Image Filtering

COSC342

Lecture 3
7 Mar 2017

In this Lecture

- Image filters
- Kernels and convolution
- Blurring – mean filter, Gaussian filter
- Edge detection – Sobel filters
- Separable filters

Image Filters

- A filter takes an image and applies some operation at every pixel
  - The same operation is applied at all pixels
  - This operation produces a new value for each pixel
  - Often this is viewed as an image itself, which is the result of the filter
- We will consider just greyscale images
  - Can apply filters independently to RGB colour channels
  - In HSV etc, might just apply filters to the intensity channel
- Many filters are applied in a process called *convolution*

Image Convolution

- We have a kernel, K, which is a $2k+1 \times 2k+1$ matrix of numbers
- This is applied at each point, (x, y), in an image, I as follows:

$$(I * K)(x, y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} K(i, j) I(x-i, y-j)$$

- This is called discrete *convolution*
  - The kernel moves over the image, and is applied to its centre pixel
  - We multiply corresponding kernel and image entries
  - The result is the sum of these products

Image Convolution

1	1	1	2	4	5	6	7
1	3	2	4	5	7	7	8
2	1	3	5	6	6	8	8
3	2	1	3	2	6	7	8
4	3	2	3	1	4	6	7
5	4	4	5	4	2	3	6
5	6	7	8	7	6	5	6
7	8	8	8	8	8	7	5

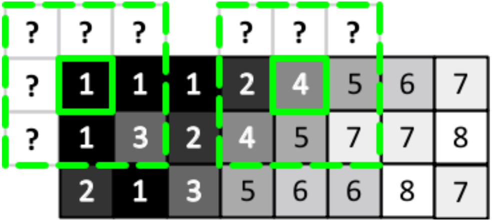
0	1/8	0
1/8	1/2	1/8
0	1/8	0

$$(I * K)(5,4) = (0 \times 5) + \left(\frac{1}{8} \times 6\right) + (0 \times 6) + \left(\frac{1}{8} \times 3\right) + \left(\frac{1}{2} \times 2\right) + \dots$$

$$= \frac{0 + 6 + 0 + 3 + 8 + 6 + 0 + 3 + 0}{8} = \frac{24}{8} = 3$$

Convolution Problems

- Some filters give negative outputs, fractions, or go out of range
  - Can round values, or change type of image data, etc.
  - Can use mid-grey as zero, white as positive, black as negative
- All filters have problems at the edges of images



  - Can ignore the edges (shrinking the output image a bit)
  - Can predict the values (nearest neighbour, linear models, etc.)
  - Can tile or mirror the image

Blurring Filters: Mean Filters

- Mean filters take a simple average over a window
- There is a family of them depending on size – 3×3, 5×5, etc.

$$M_{3\times 3} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad M_{5\times 5} = \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- These blur and smooth the image

Mean Filters

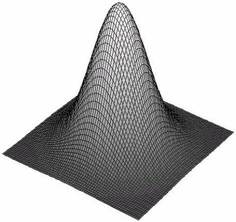
$*M_{3\times 3} =$

$*M_{7\times 7} =$

More Blurring: Gaussian Filters

- Mean filters treat all pixels equally
- There are good reasons to give more weight to pixels nearer the centre
- A common way to do this is with Gaussian filters

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



- The parameter σ determines how much blurring we get
  - Beyond about 3σ the values are very close to zero
  - So we can use a roughly $6\sigma \times 6\sigma$ kernel
  - Need to normalise kernel values so that they add to 1

Gaussian Filters

$* G_{\sigma=1} =$

$* G_{\sigma=3} =$

Edge Detection: Sobel Filters

- Another common use of filters is in *edge detection*
- An edge in an image is a place where the intensity changes quickly
- The Sobel filters detect edges in the x- and y-directions:

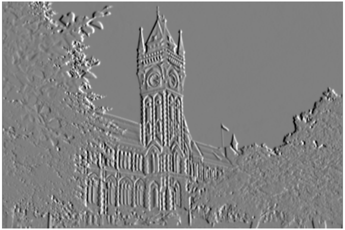
$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- These can be combined to make a gradient vector, $\mathbf{g} = \begin{bmatrix} S_x \\ S_y \end{bmatrix}$

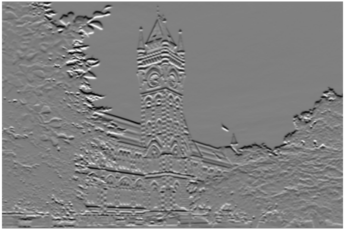
Sobel Filters

I

$I * S_x$



$I * S_y$



$|\mathbf{g}|$

Separable Filters

- If we have an image with n pixels, applying a $k \times k$ filter is $O(nk^2)$
- This can get expensive with large filters
  - $G_{\sigma=3}$, for example is 19×19 – 361 multiplies and additions per pixel
- Many 2D filters can be separated into two 1D filters
  - First we apply a $1 \times k$ filter, F_x, to each pixel
  - Next we apply a $k \times 1$ filter, F_y, to each pixel
  - This has the same effect as applying the 2D filter $(F_y F_x)$

Separable Filters

$$M_{3\times 3} = \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix} = \left(\frac{1}{3}\begin{bmatrix}1\\1\\1\end{bmatrix}\right)\left(\frac{1}{3}\begin{bmatrix}1 & 1 & 1\end{bmatrix}\right)$$

$$G_{\sigma=\frac{1}{2}} \approx \begin{bmatrix}0.04 & 0.12 & 0.04\\0.12 & 0.36 & 0.12\\0.04 & 0.12 & 0.04\end{bmatrix} = \begin{bmatrix}0.2\\0.6\\0.2\end{bmatrix}\begin{bmatrix}0.2 & 0.6 & 0.2\end{bmatrix}$$

$$S_x = \begin{bmatrix}-1 & 0 & 1\\-2 & 0 & 2\\-1 & 0 & 1\end{bmatrix} = \begin{bmatrix}1\\2\\1\end{bmatrix}\begin{bmatrix}-1 & 0 & 1\end{bmatrix}$$

Other Filters

- Sharpening filter

$$* \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} =$$

- Emboss filter

$$* \begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} =$$

- Are these separable?

The Median Filter

- Not all filters can be implemented as convolution
- A common filter is the Median filter:
  - Sort all of the pixel values in a $k \times k$ window
  - Choose the middle value as the result

$\underset{5\times5}{\longrightarrow}$

Tutorials and Labs

- This week's tutorial:
  - Matrix and vector mathematics
  - Have a go *before the tutorial*
  - Come along if you have any questions or issues
- Monday's Lab:
  - Matrices in C++ / OpenCV introduction