Ioannis Pitas

Digital Image Processing Fundamentals

Chapter 2

Digital image transform algorithms Lab exercises in EIKONA

Thessaloniki 1998

Chapter 2: Digital image transforms.

Exercise 2.1: 2-dimensional Fast Fourier Transform (2D FFT) computation.

In this exercise we see how we can use EIKONA to apply the 2D FFT algorithm on an image. We assume that the file BABOON.RAW is already loaded in BW buffer 0 and that no other buffers are allocated. We choose ***"Black and white Transforms 2-D FFT"*** and a message box appears informing as that at least two float buffers are required for this operation. After allocating two float buffers of dimensions 256×256 through the menu sequence ***"File Buffers"*** and repeating the above selection, a dialog box appears where we can choose BW buffer 0 as the source and the just allocated float buffers as the real and imaginary parts of the Fourier Transform. After the conclusion of the operation we can test the obtained results through the menu option ***"Black and white Transforms Inverse 2-D FFT"*** and see if the resulting image resembles our original one. We could also select ***"Black and white*** **Basic** ***Matrix to image (trunc.)"*** from the menu to have either the real or the imaginary part of the DFT, converted to image format in order for us to be able to view it. As an example, assuming that the imaginary part of the transform was stored in Float buffer 1, then by selecting ***"Black and white*** **Basic** ***Matrix to image (norm.)"*** and specifying Float buffer 1 and **<New Buffer>** as the source and destination buffers, we can convert the imaginary part of the FFT of BABOON to image format, as shown in Figure 1.

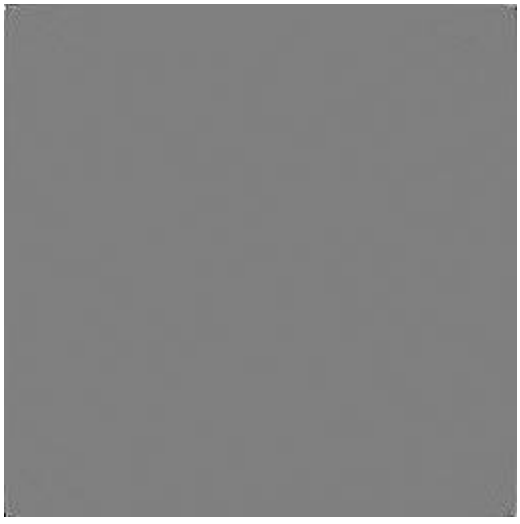

Figure 1: Imaginary part of the 2D FFT of BABOON.

Exercise 2.2: 2-dimensional Discrete Cosine Transform (2D DCT) computation.

In this exercise we learn how to apply the 2D DCT on an image. The 2D DCT is widely used in image compression techniques. For instance, the 2D DCT is used in the JPEG compression algorithm. As an example, we apply the 2D DCT on BABOON.

Assuming BABOON.RAW is already loaded in BW buffer 0, we select the menu option ***"Black and white Transforms DCT"*** and a message box appears reminding us that we need at least one float buffer for this operation. After allocating a float buffer of dimensions 256×256 through the menu option ***"File Buffers"*** and repeating the above selection, a dialog box appears where we choose BW buffer 0 as the *Source* the just allocated float buffer as the *Coefficients* matrix of the DCT Transform. Then, we press the "OK" button to perform the DCT transform. If we want to test the operation results, we can perform an inverse DCT by selecting the menu option ***"Black and white Transforms Inverse DCT"*** and see if the resulting image resembles our original one.

Exercise 2.3: Computing the periodogram of an image.

In this exercise we use EIKONA to compute the periodogram of BABOON. The periodogram of an image is a way to estimate its *Power Spectral Density*. Assuming that BABOON is loaded on buffer 0 and that at least two float buffers of size 256×256 have already been initialised, we select the menu option ***"Black and white Transforms Periodogram"***. We select 0 and **<New Buffer>** for the *Source* and *Destination* image buffers respectively, and press "OK". The periodogram of BABOON now resides in the newly created image buffer and can be seen in Figure 2.

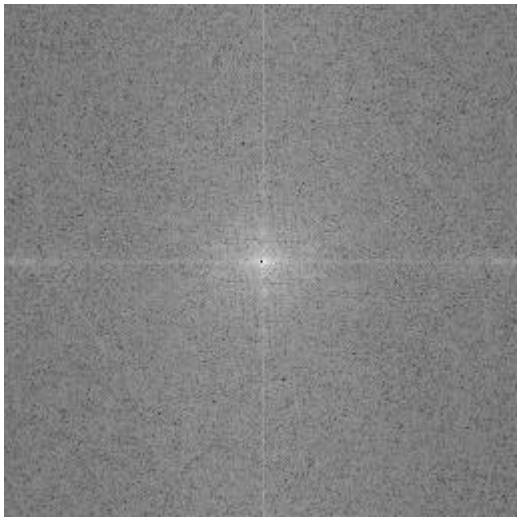

Figure 2: The periodogram of BABOON.

Exercise 2.4: Computing the AR Power Spectral Density (PSD) of an image.

Like the periodogram, the AR PSD and Blackman Tukey PSD are used for Power Spectral Density estimation of images. In this example, we perform an AR PSD estimation on BABOON.

Assuming that BABOON is loaded in BW buffer 0, we can get an AR PSD estimate, by selecting the menu option ***"Black and white Transforms AR PSD"***. In the dialog box that appears we specify 0 as the source image buffer and **<NewBuffer>** as the destination buffer. If we enter 3, 3, and 3 as the AR model *left x*, *right x* and *y* coefficient window sizes, then the result will resemble the one of Figure 2. Notice the similarity between Figure 2 and 3; one can see that the periodogram is a "noisy" version of the AR PSD.

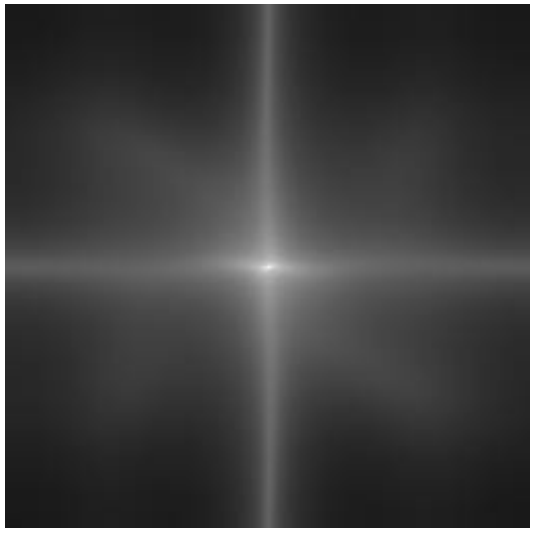

Figure 3: AR PSD of BABOON.

Exercise 2.5: Computing the convolution of two images.

Let us suppose that the 256×256 image BABOON is loaded in BW buffer 0. In the dialog box that appears through the menu option ***"Black and white Transforms Convolution"***. As an example, we will perform the convolution of BABOON with a 11×11 BW image whose pixel values have been set to a brightness level of 1 (a 11×11 binary "white" image).

First, we create two image buffers and two float buffers of size 512×512. This is done by selecting the menu option "***File→Buffers→New Buffer***" and providing the proper buffer size in the window that appears. Next, we open BABOON.JPG, and, after it is displayed on the screen, we copy it to the first of the two image buffers we created. The next step is to create the 11×11 image. This can be done by selecting the first 11×11 block (in the upper left corner of the image) of the second 512×512 empty BW buffer (which is essentially a black image) using the mouse, and change its pixel values from 0 to 1.

The selection of the upper left 11×11 block is done by clicking the right button of the mouse on pixel 10×10 and dragging the mouse to pixel 0×0 (the coordinates of the pixel where the mouse is at any moment, as well as the size of the region of interest that is selected, can be seen at the ***Control*** window on the upper right). This can be done by using the binary *NOT* logical operator, which is accessible trough the selection ***"Black and white Basic Not "***.

After completing the above procedure, we perform the convolution operation: we set the two 512×512 image buffers as the source buffers and the first of the float buffers as the destination buffer, and finally we press the "OK" button. In the end, the convolution of the two images resides in the float buffer we set as *Destination*. By selecting "***Black and white→Basic→Matrix to Image(norm.)***" we convert the output float buffer to an image buffer and yield the result shown in Figure 4.

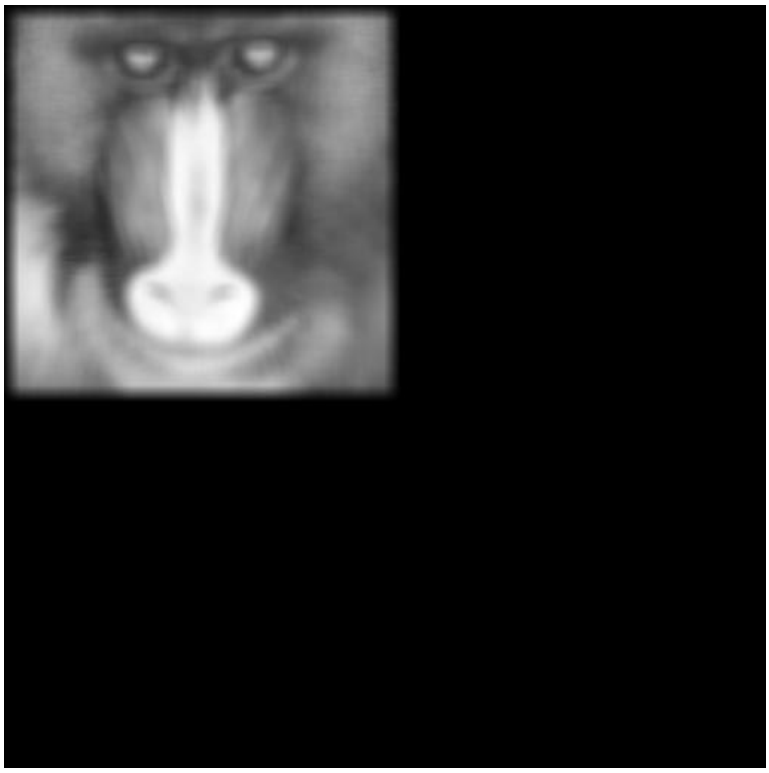

Figure 4: Convolution of BABOON with an 11×11 "white" binary image.