Normalized Convolution In Restoration

Research Paper · December 2022

Joanikij Chulev, Martin Kroft and Bea Ribeiro de Freitas

Utrecht University,
University College Roosevelt,
Middelburg, The Netherlands

E-mail: j.chulev@ucr.nl, m.kroft@ucr.nl, b.ribeirodefreitas@ucr.nl

Abstract

This paper shows in detail an approach to signal and image reconditioning to a certain extent where the image or signal were deemed to be restorable. We aimed to represent how normalized convolution would work to return images to their original state. This was all done virtually. We did this using a 1-Dimensional signal function, a 2-Dimension matrix function, with an image layout and finally on a high-resolution image, whilst using a Gaussian Filter. We then investigated how much of an image could be removed before it would no longer be possible to restore the original signal.

Contents

1. Introduction

As AI and technology progress, the world is getting more complicated along with technological approaches. Computer vision is a part of the AI world that tackles with images, with retrieving and recognizing them.

All multi-level signal processing systems have an essential challenge with information representation. The analysis of irregularly collected data is advantageous to or necessary for many scientific disciplines of study. However, it is more difficult to analyze data with irregular sampling than with regular spacing or sampling. Reconstructing the erratically sampled signal or resampling it into a regular grid is frequently necessary and often gives better results. The issue of image analysis on irregularly sampled picture data is taken into account in the article of Knutsson and Westin [3] with different experiments.

One of their findings, normalized convolution, is a general method for filtering incomplete or uncertain data and is based on the separation of both data and operator into a signal part and a certainty part [5]. Thus, the convolution can be made more effective by a normalization operation that takes into account the possibility of missing samples [3].

2. Overview of Normalized Convolution

The formula we used in order to normally convolve the signals chosen is:

$$\tilde{x} = \frac{F * (c \cdot x)}{F * c}$$

where $\tilde{x}$ is the approximation of the restored data, F is a filter that normalizes the data, c is the certainty of the data, and x is the data itself. In other words, the sampled data is first written as a product of the certainty of that data with the evaluated point, then convolved with a normalizing filter, and divided by the convolution of said filter with the certainty of the data.

We wanted to investigate how close a restored signal would be to the original signal, so we always started with a fully intact signal, and then removed data ourselves to represent the sampled signal. To do this, the data is multiplied by a map of 0's and 1 s, where multiplying by 0 means that that specific point in time was not sampled, as opposed to being multiplied by 1.

As such, the map of 0's and 1's also already represented the confidence of each point, since if a point was multiplied by 0, or erased, then there wouldn't be certainty in that measurement, since it means that it wasn't even measured in the first place. Therefore, after simulating the sampling of the signal, all that is left is just convolving with the filter and dividing by the convolution of that filter with the map of 0's and 1's.

3. Gaussian Filter

In this paper, the way we have chosen to normalize data is through the use of a Gaussian filter. The Gaussian Filter is one of the most widely used filtering algorithms [6]. The Gaussian Filter represents the belief of the current state by a Gaussian distribution, whose mean is an affine function of the measurement [6]. In statistics, bell curves—curves more formally known as normal distributions or Gaussian distributions are a type of continuous probability distribution for a real-valued random variable [4]. To finally add, an affine function is a linear function plus a translation or offset [1].

We will take the following formula as point of departure for defining a Gaussian signal:

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right)$$

Since the objective is to convolve using this filter, for a given point we want it to taken into account the points in its immediate proximity, so we will take μ to be 0, such that the Gaussian function will be defined as:

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right)$$

4. Applications

The main goal was to take different signals and or images with damaged quality due to the certainty filter, implement the Gaussian filter and try to revert original image quality back to or close to the original standard.

This was done in the Wolfram Mathematica workspace environment [2], under version 13.1 of the Mathematical application (most up-to-date at that time). Mathematica proves to be an essential tool for these types of representations and problems due to its simple syntax with built-in functions as well as support for mathematical nomenclature.

4.1 1-D Restoration

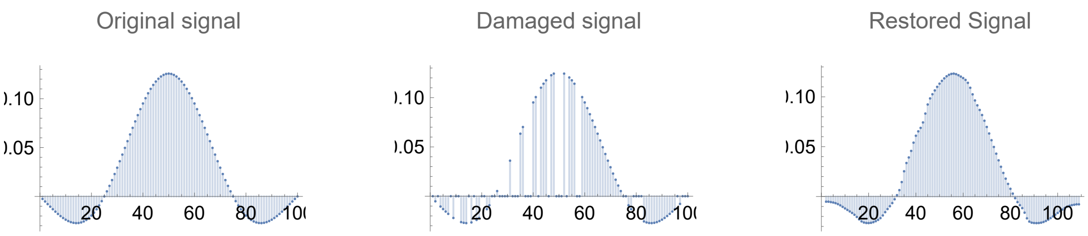
Signal restoration is best explained starting in a single dimension. For example, a wave function plot will be used, namely:

$$x[n] = \frac{\sin\left(\frac{2\pi}{50}(n-50)\right)}{n-50}$$

This wave is first generated in Mathematica, and then at random an amount of signal values is removed to acquire the 'damaged signal'. The challenge is now how to restore the original signal as best as possible. To this end an estimate for the missing values must be calculated, using the methodology described in section 2.

The damaged cosine displays sharp changes in the signal value. The Gaussian filter is intended to smooth out these sudden shifts. It does so by taking the adjacent signal values and applying a weight to them. The highest weight is applied to directly adjacent values, and farther values are given a lower weight. The sum of all weights adds up to 1, and so including more signal values in the filter (i.e. making the filter broader) will reduce the individual weights. This results in more severe smoothing of the output function, as seen in the restored signal.

Figure 1: Results of trying to restore a signal with approximately 50% of its data removed



However, it is notable that the restored signal seems to be wider ($n \approx 110$) than the original signal ($n = 100$). This is likely due to the nature of convolution, since convolving with a signal with more than one data point (as is the case with a Gaussian Filter) results in a wider signal. This then means that it is quite difficult to compare the original and restored signal numerically, although graphically they do look quite similar.

4.2 2-D Restoration

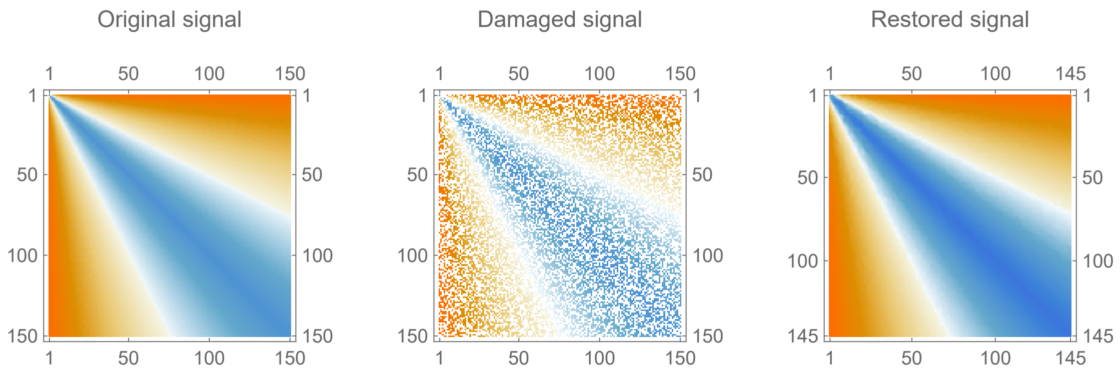
There were two different ways we investigated 2-dimensional normalized convolution, although both are quite similar. However, the approaches used for each one differed due to the nature of the Mathematica program we used. These two approaches were through matrices, and through images.

4.2.1 Using matrices

The restoration of 2-dimensional data is similar to that of the 1-dimensional wave. The key difference is that the signal and filter are now both matrices. By convolving the signal and the filter matrices, each cell of the output matrix becomes the weighted sum of the adjacent signal values of the respective cell in the input matrix.

To make the experiment more illustrative, the input and output matrices are colour-coded. That is, the numerical value in each cell of the matrix is converted into a colour, with high numbers becoming a warmer shade (oranges) and low values becoming a cooler shade (blues). Similar to the 1-dimensional experiment, each image is now damaged by randomly removing pixels, (or setting matrix cells to zero.) This damaged image is now convolved with a Gaussian filter, and then divided by the confidence map convolved with the Gaussian filter.

Figure 2: Results of trying to restore a matrix with approximately 50% of its data removed



We also implemented a manipulate plot that allows the sigma value of the Gaussian to be changed, as well as the amount of the original signal that is preserved. After investigation, we found that increasing σ results in the edges getting more blurred. This can easily be explained by the fact that an increased σ-value results in a wider Gaussian Filter, which means that the convolution results in each value of the matrix being weighed with more points, that are further away. This essentially results in the values being more averaged out.

As for changing the proportion of the original matrix preserved, as little as 20% of the original matrix can be preserved in order for the restored signal to be similar to the original one. However, at this point, Mathematica might start giving errors, appearing as "holes" in the matrix plot. This is likely due to the fact that the convolution of the Gaussian filter with the confidence map yields 0 at those points, such that those values are being divided by 0, resulting in undefined values.

4.2.2 Using images

The same processed used for matrices was applied to images in Mathematica. Just like in the previous examples, removing approximately 50% of the values results in a restored signal that is still quite similar to the original signal.

Figure 3: Results of trying to restore an image with approximately 50% of its data removed

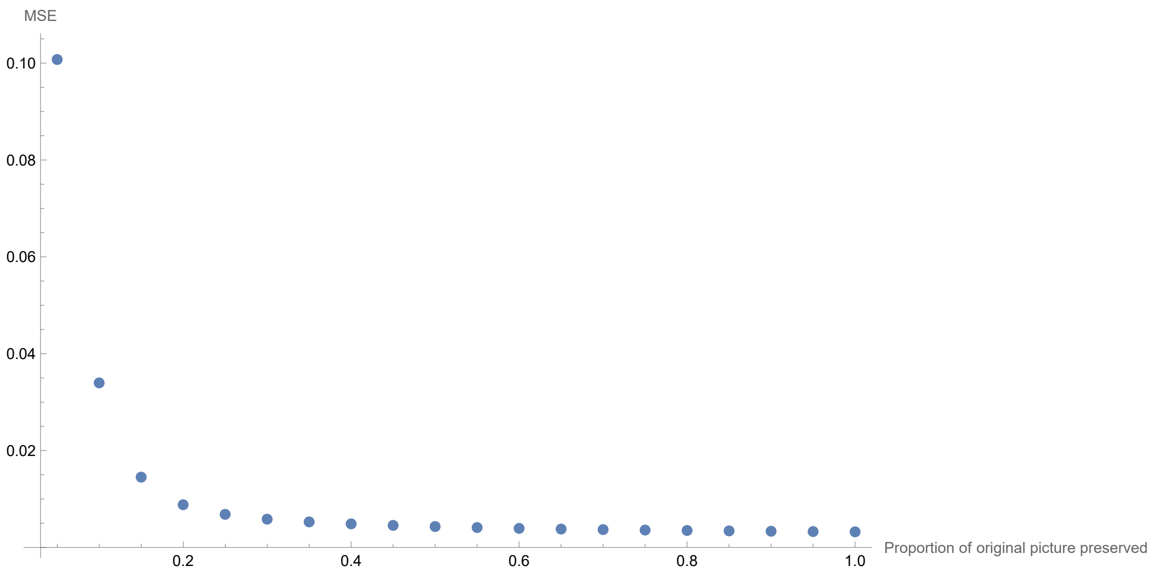
Additionally, in this example we also investigated the mean squared error between the original image and the restored one, since this method of using images in Mathematica is the only one that yields a restored signal of the same dimensions as the original signal.

The following formula for calculating mean squared error was adapted in order to apply to a 2-dimensional plane:

$$\frac{1}{n}\sum_{i=1}^{n}\left(Y_i-\hat{Y}_i\right)^2 \quad \Rightarrow \quad \frac{1}{\alpha\cdot\beta}\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta}\left(Y_{i,j}-\hat{Y}_{i,j}\right)^2$$

where α and β are the dimensions of the image. After plotting the MSE against the proportion of original image that is kept, it can be observed that there appears to be an inverse relation between MSE and the proportion of original image kept, such that, starting from 0.3, MSE seems to be approaching a value. This would mean that removing up to 70% of the original image should result in little to no changes from the original pictures. In fact, at around 0.2, the value of MSE is still quite similar to the value it approximates as the proportion increases, which leads us to believe that removing 80% of the original image will still result in a similar restored signal.

Figure 4: Mean Squared Error (MSE) of restored image compared to the original signal plotted against the proportion of the original picture that is preserved



5. Conclusions

A commonly used operation in image processing is convolution [3]. In general the reverted signals were significantly close to the originals, even at damage ratios of up to 80%. This has clearly demonstrated the effectiveness of normalized convolution for purposes of image restoration. Conversely, this result has major implications for data compression. As 80% of matrix data can be removed in order to still be able to generate a reasonable approximation of the original matrix, only 20% of data needs to be stored.

Acknowledgments

We express great gratitude to professor Dr. Ir. Richard van den Doel for reviewing this paper and aiding us in the explanations of the previously discussed topics above.

References

[1] Industrial organization, spring, 2010.

[2] Wolfram mathematica: Modern technical computing.

[3] H. Knutsson and C.-F. Westin. Normalized and differential convolution. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 515–523. IEEE Comput. Soc. Press, 1993.

[4] Aidan Lyon. Why are normal distributions normal? 65(3):621–649, 2014.

[5] C.-F. Westin, K. Nordberg, and H. Knutsson. On the equivalence of normalized convolution and normalized differential convolution. In *Proceedings of ICASSP '94. IEEE International Conference on Acoustics, Speech and Signal Processing*, volume v, pages V/457–V/460. IEEE, 1994.

[6] Manuel Wüthrich, Sebastian Trimpe, Cristina Garcia Cifuentes, Daniel Kappler, and Stefan Schaal. A new perspective and extension of the gaussian filter. 35(14):1731–1749, 2016.