SuperPoint

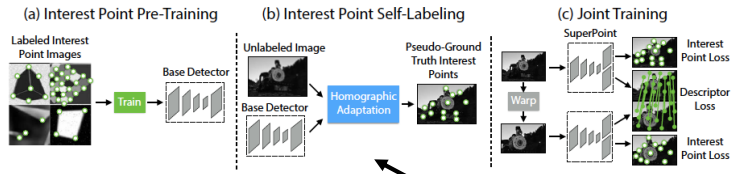


Figure 2. **Self-Supervised Training Overview.** In our self-supervised approach, we (a) pre-train an initial interest point detector on synthetic data and (b) apply a novel Homographic Adaptation procedure to automatically label images from a target, unlabeled domain. The generated labels are used to (c) train a fully-convolutional network that jointly extracts interest points and descriptors from an image.

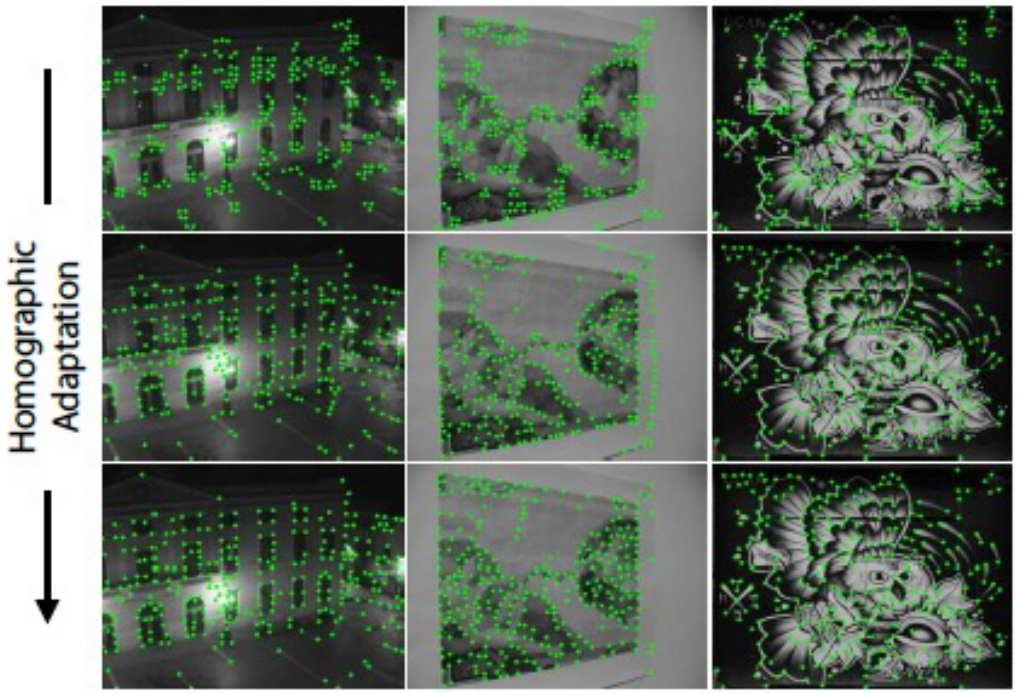


Figure 7. **Iterative Homographic Adaptation.** Top row: initial base detector (MagicPoint) struggles to find repeatable detections. Middle and bottom rows: further training with Homographic Adaption improves detector performance.

DeTone et al, 18

SuperPoint

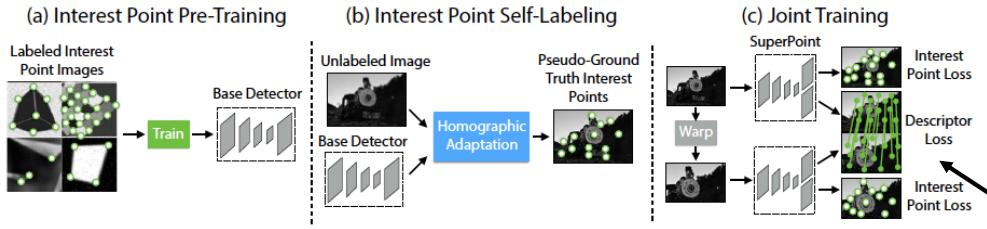
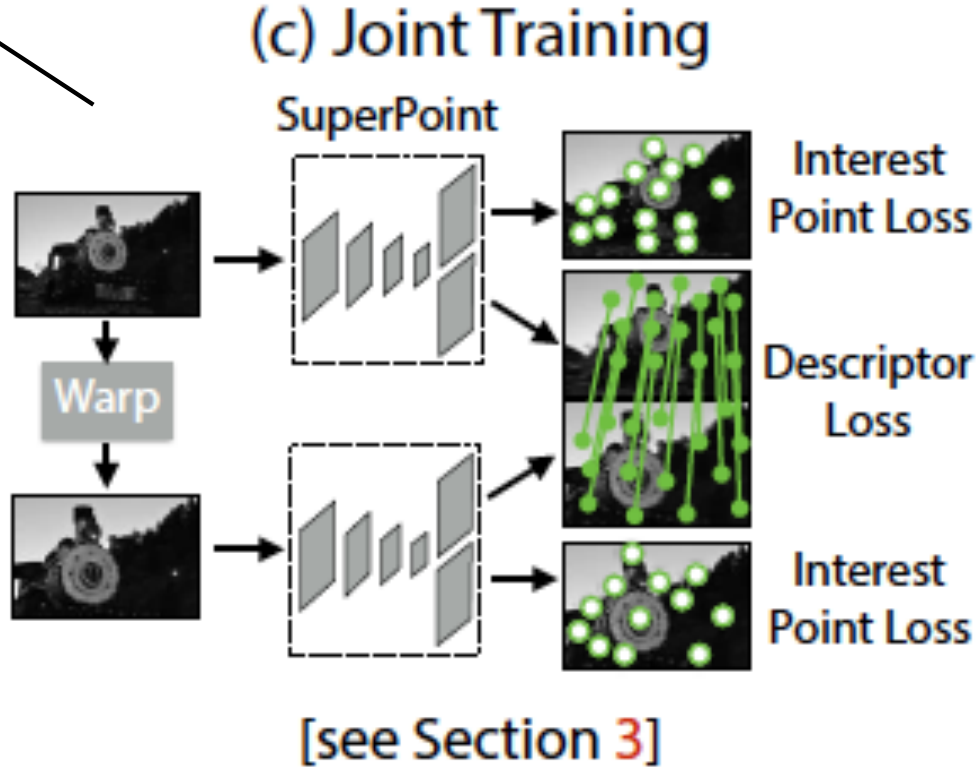


Figure 2. **Self-Supervised Training Overview.** In our self-supervised approach, we (a) pre-train an initial interest point detector on synthetic data and (b) apply a novel Homographic Adaptation procedure to automatically label images from a target, unlabeled domain. The generated labels are used to (c) train a fully-convolutional network that jointly extracts interest points and descriptors from an image.



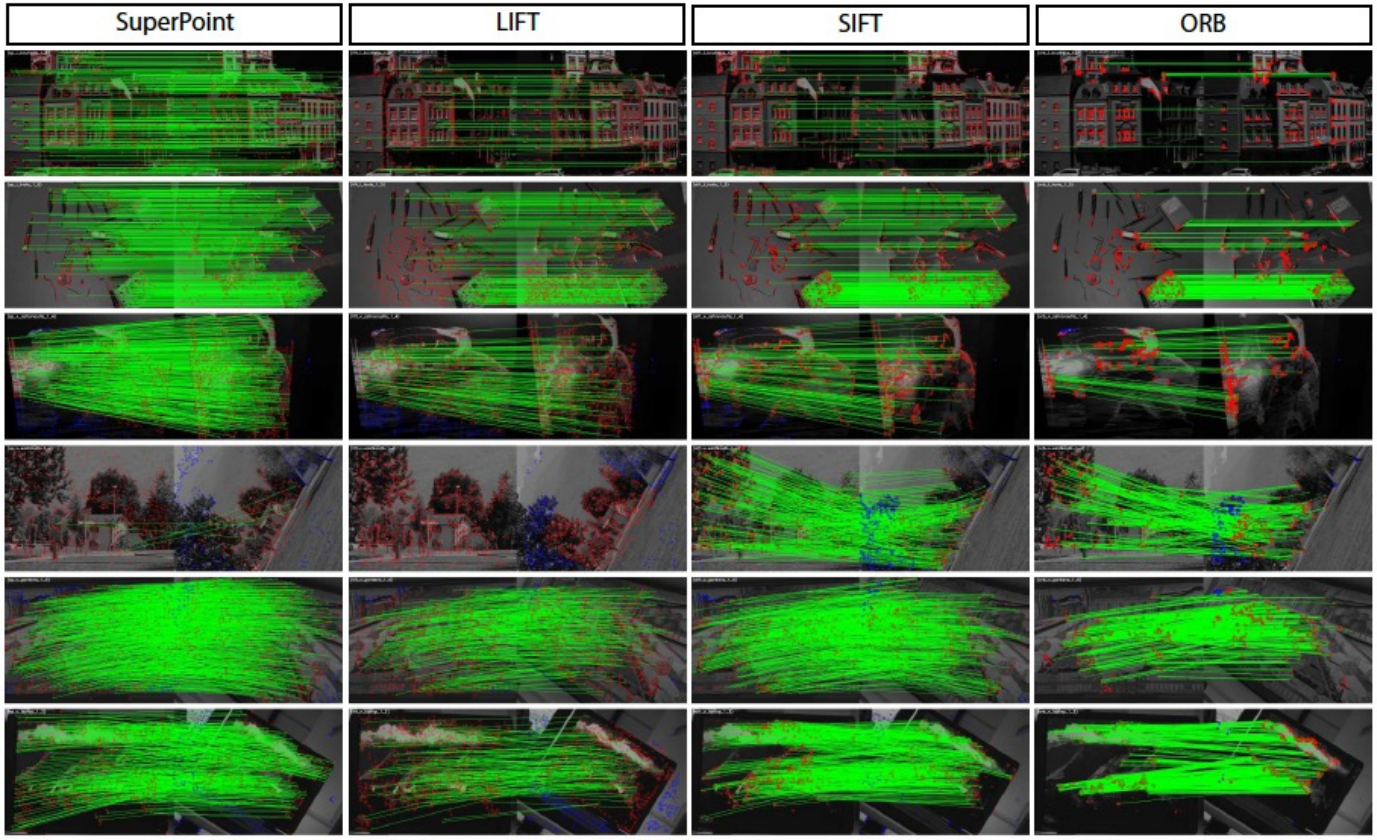


Figure 8. **Qualitative Results on HPatches**. The green lines show correct correspondences. SuperPoint tends to produce more dense and correct matches compared to LIFT, SIFT and ORB. While ORB has the highest average repeatability, the detections cluster together and generally do not result in more matches or more accurate homography estimates (see 4). Row 4: Failure case of SuperPoint and LIFT due to extreme in-plane rotation not seen in the training examples. See Appendix D for additional homography estimation example pairs.

DeTone et al, 18

Overview

- Windows
  - Location and scale
- Laplacian of Gaussian filter
- Orientation
- SIFT descriptors
- Learning to find interest points and their descriptions