Image Processing: Deblurring and Denoising

Omotoyosi M. Taiwo, Fisk University
Advisors: Daniela Calvetti, PhD, Erkki Somersalo, PhD, Charles Joseph.
Department of Mathematics, Case Western Reserve University, Cleveland, OH 44106

Abstract

Images have an important role in scientific research and life in general. They help store data and memories. The great increase in image definition in recent years has made them larger in size, thus creating a challenge when it comes to computational aspects of images. The process of collecting and analyzing images must take into account the noise corrupting the specimens Noise in images may come from the measuring device, conditions of the surrounding environment or round off in calculations carried out in preprocessing steps. Small noise may cause unwanted variations in brightness and may be greatly amplified to the point of dominating the processed image. There are mathematical methods for removing noise from images and limiting their deleterious effects. The minimization of the loss of image detail in the process of limiting the effect of the noise is an important topic of research. In our project we use three methods for the restoration of a black and white image: truncated SVD and truncated Fourier transforms for denoising and a filtered deconvolution for deblurring and denoising. All three methods are regarded as matrix operatiions on matrices representing images. We investigate the effectiveness and limitations of the methods by testing with different levels of additive Gaussian noise in the images and with various Gaussian blurrings. The results show that these methods successfully removed noise at the price of having the image lose some definition. The higher the noise level, the more definition had to be removed from the denoised image. Future directions of this project will include studying how to keep as much detail as possible in restored images.

Singular Value Decomposition

From images to matrices

•Any rectangular image can be subdivided into *m X n* pixels. The larger the values of m and n, the higher the resolution. This naturally defines a matrix A.
•Each pixel location is assumed to have constant light intensity. The (i,j)th entry of A is the light intensity at the (i,j)th pixel. Thus A is a representation of the image.
•All matrix entries are nonnegative.
•The Singular Value Decomposition (SVD) of a matrix A is a factorization

$$\mathbf{A = U\,\Sigma\,V^*} = \sigma_1 u_1 * v_1^{\,T} + \sigma_2 u_2 * v_2^{\,T} + \ldots + \sigma_r u_r * v_r^{\,T}$$

where U_{mxn} and V_{nxn} are orthogonal matrices and **Σ** is a diagonal matrix with nonzero decreasing diagonal entries, and r is the maximum number of linearly independent vectors.
•A singular triplet is (σj, u_j , v_j), where u_j is the jth column of **U**, v_j is the jth column of v and σ j the jth diagonal entry of **Σ**. Note that $\sigma j u_j * v_j^T$ is an ***mXn*** image.
•The singular triplets of the matrix A contain information about the image . The first singular triplet A_1 gives the outline of the image, as shown in Figure 1.

Figure 1. BW Image of the PBL building

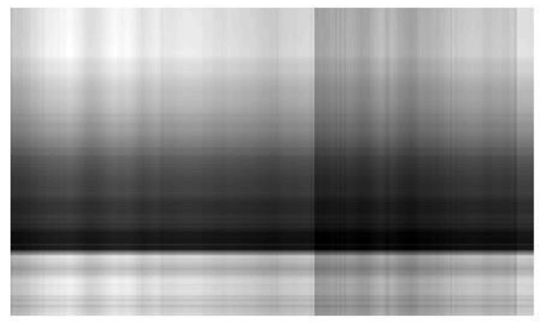

Figure 2. Outline the PBL building.

• The subsequent singular images $A_2 \ldots A_n$ hold information about finer details that help define the image, The singular values σ$_i$, weigh the contribution of each layer to the image, and that is why σ$_1$ is dominant. A_{20} and A_{50} can be seen below:

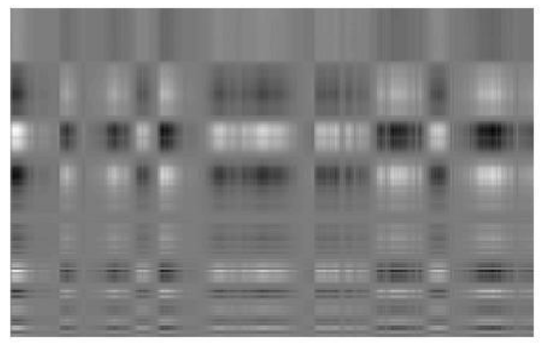

Figure 3. A_5, 5th scaled triplet

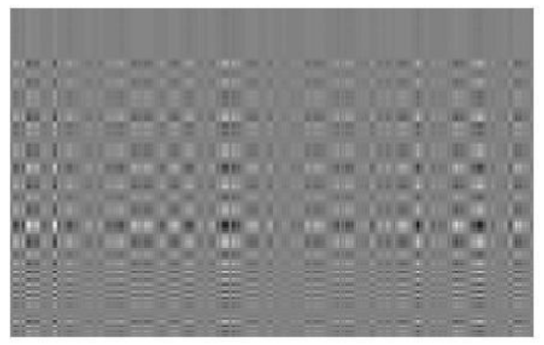

Figure 4. A_{20}, 20th scaled triplet

Fourier Transform.

Images as Waves

- The idea behind the Fourier transform like any other transform in Mathematics, is to take a problem in one domain and transform it into something easier to work with.
- The Fourier transform converts a signal *f* from the time domain to a frequency domain. The signal can be thought of as a sum of different sine and cosine functions at different frequencies.
- The jth Fourier coefficient of *f* indicates how strong the contribution of the sinusoidal waves with frequency j, is to *f*.
- Images can be interpreted in wave terms analogously in 2-dimensional space.
- The FFT (fast Fourier Transform) is an algorithm which very quickly calculates the Fourier transform of a signal. In the case of images, we use its 2 dimensional version, FFT2 in MATLAB.

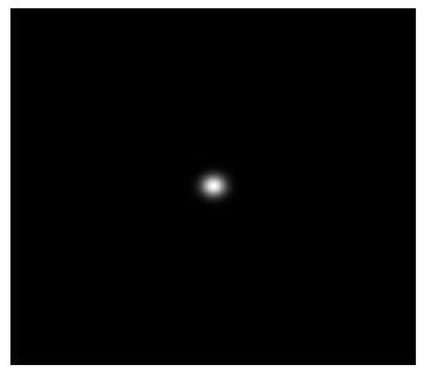

Figure5. Gaussian kernel

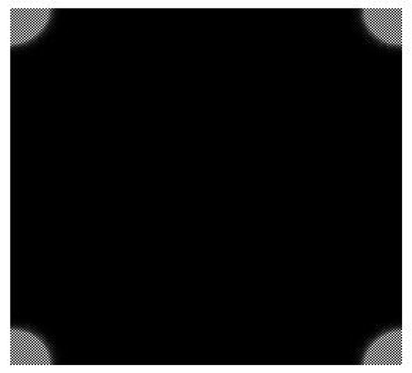

Figure6. FFT2 of Gaussian kernel

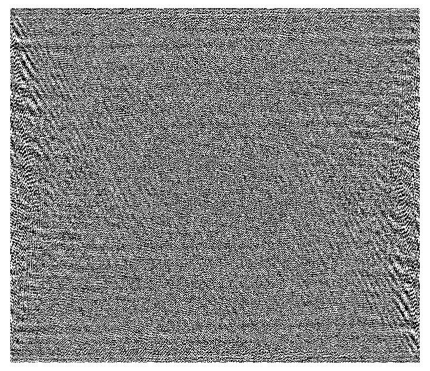

Figure7. FFT2 of Figure.1

- The FFT2 of the kernel (LATER).

Noise in Images

- If we have additive white Gaussian noise in the image then

$$A_{noisy} = A + nE$$

- E is a matrix of size A of random numbers and n the noise factor . The noise affects the fine details more, like brightness, but less the general features.
- **Image as matrix**: The last singular triplet of … e first

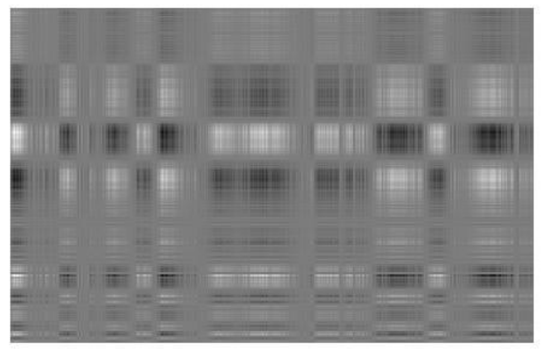

Figure 8. A_{noisy5}, 5th triplet, n= 0.05

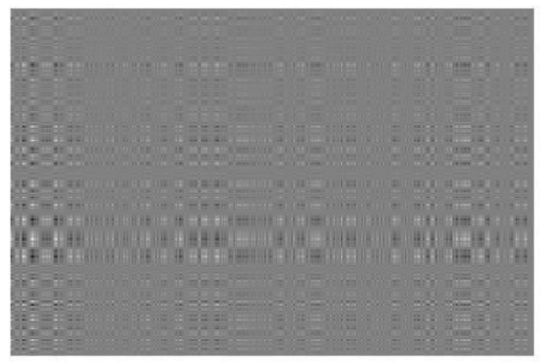

Figure9. $A_{noisy20}$, 20th triplet, n = 0.05

- The difference in the 5th and 20th triplets of the original image and the 0.05 noisy image can be seen by comparing Figure3 to Figure8 and Figure4 to Figure9. The singular triplets corresponding to smaller singular values are more affected by the noise damaged than those corresponding to the larger ones.
- **Image as waves**: The noise in the frequency domain affects component waves with the higher frequencies more, suggesting that denoising can be implemented by truncation.
- This, in turn, will reduce the definition of the image.

Denoising using TSVD

- Given an image corrupted by additive noise, we can filter the noise by taking the SVD of its matrix representation and replacing the image by one whose matrix representation is the sum of the first k singular triplets.
- $$A_{denoised} = \sum_{j=1}^{k} \sigma_j u_j v_j^T \qquad \text{where } k \le n$$
- Some care must be put into choosing a cutoff value, k.
- From results it could be seen that the larger the noise the lower the value of k needed to remove the noise.
- This makes sense because the noise is larger it would significantly corrupt larger singular triplets also, hence the need to truncate them.

Figure10. Image with noise factor 0.05

Figure11. Image denoised via TSVD with k = 80

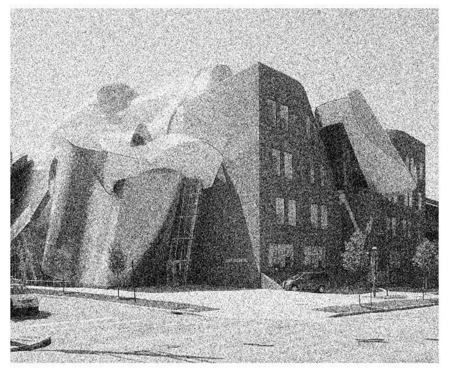

Figure12. Image with noise factor 0.2

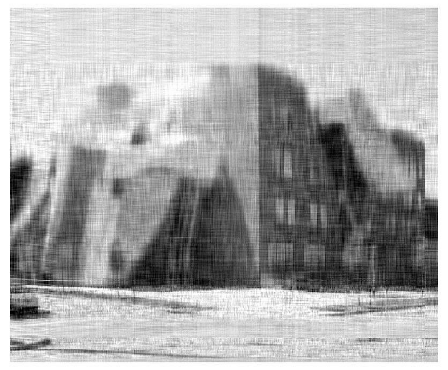

Figure13. Image denoised via TSVD with k = 20

Denoising using FFT

- We convert the image to frequency domain.
- The higher frequencies are usually at the borders of the FFT2 image; to truncate them we set these entries to zero
- The inverse FFT2 (IFFT2) takes us back to the time t domain.

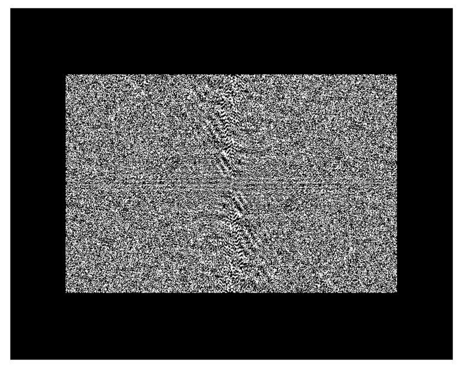

Figure14. Denoising in frequency domain the image of Figure10, n = 100.

Figure15 . Image in Figure 12 denoised in frequency domain

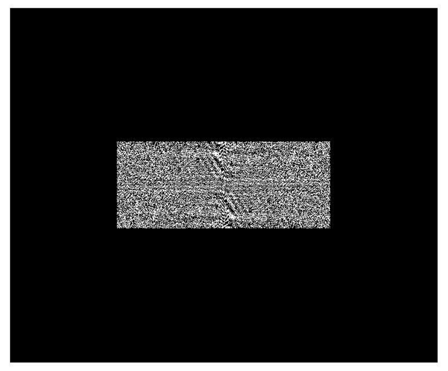

Figure16. Denoising in frequency domain the image of Figure12, n = 200.

Figure17. Image in Figure 12 denoised in frequency domain

- The higher the noise level, the more high frequencies we have to cut off, ***n*** in the captions above is the number of pixels set to zero.
- When the noise level is high, it cannot totally be removed without removing much of the image, therefore a balance must be reached so that the image is still viable and noise is significantly reduced.

Convolution

Images can be degraded by blurring, smearing the light across pixels. We blur the PBL building image in the frequency domain by an element-wise multiplication of its Fourier transform with that of a Gaussian kernel shown in Fig. 5. The blurred image is shown in Figure 18

Figure18. Blurred Image

Figure19. Clear Image

Figure20. Deblurred image with noise factor 0.2 and ε = 1e-1

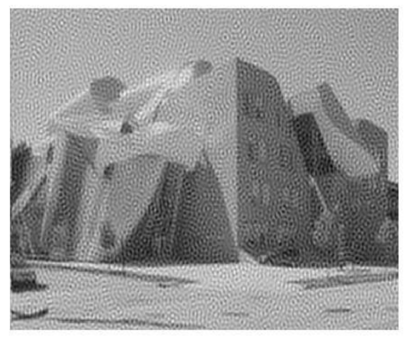

Figure21. Deblurred, image with noise factor 0.2 and ε = 1e-3

- Naively, we can deblur the image by inverting the blurring operation, that is by performing element-wise division in frequency space.
- The rounding of small numbers to 0 in finite precision arithmetic creates problems. To overcome this, a matrix of very low numbers (e.g., ε = 1e-6) is added to the Fourier transform of the Gaussian image to prevent division by zero. With no noise this can give nice results, see Figure 19.
- The formula for computing the FFT of the deblurred image is

$$\hat{X}_\varepsilon = \hat{Y} ./ (\overline{\hat{H}} .* \hat{H} + \varepsilon 1) .* \overline{\hat{H}}$$

- Blurry and noisy images can be deblurred and denoised as can be seen in Figure20. The value of the ε chosen depends on the noise level of the images. An example of what happens with a wrong choice of ε is shown in Figure21.

Acknowledgement: This work was conducted as part of the Summer Undergraduate Research Program and supported in part by **ACES+**

References

[1] Lloyd N. Trefethen, David Bau, *Numerical Linear Algebra* (Siam , 1997).
[2] S. Allen Broughton, Kurt Bryan, Discrete Fourier Analysis and Wavelets (Wiley, 2009)

Fisk University

Case Western Reserve University