Recap

- Where do we look in the image, to describe an image locally?..

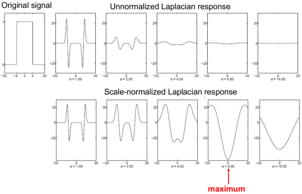
Recap

- Where do we look in the image, to describe an image locally?..

Informative regions: along edges, around corners/keypoints, blobs.

- How do we detect these informative regions in the image?..

- How do we detect these informative regions in the image?..



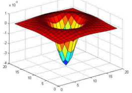
Recap

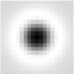
- What is the Laplacian? And in what way is it normalized?

- What is the Laplacian? And in what way is it normalized?

Blob detection in 2D

- Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D





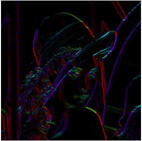
Scale-normalized: $\nabla^2_{\text{norm}} g = \sigma^2 \left(\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)$

Recap

- What image information do we extract to describe an image (locally)?..

- Image edge orientations.

Lena



Gradient orientation

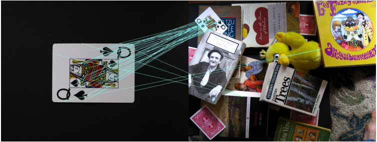
Recap

- What image information do we extract to describe an image (locally)?..
- Image edge magnitude.

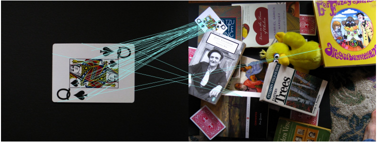

Lena


Gradient magnitude

SIFT: Motivation



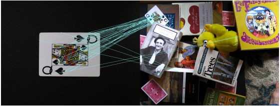
SIFT: Motivation



Find local features — object dependent, focused on the consistent part, ignore background.

SIFT: Motivation

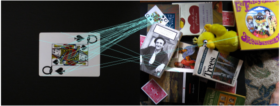
- Image matching...



- Desirable characteristics...

SIFT: Motivation

- Image matching...



- Desirable characteristics...
  - Scale invariance.
  - Rotation invariance.
  - Illumination invariance.
  - Viewpoint invariance.

SIFT: Method overview

1. **Constructing a scale-space.**
2. Laplacian of Gaussian approximation.
3. Finding keypoints.
4. Remove uninformative keypoints.
5. Assign orientation to keypoints.
6. Generate a SIFT descriptor.

Constructing a scale-space

Why do we need a scale-space?...

Constructing a scale-space

How do we construct a scale-space?...

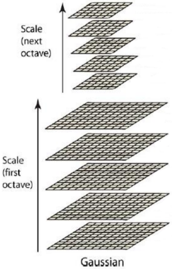
Constructing a scale-space

- Gaussian kernel is the only that is mathematically proven to construct a scale space without spurious resolution [Lindberg, 1994]

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \tag{1}$$

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

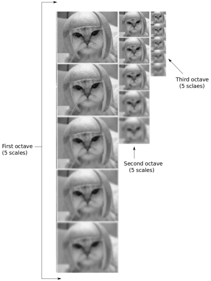
SIFT scale-space: octaves



SIFT scale-space: octaves

1. Take the original image and progressively blur it 5$\times$ with increasing σ.
2. Resize the image to its half and repeat the process.
3. Each "octave" has 5 image.
4. Each image in an octave has a different “scale" (amount of blur).

SIFT scale-space: octaves



SIFT: Method overview

1. Constructing a scale-space.
2. **Laplacian of Gaussian approximation.**
3. Finding keypoints.
4. Remove uninformative keypoints.
5. Assign orientation to keypoints.
6. Generate a SIFT descriptor.

Laplacian of Gaussian approximation

- Laplacian of a 2 D function $f(x, y)$...

Laplacian of Gaussian approximation

- Laplacian of a 2 D function $f(x, y)$...

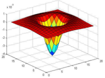
$$\nabla^2 f(x, y) = \frac{\partial^2}{\partial x^2} f(x, y) + \frac{\partial^2}{\partial y^2} f(x, y) \tag{3}$$

Laplacian of Gaussian approximation

- Laplacian of a 2 D function $f(x,y)$...

$$\nabla^2 f(x,y) = \frac{\partial^2}{\partial x^2} f(x,y) + \frac{\partial^2}{\partial y^2} f(x,y) \tag{3}$$

- Laplacian of Gaussian:



$$\nabla^2 G(x,y,\sigma) = \left(\frac{x^2+y^2-\sigma^2}{\sigma^4}\right) e^{-(x^2+y^2)/2\sigma^2}$$

Laplacian of Gaussian approximation

- Laplacian of Gaussian computation on images...

Laplacian of Gaussian approximation

- Laplacian of Gaussian computation on images...

$$\underbrace{\nabla^2(f(x,y)\otimes G(x,y))}_{\boxed{\text{Laplacian of Gaussian-filtered image}}} = \underbrace{\nabla^2 G(x,y)\otimes f(x,y)}_{\boxed{\text{Laplacian of Gaussian (LoG) -filtered image}}}$$

Laplacian of Gaussian approximation

- Laplacian of Gaussian computation on images...

$$\underbrace{\nabla^2 (f(x,y) \otimes G(x,y))}_{\text{Laplacian of Gaussian-filtered image}} = \underbrace{\nabla^2 G(x,y) \otimes f(x,y)}_{\text{Laplacian of Gaussian (LoG)-filtered image}}$$

- It is useful for...

Laplacian of Gaussian approximation

- Laplacian of Gaussian computation on images...

$$\underbrace{\nabla^2(f(x,y)\otimes G(x,y))}_{\text{Laplacian of Gaussian-filtered image}} = \underbrace{\nabla^2 G(x,y)\otimes f(x,y)}_{\text{Laplacian of Gaussian (LoG) -filtered image}}$$

- It is useful for...
  - Locating edges and corners in an image.
  - Edges and corners are good for finding keypoints.

Laplacian of Gaussian approximation

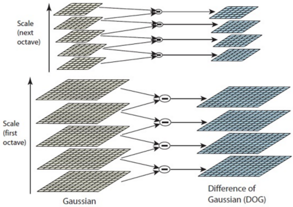
- Laplacian of Gaussian computation on images...

$$\underbrace{\nabla^2(f(x,y) \otimes G(x,y))}_{\boxed{\text{Laplacian of Gaussian-filtered image}}} = \underbrace{\nabla^2 G(x,y) \otimes f(x,y)}_{\boxed{\text{Laplacian of Gaussian (LoG) -filtered image}}}$$

- It is useful for...
  - Locating edges and corners in an image.
  - Edges and corners are good for finding keypoints.
- But second order derivatives are expensive to compute.. and we already have the blurred images.

Laplacian of Gaussian approximation

We can approximate the Laplacian of Gaussian (LoG) with a difference of Gaussian (DoG). Why?



Laplacian of Gaussian approximation

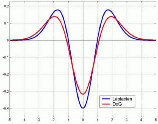
LoG can be approximate by a Difference of two Gaussians (DoG) at different scales.

$$LoG = \sigma^2 \left(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)$$

(Laplacian)

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)



To approximate a LoG for some σ, we need two gaussians with sigmas: $\sqrt{\sigma^2 + \Delta t}$ and $\sqrt{\sigma^2 - \Delta t}$.

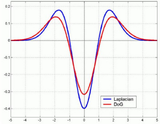
LoG can be approximate by a Difference of two Gaussians (DoG) at different scales.

$$LoG = \sigma^2 \left(G_{xx}(x,y,\sigma) + G_{yy}(x,y,\sigma) \right)$$

(Laplacian)

$$DoG = G(x,y,k\sigma) - G(x,y,\sigma)$$

(Difference of Gaussians)



To approximate a LoG for some σ, we need two gaussians with sigmas: $\sqrt{\sigma^2 + \Delta t}$ and $\sqrt{\sigma^2 - \Delta t}$.

Laplacian of Gaussian approximation: benefits

- Images blurred with a Gaussian kernel are <u>not</u> scale independent. Why?...

Laplacian of Gaussian approximation: benefits

- Images blurred with a Gaussian kernel are not scale independent. Why?...

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (4)$$

Laplacian of Gaussian approximation: benefits

- Images blurred with a Gaussian kernel are not scale independent. Why?...

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{4}$$

- The "scale-invariant" Laplacian should look like this: $\sigma^2 \nabla^2 G(x, y, \sigma)$.

Laplacian of Gaussian approximation: benefits

- Images blurred with a Gaussian kernel are <u>not</u> scale independent. Why?...

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{4}$$

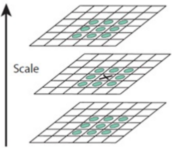
- The "scale-invariant" Laplacian should look like this: $\sigma^2 \nabla^2 G(x,y,\sigma)$.
- And that is the result of DoG.

SIFT: Method overview

1. Constructing a scale-space.
2. Laplacian of Gaussian approximation.
3. **Finding keypoints.**
4. Remove uninformative keypoints.
5. Assign orientation to keypoints.
6. Generate a SIFT descriptor.

Finding keypoints

Find the local minima/maxima by looking at all neighbors of all pixels (across scales also):



To find keypoints at subpixel location, an image Taylor expansion around the minima/maxima points is used.

Finding keypoints

- How many extrema (minima/maxima) images we have in the end per octave?...

Finding keypoints

- How many extrema (minima/maxima) images we have in the end per octave?...
- 5 scales $\rightarrow$ 4 DoG images $\rightarrow$ 2 extrema images (the first and last DoG layer do not have sufficient neighbors, thus no extrema there).