Recap: Smoothing with a Gaussian

Recall: parameter $\sigma$ is the “scale” / “width” / “spread” of the Gaussian kernel, and controls the amount of smoothing.

Recap: Effect of $\sigma$ on derivatives

The apparent structures differ depending on Gaussian’s scale parameter.

Larger values: larger scale edges detected
Smaller values: finer features detected

Multi-scale feature detection and matching

- An interesting property of edges as defined by the zero-crossings of multi-scale operators whose scale is determined by convolution with a Gaussian, is that as the Gaussian is made coarser (larger), new edges (new zero-crossings) can never appear. They can only merge and thus become fewer in number. This property is called causality. It is also sometimes called “monotonicity,” or “the evolution property,” or “nice scaling behaviour.”

- One reason why causality is important is that it ensures that features detected at a coarse scale of analysis were not spuriously created by the blurring process (convolution with a low-pass filter, which is the normal way to create a multi-scale image pyramid using a hierarchy of increasing kernel sizes). One would like to know that image features detected at a certain scale are “grounded” in image detail at the finest resolution.

Multi-scale feature detection and matching

- For purposes of edge detection at multiple scales, a plot showing the evolution of zero-crossings in the image after convolution with a linear operator as a function of the scale of the operator which sets the scale (i.e., the width of the Gaussian), is called scale-space.

- Scale-space has a dimensionality that is one greater than the dimensionality of the signal. Thus a 1D waveform projects into a 2D scale-space. An image projects into a 3D scale space, with its zero-crossings (edges) forming surfaces that evolve as the scale of the Gaussian changes. The scale of the Gaussian, usually denoted by $\sigma$, creates the added dimension.

- A mapping of the edges in an image (its zero-crossings after such filtering operations, evolving with operator scale) is called a scale-space fingerprint. Several theorems exist called “fingerprint theorems” showing that the Gaussian blurring operator uniquely possesses the property of causality. In this respect, it is a preferred edge detector when combined with a bandpass or differentiating kernel such as the Laplacian.

- However, other non-linear operators have advantageous properties, such as reduced noise-sensitivity and greater applicability for extracting features that are more complicated (and more useful) than mere edges.
Scale Invariant Detection

- Consider regions (e.g. circles) of different sizes around a point
- Regions of corresponding sizes will look the same in both images

Scale Invariant Detection

- The problem: how do we choose corresponding circles independently in each image?

Scale Invariant Detection

- Solution:
  - Design a function on the region (circle), which is "scale invariant" (the same for corresponding regions, even if they are at different scales)

Scale Invariant Detection: Summary

- Given: two images of the same scene with a large scale difference between them
- Goal: find the same interest points independently in each image
- Solution: search for extrema of suitable functions in scale and in space (over the image)

Methods:
1. Harris-Laplacian [Mikolajczyk, Schmid]: maximise Laplacian over scale, Harris measure of corner response over the image
2. SIFT [Lowe]: maximise Difference of Gaussians over scale and space

Image Matching

- Algorithm for finding points and representing their patches should produce similar results even when conditions vary
- Buzzword is "invariance"
  - geometric invariance: translation, rotation, scale
  - photometric invariance: brightness, exposure, ...

Invariant local features

- Algorithm for finding points and representing their patches should produce similar results even when conditions vary
- Buzzword is "invariance"
  - geometric invariance: translation, rotation, scale
  - photometric invariance: brightness, exposure, ...

Feature Descriptors
**Feature detection**

Local measure of feature uniqueness

- **“flat” region:** no change in all directions
- **“edge”:** no change along the edge direction
- **“corner”:** significant change in all directions

**Scale invariant detection**

Suppose you’re looking for corners

Key idea: find scale that gives local maximum response in both position and scale: use a Laplacian approximated by difference between two Gaussian filtered images with different sigmas.

---

**Gaussian Pyramid**

All the extra levels add very little overhead for memory or computation!

**The Gaussian Pyramid**

Low resolution

High resolution

**The Laplacian Pyramid**

\[ L_n = G_n - \text{expand}(G_{n+1}) \]

\[ L_0 = G_0 \]

**Laplacian ~ Difference of Gaussian**

\[ \text{DoG} = \text{Difference of Gaussians} \]

Cheap approximation — no derivatives needed.
DoG approximation to LoG

- We can efficiently approximate the (scale-normalised) Laplacian of a Gaussian with a difference of Gaussians:

Scale-Space Pyramid

- Multiple scales must be examined to identify scale-invariant features
- An efficient function is to compute the Difference of Gaussian (DOG) pyramid (Burt & Adelson, 1983)

Laplacian pyramid algorithm

Notice that each layer shows detail at a particular scale — these are bandpass filtered versions of the image.
Showing, at full resolution, the information captured at each level of a Gaussian (top) and Laplacian (bottom) pyramid.

http://www-bcs.mit.edu/people/adelson/pub_pdf/pyramid83.pdf

DoG approximates scale-normalised Laplacian of a Gaussian

\[ \frac{\sigma^2 \nabla^2 G}{\partial \sigma^2} \]

\[ \text{DoG}(x, y, \sigma) = (G(x, y, \sigma) - G(x, y, \sigma) \ast I(x, y) \]

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

\[ \frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G \] (heat diffusion equation)

If we consider the finite difference approximation to \( \frac{\partial G}{\partial \sigma} \) at neighbouring scales \( k\sigma \) and \( \sigma \)

\[ \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \]

then by multiplying by \( k\sigma - \sigma = (k-1)\sigma \) we get

\[ G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma \nabla^2 G \] (15)

Dr Chris Town

Octave increment in scale of the Gaussian Pyramid

\[ \sigma_{i+1} = 2\sigma_i \]

followed by factor-of-two downsampling (for efficiency).

To achieve better performance, each octave is further divided into \( s \) intervals.

Remember that we defined neighbouring scales as

\[ \text{DoG}(x, y, \sigma) = (G(x, y, \sigma) - G(x, y, \sigma) \ast I(x, y) \]

So starting with some \( \sigma_0 \), the next scale parameter will be \( k_0\sigma_0 \), followed by \( k_1k_0\sigma_0 \) etc., so that after \( s \) sublevels of the pyramid we have a complete octave with

\[ k^s\sigma_0 = 2\sigma_0 \]

Therefore \( k = 2^{1/s} \)

Dr Chris Town

SIFT – Scale Invariant Feature Transform

- The \( \sigma \) of the Gaussian filters smooths the image by blurring it, which helps to eliminate noise but also eliminates detail (low-pass filter in the Fourier domain). Convolution with a Gaussian followed by re-sampling is the standard technique for downsampling images, for reasons discussed at the start of this section.

- The constant \( k \) is a multiplicative factor between neighbouring Gaussian-blurred images whose difference we wish to compute to extract stable features. SIFT does this by comparing each pixel in the DoG images to its eight neighbours at the same scale and nine corresponding neighbouring pixels in each of the adjacent scales (pyramid levels).

Dr Chris Town

\[ k = 2^{1/s} \]

and the value of \( \sigma \) at octave \( i \) and interval \( n \) of the pyramid is given by

\[ \sigma(i, n) = \sigma_0 2^{i+n/s} \]

\[ n \in [0, s-1] \]

A value of \( s = 3 \) was found by Lowe to provide a good accuracy vs efficiency trade-off. The number of octaves depends on original image resolution.

Dr Chris Town
Key point localization with DoG

- Detect extrema of difference-of-Gaussian (DoG) in scale space
- Then reject points with low contrast (threshold)
- Eliminate edge responses

Feature Descriptors: SIFT

- Scale Invariant Feature Transform
- Descriptor computation:
  - Divide patch into 4x4 sub-patches: 16 cells
  - Compute histogram of gradient orientations (8 reference angles) for all pixels inside each sub-patch
  - Resulting descriptor: 4x4x8 = 128 dimensions

Rotation Invariant Descriptors

- Find local orientation
  - Dominant direction of gradient for the image patch
- Rotate patch according to this angle
  - This puts the patches into a canonical orientation.

Example of Keypoint Detection

- Slide credit: David Lowe

Feature Descriptors: SIFT

- Scale Invariant Feature Transform
- Descriptor computation:
  - Divide patch into 4x4 sub-patches: 16 cells
  - Compute histogram of gradient orientations (8 reference angles) for all pixels inside each sub-patch
  - Resulting descriptor: 4x4x8 = 128 dimensions

Rotation Invariant Descriptors

- Find local orientation
  - Dominant direction of gradient for the image patch
- Rotate patch according to this angle
  - This puts the patches into a canonical orientation.
Orientation Normalisation: Computation

- Compute orientation histogram
- Select dominant orientation
- Normalise: rotate to fixed orientation

Feature stability to noise
- Match features after random change in image scale & orientation, with differing levels of image noise
- Find nearest neighbor in database of 30,000 features

Feature stability to affine change
- Match features after random change in image scale & orientation, with 2% image noise, and affine distortion
- Find nearest neighbor in database of 30,000 features

Distinctiveness of features
- Vary size of database of features, with 30 degree affine change, 2% image noise
- Measure % correct for single nearest neighbor match

Working with SIFT Descriptors
- One image yields:
  - n 128-dimensional descriptors: each one is a histogram of the gradient orientations within a patch
    - \([n \times 128]\) matrix
  - n scale parameters specifying the size of each patch
    - \([n \times 1]\) vector
  - n orientation parameters specifying the angle of the patch
    - \([n \times 1]\) vector
  - n 2D points giving positions of the patches
    - \([n \times 2]\) matrix

SIFT

Figure 12: The training images for two objects are shown on the left. These can be marginalized to a cluttered image with extensive occlusions, shown in the middle. The results of recognition are shown on the right. A part histogram is drawn around each recognized object showing the boundaries of the original training image under the affine transformation subject for buyer recognition. Smaller squares indicate the keypoints that were used for recognition.
**Feature matching**

![Feature matching image](image1)

---

**Image stitching**

![Image stitching image](image2)

---

**Nearest-neighbor matching**

- Solve following problem for all feature vectors, \( x \):
  \[
  \forall j \, \text{NN}(j) = \arg \min_i \| x_i - x_j \|, \; i \neq j
  \]
- Nearest-neighbour matching is the major computational bottleneck
  - Linear search performs \( d n^2 \) operations for \( n \) features and \( d \) dimensions
  - No exact methods are faster than linear search for \( d > 10 \)
  - Approximate methods can be much faster, but at the cost of missing some correct matches. Failure rate gets worse for large datasets.

---

**Approximate k-d tree matching**

Key idea:
- Search k-d tree bins in order of distance from query
- Requires use of a priority queue

---

**K-d tree construction**

**Simple 2D example**

![K-d tree construction example](image3)

---

**K-d tree query**

![K-d tree query example](image4)
Recognition with Local Features
• Image content is transformed into local features that are invariant to translation, rotation, and scale

Fourier transform
\[ \text{Fourier transform} = \text{Fourier bases} \times \text{pixel domain image} \]

Local Features, e.g. SIFT

Dr Chris Town

Gaussian pyramid
\[ \text{Gaussian pyramid} = \text{pixel image} \times \] Overcomplete representation. Low-pass filters, sampled appropriately for their blur.

Laplacian pyramid
\[ \text{Laplacian pyramid} = \text{pixel image} \times \] Overcomplete representation. Transformed pixels represent bandpassed image information.

Edge Fitting
• Edge Detection:
  - The process of labeling the locations in the image where the gray level’s “rate of change” is high.
  • OUTPUT: “edgel” locations, direction, strength

Edge Integration, Contour fitting:
  - The process of combining “local” and perhaps sparse and non-contiguous “edgel” data into meaningful, long edge curves (or closed contours) for segmentation
  • OUTPUT: edges/curves consistent with the local data

Active Contours (“snakes”).
\[ \arg \min \int ((M - I)^2 + \lambda (\nabla M)^2) \, dx \]
where \( M \) is the shape model, and \( I \) is the image data

Thus we have the combination of two factors: a data term and a cost term (the latter sometimes also called a smoothness term or an energy term), which are in contention, in the following sense: we could fit the available edge data with arbitrary high precision, if we used a model with enough complexity; but simpler models are generally more useful and credible than overly complex models (which “over-fit” the data).
Framework for snakes

- A higher level process or a user initialises any curve close to the object boundary.
- The snake then starts deforming and moving towards the desired object boundary.
- In the end it completely “shrink-wraps” around the object.

Modeling

- The contour is defined in the (x, y) plane of an image as a parametric curve:
  \[ \mathbf{w}(s) = (x(s), y(s)) \]
- Contour is said to possess an energy (\( E_{\text{snake}} \)) which is defined as the sum of the three energy terms:
  \[ E_{\text{snake}} = E_{\text{internal}} + E_{\text{external}} + E_{\text{constraint}} \]
- The energy terms are defined in a way such that the final position of the contour will have minimum energy (\( E_{\text{min}} \)).
- Therefore our problem of detecting objects reduces to an energy minimisation problem.

Internal Energy (\( E_{\text{int}} \))

- Depends on the intrinsic properties of the curve.
- Sum of elastic energy and bending energy.

Elastic Energy (\( E_{\text{elastic}} \)):

- The curve is treated as an elastic rubber band possessing elastic potential energy.
- It discourages stretching by introducing tension.
  \[ E_{\text{elastic}} = \frac{1}{2} \int \alpha(s) |v_s|^2 \, ds \]
- Weight \( \alpha(s) \) allows us to control elastic energy along different parts of the contour. Considered to be constant \( \alpha \) for many applications.
- Responsible for shrinking of the contour.

Bending Energy (\( E_{\text{bending}} \)):

- The snake is also considered to behave like a thin metal strip giving rise to bending energy.
- It is defined as sum of squared curvature of the contour.
  \[ E_{\text{bending}} = \frac{1}{2} \int \beta(s) |v_{ss}|^2 \, ds \]
- \( \beta(s) \) plays a similar role to \( \alpha(s) \).
- Bending energy is minimum for a circle.
- Total internal energy of the snake can be defined as
  \[ E_{\text{int}} = E_{\text{elastic}} + E_{\text{bending}} + \frac{1}{2} |\alpha| |v_s|^2 + \frac{1}{2} |\beta| |v_{ss}|^2 \]
External energy of the contour ($E_{\text{ext}}$)

- Image fitting
  $$E_{\text{ext}} = \int E_{\text{image}}(v(s)) ds$$

For example
- $E_{\text{edge}} = -|\nabla I(x, y)|^2$
- $E_{\text{edge}} = -|G_f * \nabla^2 I|^2$

2D Gabor “Logons” Quadrature pair wavelets

$$f(x) = \exp(-i\mu_0 (x - x_0)) \exp(- (x - x_0)^2/\alpha^2)$$

$$F(\mu) = \exp(-i\mu (\mu - \mu_0)) \exp(- (\mu - \mu_0)^2/\sigma^2)$$

Note that for the case of a wavelet $f(x)$ centred on the origin ($x_0 = 0$), its Fourier Transform $F(\mu)$ is simply a Gaussian centred on the modulation frequency $\mu = \mu_0$, and whose width is $1/\alpha$, the reciprocal of the wavelet’s space constant. This shows that it acts as a bandpass filter, passing only those frequencies that are within about $\pm 1/\alpha$ of the wavelet’s modulation frequency $\mu_0$.

Generalisation of wavelet Logons to 2D for image analysis

Two-dimensional Gabor wavelets have the functional form:

$$f(x, y) = e^{-[(x - x_0)^2 + (y - y_0)^2]/\rho^2} e^{-i\mu_0 (x - x_0) + i\alpha \cdot \arctan(y_0/\rho_0)}$$

where $(x_0, y_0)$ specify position in the image, $(\alpha, \beta)$ specify effective width and length, and $(\rho_0, \sigma_0)$ specify modulation, which has spatial frequency $\omega_0 = \sqrt{\rho_0^2 + \sigma_0^2}$ and direction $\theta_0 = \arctan(y_0/\rho_0)$. (A further degree-of-freedom not included above is the relative orientation of the elliptic Gaussian envelope, which creates cross-terms in $x$ and $y$).

2D Fourier transform $F(u, v)$

$$F(u, v) = e^{-[(u - \mu_0)^2 + (v - \mu_0)^2]/\rho^2} e^{-i\mu (u - \mu_0) + i\alpha \cdot \arctan(v_0/\rho_0)}$$

The real part of one member of the 2D Gabor filter family, centred at the origin $(x_0, y_0) = (0, 0)$ and with unity aspect ratio $\beta/\alpha = 1$.
Generating Functions

By appropriately parameterising them for dilation, rotation, and translation, 2D Gabor wavelets can form a complete self-similar (but non-orthogonal) expansion basis for images.

\[ \Psi_{mpq}(x', y') = 2^{-2m} \Psi(x', y') \]

where the substituted variables \((x', y')\) incorporate dilations in size by \(2^{-m}\), translations in position \((p, q)\), and rotations through orientation \(\theta\).

\[
x' = 2^{-m}[x \cos(\theta) + y \sin(\theta)] - p
\]

\[
y' = 2^{-m}[-x \sin(\theta) + y \cos(\theta)] - q
\]

Since the wavelets are dilates, translates, and rotates of each other, such a transform seeks to extract image structure in a way that may be invariant to dilation, translation, and rotation of the original image or pattern.

Gabor wavelets

\[ \psi_c(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}} \cos(2\pi u_0 x) \]

\[ \psi_s(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}} \sin(2\pi u_0 x) \]

A. Torralba

Dilation and rotation

Frequency, orientation and symmetry (phase)
Wavelet (QMF) transform

\[
\text{Wavelet pyramid} = \ast
\]

Ortho-normal transform (like Fourier transform), but with localized basis functions.

Steerable pyramid

\[
\text{Steerable pyramid} = \ast
\]

Multiple orientations at one scale

Over-complete representation, but non-aliased subbands.

Unification of Domains

\[ f(x) = e^{-\alpha x^2} e^{-(x-x_0)^2/\alpha^2} \]

The single parameter \( \alpha \) (the space-constant in the Gaussian term) actually builds a continuous bridge between the two domains: if the parameter \( \alpha \) is made very large, then the second exponential above approaches 1, and so in the limit our expansion becomes

\[ \lim_{\alpha \to \infty} f(x) = e^{\alpha x^2} \]

the ordinary Fourier basis! If the parameter \( \alpha \) is instead made very small, the Gaussian term becomes the approximation to a delta function at location \( x_0 \). Thus our expansion basis implements pure space-domain sampling:

\[ \lim_{\alpha \to 0} f(x) = \delta(x-x_0) \]

Hence the Gabor expansion basis “contains” both domains at once. It allows us to make a continuous deformation that selects a representation lying anywhere on a one-parameter continuum between two domains that were hitherto distinct.

Gabor-Heisenberg-Weyl Uncertainty Principle

then it can be proven (by Schwartz Inequality arguments) that there exists a fundamental lower bound on the product of these two “spreads,” regardless of the function \( f(x) \):

\[ (\Delta x)(\Delta \nu) \geq \frac{1}{4\pi} \]

The unique family of signals that actually achieve the lower bound in the Gabor-Heisenberg-Weyl Uncertainty Relation are the complex exponentials multiplied by Gaussians. These are sometimes referred to as “Gabor wavelets.”

\[ f(x) = e^{-i\omega x} e^{-(x-x_0)^2/\sigma^2} \]

Illustrations of the Bicubic Interpolator used for Huang’s Fruit Shape. Left panel: original image. Middle panel: interpolation from top left. The end result after 2D Gabor wavelet convolution: the image now has the modulated energy and the energy concentrated in the original (higher) image. The intuition behind the Fourier transform.