CPSC 425: Computer Vision

Instructor: Jim Little
little@cs.ubc.ca

Department of Computer Science
University of British Columbia

Lecture Notes 2016/2017 Term 2
Menu January 17, 2017

Topics:
- Filter Examples: Gaussian, Pillbox
- Separability
- Non-linear Filters

Reading:
- Today: Forsyth & Ponce (2nd ed.) 4.1, 4.5
- Next: [Optional] Forsyth & Ponce (2nd ed.) 4.4

Reminders:
- Assignment 2 due Tuesday, January 24, start of lecture
- Bring your iClicker next class
- piazza: https://piazza.com/ubc.ca/winterterm2015/cpsc425/
Today’s ‘Fun’ Example

Technology Review (MIT) highlighted our work on using computer games to train networks to identify the elements of outdoor scenes.

Animation

Quote: “Hyper-realistic computer games may offer an efficient way to teach AI algorithms about the real world.”

We’ll see later how convolutional neural networks are composed of layers of learned filters.
The correlation of $F(X, Y)$ and $I(X, Y)$ is

$$I'(X, Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i, j) I(X + i, Y + j)$$

Visual interpretation: Superimpose the filter $F$ on the image $I$ at $(X,Y)$, perform an element-wise multiply, and sum up the values.

Convolution is like correlation except filter “flipped”
— When $F(-i, -j) = F(i, j)$ the two are equivalent.
Lecture 3: Re-cap

Example (we’ll do a convolution this time):
Lecture 3: Re-cap

- Ways to handle boundaries
  - Ignore/discard. Make the computation undefined for top/bottom $k$ rows and leftmost/rightmost $k$ columns.
  - Pad with zeros. Return zero whenever a value of $I$ is required beyond the image bounds.
  - Assume periodicity. Top row wraps around to the bottom row; leftmost column wraps around to rightmost column.

- Simple examples of filtering:
  - copy, shift
  - smoothing
  - sharpening

- Characterization Theorem: Any linear, shift-invariant operation can be expressed as a convolution
Example 1 (cont’d): Smoothing with a Gaussian

Idea: Weigh contributions of neighbouring pixels by nearness

2D Gaussian (continuous case):

\[ G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp\left( -\frac{x^2 + y^2}{2\sigma^2} \right) \]

Forsyth & Ponce (2nd ed.)
Figure 4.2
Example 1 (cont’d): Smoothing with a Gaussian

Forsyth & Ponce (2nd ed.) Figure 4.1 (left & right)
Gaussian: Area Under the Curve

- 68% within ±1σ
- 95% within ±2σ
- 99.7% within ±3σ
- 99.99% within ±4σ
Efficient Implementation: Separability

- A 2D function of $x$ and $y$ is **separable** if it can be written as the product of two functions, one a function only of $x$ and the other a function only of $y$.

- Both the 2D box filter and the 2D Gaussian filter are separable.

- Both can be implemented as two 1D convolutions:
  - First, convolve each row with a 1D filter
  - Then, convolve each column with a 1D filter
  - Aside: or *vice versa*

- The 2D Gaussian is the only (non trivial) 2D function that is both separable and rotationally invariant.
Efficient Implementation: Separability (cont’d)

For example, recall the 2D Gaussian

\[ G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp \left( -\frac{x^2 + y^2}{2\sigma^2} \right) \]

\[ = \left( \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{x^2}{2\sigma^2} \right) \right) \left( \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{y^2}{2\sigma^2} \right) \right) \]

The 2D Gaussian can be expressed as the product of two functions, one a function of \( x \) and the other a function of \( y \)

In this case, the two functions are the (identical) 1D Gaussian
Example 2: Smoothing with a Pillbox

Let the radius (i.e., half diameter) of the filter be $r$

In the continuous domain, a 2D (circular) pillbox filter, $f(x, y)$, is defined as

$$f(x, y) = \frac{1}{\pi r^2} \begin{cases} 
1 & \text{if } x^2 + y^2 \leq r^2 \\
0 & \text{otherwise}
\end{cases}$$

The scaling constant, $\frac{1}{\pi r^2}$, ensures that the area of the filter is one
Example 2: Smoothing with a Pillbox

- Recall that the 2D Gaussian is the only (non trivial) 2D function that is both separable and rotationally invariant.

- A 2D pillbox is rotationally invariant but not separable.

- There are occasions when we want to convolve an image with a 2D pillbox. Thus, it worth exploring possibilities for efficient implementation.
Example 2: Smoothing with a Pillbox

- A 2D box filter can be expressed as the sum of a 2D pillbox and some “extra corner bits”

![Diagram showing the sum of a 2D box filter as the sum of a 2D pillbox and extra corner bits]
Example 2: Smoothing with a Pillbox

Therefore, a 2D pillbox filter can be expressed as the difference of a 2D box filter and those same “extra corner bits”
Example 2: Smoothing with a Pillbox

Implementing convolution with a 2D pillbox filter as the difference between convolution with a box filter and convolution with the “extra corner bits” filter allows us to take advantage of the separability of a box filter.

Further, we can postpone scaling the output to a single, final step so that convolution involves filters containing all 0’s and 1’s — This means the required convolutions can be implemented without any multiplication at all.
Example 2: Smoothing with a Pillbox

original 11 × 11 pillbox
Aside: Speeding Up Convolution (The Convolution Theorem)

Let $z$ be the product of two numbers, $x$ and $y$. That is,

$$z = xy$$

Taking logarithms of both sides, one obtains

$$\ln z = \ln x + \ln y$$

Therefore,

$$z = e^{\ln z} = e^{(\ln x + \ln y)}$$

**Interpretation:** At the expense of two $\ln()$ computations and one $\exp()$ computation, multiplication is reduced to addition.
Aside: Speeding Up Convolution (The Convolution Theorem)

Another analogy: 2D rotation of a point by an angle $\alpha$ about the origin

The standard approach, in Euclidean coordinates, involves a matrix multiplication

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Suppose we transform to polar coordinates

$$(x, y) \rightarrow (\rho, \theta) \rightarrow (\rho, \theta + \alpha) \rightarrow (x', y')$$

Rotation becomes addition, at the expense of one polar coordinate transform and one inverse polar coordinate transform
Aside: Speeding Up Convolution (The Convolution Theorem)

Similarly, some image processing operations become cheaper in a transform domain

Image credit: Gonzalez and Woods (3rd ed.) Fig. 2.39
Aside: Speeding Up Convolution (The Convolution Theorem)

Convolution Theorem:

Let \( i'(x, y) = f(x, y) \otimes i(x, y) \)

then \( \mathcal{I}'(\omega_x, \omega_y) = \mathcal{F}(\omega_x, \omega_y) \mathcal{I}(\omega_x, \omega_y) \)

where \( \mathcal{I}'(\omega_x, \omega_y) \), \( \mathcal{F}(\omega_x, \omega_y) \), and \( \mathcal{I}(\omega_x, \omega_y) \) are the Fourier transforms of \( i'(x, y) \), \( f(x, y) \), and \( i(x, y) \)

At the expense of two Fourier transforms and one inverse Fourier transform, convolution can be reduced to (complex) multiplication
Let $\otimes$ denote convolution. Let $I(X, Y)$ be a digital image. Let $F$ and $G$ be digital filters.

- Convolution is **associative**. That is,

$$G \otimes (F \otimes I(X, Y)) = (G \otimes F) \otimes I(X, Y)$$

- Convolution is **symmetric**. That is,

$$(F \otimes G) \otimes I(X, Y) = (G \otimes F) \otimes I(X, Y)$$

Convolving $I(X, Y)$ with filter $F$ and then convolving the result with filter $G$ can be achieved in a single step, namely convolving $I(X, Y)$ with filter $G \otimes F = G \otimes F$.
Example 3:

\[ \frac{1}{9} \begin{array}{ccc} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array} \times \frac{1}{9} \begin{array}{ccc} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array} = \begin{array}{ccc} & & \\ & & \\ & & \end{array} \]

3 \times 3 \quad 3 \times 3 \quad 5 \times 5

Box \quad Box
Example 3:

\[
\frac{1}{9} \begin{array}{ccc}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{array} \quad \times \quad \frac{1}{9} \begin{array}{ccc}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{array} = \frac{1}{81} \begin{array}{cccc}
1 & 2 & 3 & 2 & 1 \\
2 & 4 & 6 & 4 & 2 \\
3 & 6 & 9 & 6 & 3 \\
2 & 4 & 6 & 4 & 2 \\
1 & 2 & 3 & 2 & 1 \\
\end{array}
\]

3 × 3 Box

5 × 5
Gaussian: An Additional Property

Let $\otimes$ denote convolution. Let $G_{\sigma_1}(x)$ and $G_{\sigma_2}(x)$ be two 1D Gaussians

$$G_{\sigma_1}(x) \otimes G_{\sigma_2}(x) \equiv G_{\sqrt{\sigma_1^2 + \sigma_2^2}}(x)$$

Convolution of a Gaussian with a Gaussian is another Gaussian

**Special case:** Convolution with $G_{\sigma}(x)$ twice is equivalent to convolution with $G_{\sqrt{2}\sigma}(x)$
Non-linear Filters

- We’ve seen that linear filters can perform a variety of image transformations
  — shifting
  — smoothing
  — sharpening

- In some applications, better performance can be obtained by using non-linear filters.

- For example, the median filter selects the median value from each pixel’s neighbourhood.
Median Filter

Example:
Median Filter

- Effective at reducing certain kinds of noise, such as impulse noise (a.k.a ‘salt and pepper’ noise or ’shot’ noise)
- The median filter forces points with distinct values to be more like their neighbors

Image credit: Gonzalez and Woods (3rd ed.) Fig. 3.35
Non-linear Filters

Video: Spotlight on the rolling guidance filter
Summary

- We covered two additional linear filters: Gaussian, pillbox
- Separability (of a 2D filter) allows for more efficient implementation (as two 1D filters)
- Convolution is associative and symmetric
- Convolution of a Gaussian with a Gaussian is another Gaussian
- The median filter is a non-linear filter that selects the median in the neighbourhood
Reminders:

**Assignment 2** due start of lecture on Tuesday, January 24
We’ll have some iClicker exercises next lecture


**piazza:** [https://piazza.com/ubc.ca/winterterm22015/cpsc425/](https://piazza.com/ubc.ca/winterterm22015/cpsc425/)