Computer Vision
Homework 06: Image Filtering
CS 670, Fall 2019

Name: Kunjal Panchal
Student ID: 32126469
Email: kpanchal@umass.edu

Oct 17, 2019

Contents

1 Problem 1: Separable 2D filter kernel equivalence with two 1D filter kernels **3**
1.1 Definitions 3
1.2 1D and 2D Kernels 3
1.3 Why Separable Kernels have Equivalence between their 2D form and two 1D form counterparts 4
1.3.1 Convolution Properties 4
1.4 Conclusions 5

2 Problem 2: Convolution with Self is Another Gaussian **6**
2.1 (a)Number of Iterations and Reasonable Approximation 6
2.2 (b)Benefits 8
2.3 Conclusion 9

1 Problem 1: Separable 2D filter kernel equivalence with two 1D filter kernels

1.1 Definitions

- FILTER KERNEL is a small matrix. It is used for blurring, sharpening, embossing, edge detection, and more. This is accomplished by doing a convolution between a kernel and an image.
- CONVOLUTION is the process of adding each element of the image to its local neighbors, weighted by the kernel.
- SEPARABLE KERNELS To reduce the cost of computing the convolution; a filter can be written as product of two more simple filters.

  Typically a 2-dimensional convolution operation is separated into two 1-dimensional filters.

1.2 1D and 2D Kernels

Suppose we have a 1D, 3 x 1 filter kernel: $\begin{bmatrix}1 & 2 & 1\end{bmatrix}$

We can take this row vector/ matrix and it's transpose, which will be a 1 x 3 column vector matrix; multiply that row vector with it's transpose to get a 3 x 3 matrix, which will be our 2D filter kernel:

$$\begin{bmatrix}1\\2\\1\end{bmatrix} \cdot \begin{bmatrix}1 & 2 & 1\end{bmatrix} = \begin{bmatrix}1 & 2 & 1\\2 & 4 & 2\\1 & 2 & 1\end{bmatrix}$$

The above calculations show simple 3 x 1 and 1 x 3 matrix multiplication that results in a 2D 3 x 3 matrix.

So, a 2D filter can be factored into two 1D filters.

In case of a 2D filter kernels of size m X m; an image of n x n getting filtered would have complexity of $O(n^2m^2)$

But in case of a 1D filter kernel of size m x 1; an image of n x n getting filtered would only have complexity of $O(n^2m)$, because of the internally performed **two 1D convolutions (one along rows, one along columns)**

rather than one convolution with a 2D matrix. That saves a lot more space and time. [1]

1.3 Why Separable Kernels have Equivalence between their 2D form and two 1D form counterparts

A filter kernel is said to be separable if it can be broken into two one-dimensional signals: a vertical and a horizontal projection.

Specifically, the value of each pixel in the image is equal to the corresponding point in the horizontal projection multiplied by the corresponding point in the vertical projection.

The kernels where all rows are scaled versions of the other rows are separable. That is, each row i must be of the form:

$$r[i][j] = a[i] * b[j]$$

where b is the "model row", and $a[i]$ is the scaling for each row. This looks kind of obvious, since the multiplication above is what you get when you convolve the column kernel a with the row kernel b.

1.3.1 Convolution Properties

Convolution as one useful property of DISTRIBUTION OVER ADDITION:

$$filter * (img\_horizontal + img\_vertical)$$

$$= (filter * img\_horizontal) + (filter * img\_vertical)$$

To convolve an image with a separable filter kernel, convolve each row in the image with the horizontal projection, resulting in an intermediate image.

Next, convolve each column of this intermediate image with the vertical projection of the kernel.

The resulting image is identical to the direct convolution of the original image and the filter kernel.

Convolving the columns first and then the rows; will give same result.

NOTE: To know if a 2D kernel is separable, its rank must be 1. This indicates that all rows are scaled versions of each other.

1.4 Conclusions

A separable 2D filter kernel is equivalent to filtering it wit two 1D filter kernels because of the convolution property, which says;

Convoluting first rows/columns and then the columns/rows is equivalent to convoluting a 2D grid of rows and columns, as convolution is just a scalar multiplications and summations. See Figure 1

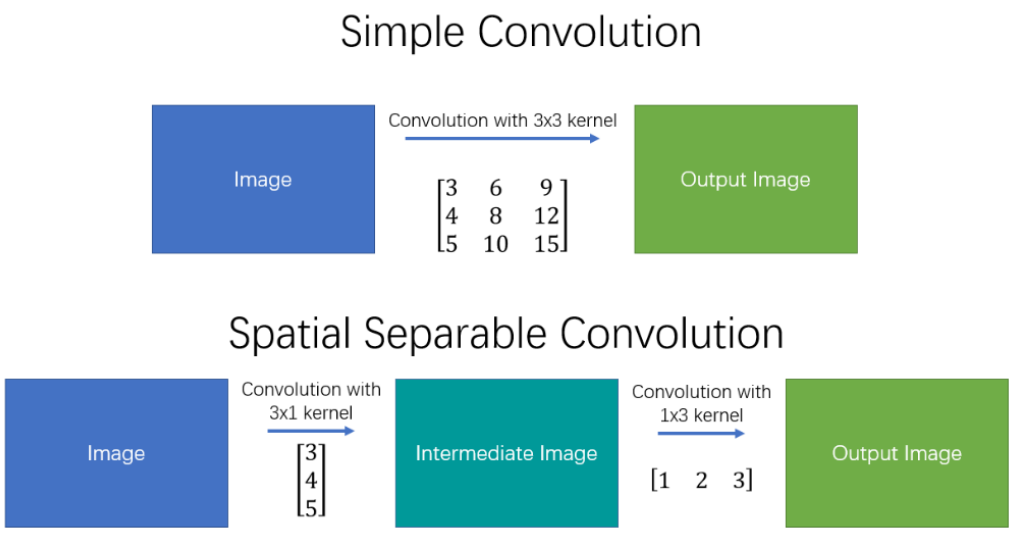

Figure 1: Simple and Spatial Separable Convolution [2]

2 Problem 2: Convolution with Self is Another Gaussian

In this section, we will compare an image filtered with gaussian filter kernel with $\sigma = 2$ and $\sigma = 3$, both, against an image filtered with gaussian filter kernel with $\sigma = 0.5$ applied to the latest filtered intermediated result, k times.

We will determine the k value of iterations:

2.1 (a) Number of Iterations and Reasonable Approximation

First, we specify few of the assumptions we made:

- To measure dissimilarity, we are using sum of squared difference measure.
- The smallest SSD will be our "most reasonable" approximation.
- To find the most reasonable approximation, we iterate more than the optimal times, to confirm that it is really the most reasonable approximation.

Now, we kept the value of σ for filter kernel which will be self-convoluted, constant at $\sigma = 0.5$.

We tried less than 0.5 but the difference between original and blurred image was so small that the deductions were hard, if not impossible, to make.

The results are shown in Figures 2 and 3.

In case of a convolution with self for a filter kernel with $\sigma = 0.5$, used for 30 iterations; comparing to a filter kernel with $\sigma = 2$ which is used only one time: we get the most reasonable approximation, with SSD of only 0.2300 on 17^{th} iteration. After that, the dissimilarity started to increase.

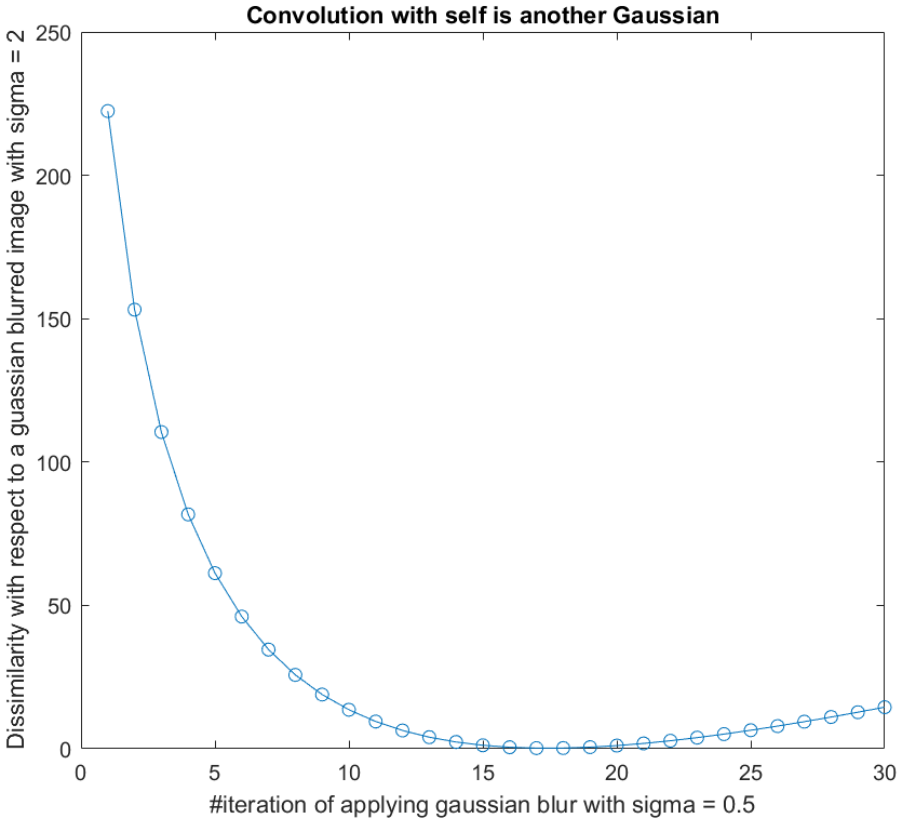

Figure 2: Convolution with self for a filter kernel with $\sigma = 0.5$, used for 30 iterations; comparing to a filter kernel with $\sigma = 2$ which is used only one time

In case of a convolution with self for a filter kernel with $\sigma = 0.5$, used for 50 iterations; comparing to a filter kernel with $\sigma = 3$ which is used only one time: we get the most reasonable approximation, with SSD of only 0.2682 on 38^{th} iteration. After that, the dissimilarity started to increase.

From these two results, we can conclude that:

- As we increase σ, we need to increase the iterations of convolutions for the smaller σ to make it reasonably the same filter.

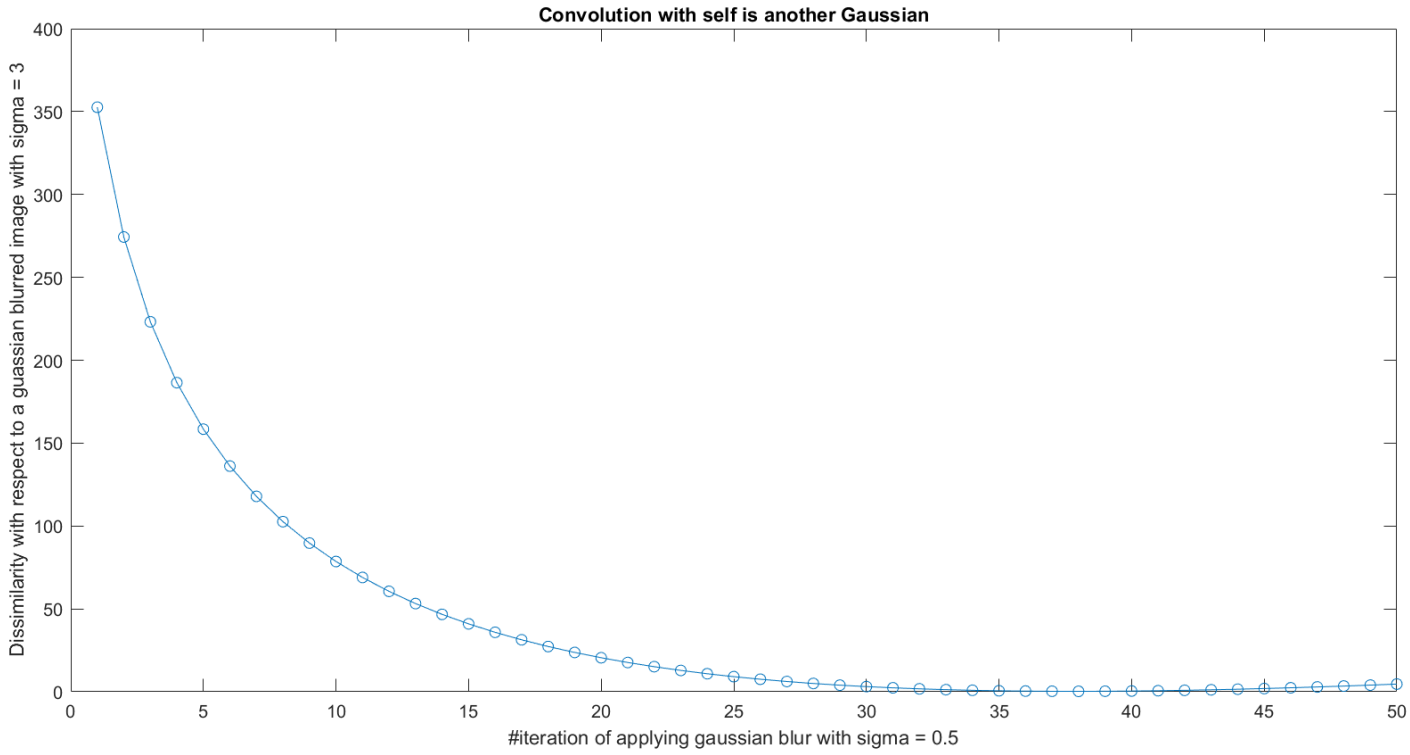

Figure 3: Convolution with self for a filter kernel with $\sigma = 0.5$, used 50 iterations; comparing to a filter kernel with $\sigma = 3$ which is used only one time

- We can indeed, create approximately similar filter displaying higher standard deviation, from convolving small standard deviated filter kernels with themselves, k number of times.

2.2 (b) Benefits

Gaussian filters are good with images with noise which has a small standard deviation. That means, most of the times, a gaussian filter having small σ is much more useful than one with a large σ.

But sometimes, one pass of filter might not be enough, in that case, we can just convolute the image with the noise till it is actually closer to the noise-less image/ original image.

Taking/Starting with a larger σ will not work as it might not act appropriately in case of small standard deviated noises. So, it is beneficial to start and stay with a small σ and just convolute it k times till we get satisfactory results.

It will give the same result as an ideal σ for that particular image would have given.

Also, smoothing with large σ (standard deviation) suppresses noise, but also blurs the image, if not chosen the ideal σ.

With self-convolution, what emerges are filters that are translationally equivariant.

THUS, if we don't know how much deviated the noise in the image is, we cannot use a large σ as it might "overdo" the filtering and blur the image. So, we start with small σ and self-convolute it till we are near the "ideal" or "approximately reasonable" σ which will give us denoised image very near to the actual noise-less image.

2.3 Conclusion

If we define convolution to be:

$$(f1 * f2)(t) = \int_{-\infty}^{\infty} f1(t) f2(\tau - t) d\tau$$

For many functions f, iterated self convolution $(f * f * ... * f)(t)$ tends to create functions increasingly close to Gaussian - closer as the number of convolutions increase:

$$g(x) = Ke^{-\frac{(x-\mu)^2}{\sigma^2}}$$

We may just rescale f in order to make it a probability density function. If the hypothesis of the central limit theorem are fulfilled, the claim is straightforward. Original proof involves Fourier Series Transforms.

References

[1] Subhransu Maji, *Linear Filtering* https://www.dropbox.com/s/tfsmuiunffyu7ml/lec09_linear_filtering.pdf?dl=0

[2] Chi-Feng Wang, *A Basic Introduction to Separable Convolutions* https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728