Filters and Pyramids

Wassily Kandinsky, "Accent in Pink"

CSC320: Introduction to Visual Computing
Michael Guerzhoy

Many slides from
Steve Marschner, Alexei Efros
Moving Average In 2D

What are the weights $H$?

$H[u, v]$
Reminder: Gradient

Horizontal Edge (absolute value)

Sobel

\[
\begin{array}{ccc}
1 & 2 & 1 \\
0 & 0 & 0 \\
-1 & -2 & -1 \\
\end{array}
\]
Reminder: Matching with filters

Goal: find in image

Method 0: filter the image with eye patch

What went wrong?

Input

Filtered Image

Side by Derek Hoiem
Cross-correlation filtering

- Let’s write this down as an equation. Assume the averaging window is \((2k+1) \times (2k+1)\):

\[
G[i, j] = \frac{1}{(2k + 1)^2} \sum_{u=-k}^{k} \sum_{v=-k}^{k} F[i + u, j + v]
\]

- We can generalize this idea by allowing different weights for different neighboring pixels:

\[
G[i, j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u, v] F[i + u, j + v]
\]

- This is called a \textbf{cross-correlation} operation and written:

\[
G = H \otimes F
\]

- \(H\) is called the “filter,” “kernel,” or “mask.”
Gaussian filtering

A Gaussian kernel gives less weight to pixels further from the center of the window

\[
h(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2 + v^2}{\sigma^2}}
\]

This kernel is an approximation of a Gaussian function.

Slide by Steve Seitz
Mean vs. Gaussian filtering
Mean vs. Gaussian filtering

box filter  gaussian
(Explanation on the blackboard)
Convolution

**cross-correlation:**  \[ G = H \otimes F \]

\[ G[i, j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u, v]F[i + u, j + v] \]

A **convolution** operation is a cross-correlation where the filter is flipped both horizontally and vertically before being applied to the image:

\[ G[i, j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u, v]F[i - u, j - v] \]

It is written:

\[ G = H * F \]

Suppose H is a Gaussian or mean kernel. How does convolution differ from cross-correlation?
Convolution is nice!

- Notation: \( b = c \ast a \)

- Convolution is a multiplication-like operation
  - commutative: \( a \ast b = b \ast a \)
  - associative: \( a \ast (b \ast c) = (a \ast b) \ast c \)
  - distributes over addition: \( a \ast (b + c) = a \ast b + a \ast c \)
  - scalars factor out: \( \alpha a \ast b = a \ast \alpha b = \alpha (a \ast b) \)
  - identity: unit impulse \( e = [\ldots, 0, 0, 1, 0, 0, \ldots] \)
    \[
    a \ast e = a
    \]

- Conceptually no distinction between filter and signal

- Usefulness of associativity
  - often apply several filters one after another: \( (((a \ast b_1) \ast b_2) \ast b_3) \)
  - this is equivalent to applying one filter: \( a \ast (b_1 \ast b_2 \ast b_3) \)
Gaussian filters

Remove “high-frequency” components from the image (low-pass filter)

• Images become more smooth

Convolution with self is another Gaussian

• So can smooth with small-width kernel, repeat, and get same result as larger-width kernel would have
• Convolving two times with Gaussian kernel of width $\sigma$ is same as convolving once with kernel of width $\sigma\sqrt{2}$

Source: K. Grauman
Gaussian filters at different scales

- Note: the camera (thin=high frequency) goes away if sigma is large

Source: Robert Collins
Practical matters

How big should the filter be?
Values at edges should be near zero
Rule of thumb for Gaussian: set filter half-width to about $3\,\sigma$
Practical matters

What about near the edge?

- the filter window falls off the edge of the image
- need to extrapolate
- methods:
  - clip filter (black)
  - wrap around
  - copy edge
  - reflect across edge

Source: S. Marschner
Image half-sizing

This image is too big to fit on the screen. How can we reduce it?

How to generate a half-sized version?
Image sub-sampling

Throw away every other row and column to create a $1/2$ size image - called *image sub-sampling*
Image sub-sampling

1/2  1/4 (2x zoom)  1/8 (4x zoom)

Aliasing! What do we do?
Gaussian (lowpass) pre-filtering

Solution: filter the image, *then* subsample

- Filter size should double for each $\frac{1}{2}$ size reduction. Why?

Slide by Steve Seitz
Subsampling with Gaussian pre-filtering

Gaussian 1/2

G 1/4

G 1/8

Slide by Steve Seitz
Compare with...

1/2

1/4  (2x zoom)

1/8  (4x zoom)

Slide by Steve Seitz
Gaussian (lowpass) pre-filtering

Solution: filter the image, \textit{then} subsample

- Filter size should double for each $\frac{1}{2}$ size reduction. Why?
- How can we speed this up?
Image Pyramids

Idea: Represent NxN image as a “pyramid” of 1x1, 2x2, 4x4,…, 2^k x 2^k images (assuming N=2^k)

Known as a **Gaussian Pyramid** [Burt and Adelson, 1983]
- In computer graphics, a *mip map* [Williams, 1983]
- A precursor to *wavelet transform*
A bar in the big images is a hair on the zebra’s nose; in smaller images, a stripe; in the smallest, the animal’s nose.

Figure from David Forsyth
What are they good for?

Improve Search

- Search over translations
  - Like project 1
  - Classic coarse-to-fine strategy
- Search over scale
  - Template matching
  - E.g. find a face at different scales

Pre-computation

- Need to access image at different blur levels
- Useful for texture mapping at different resolutions (called mip-mapping)
Gaussian pyramid construction

Repeat
• Filter
• Subsample

Until minimum resolution reached
• can specify desired number of levels (e.g., 3-level pyramid)

The whole pyramid is only $\frac{4}{3}$ the size of the original image!
Denoising

Additive Gaussian Noise

Gaussian Filter
Reducing Gaussian noise

Smoothing with larger standard deviations suppresses noise, but also blurs the image.

Source: S. Lazebnik
Reducing salt-and-pepper noise by Gaussian smoothing

3x3  

5x5

7x7
Alternative idea: Median filtering

A **median filter** operates over a window by selecting the median intensity in the window.

- Is median filtering linear?

Source: K. Grauman
Median filter

What advantage does median filtering have over Gaussian filtering?

- Robustness to outliers
Median filter

Salt-and-pepper noise

Median filtered

Source: M. Hebert
Median vs. Gaussian filtering

Gaussian

Median