

Ioannis Pitas

**Digital Image Processing
Fundamentals**

Chapter 1

**Introduction
Lab exercises in EIKONA**

Thessaloniki 1998

Chapter 1: Introduction

Exercise 1.1: Load and display of an image file.

In this exercise we learn how we can use EIKONA to load an image file and display it on the screen. In particular, we shall see how to load and display RAW, TIFF, JPEG, TARGA, BMP and GIF image files.

1.1.1. Load a JPEG image file.

Let us suppose that a 256×256 grayscale image is stored in the file BABOON.JPG in the current directory. This image can be loaded as follows:

- Choose the “**File→Open**” menu option.
- A common Open File dialog box appears on screen which is used to select the image to be opened. Click on BABOON.JPG and then on the “Open” button, or double click on the name BABOON.JPG.

Upon completion of the above procedure, the JPG image file is automatically displayed on screen.

The same procedure can be used to load and display an image file in TIFF, TARGA, GIF and BMP format.

1.1.2. Load and display of a RAW image file.

A slightly different procedure must be used in order for EIKONA to open a RAW image file. This happens because EIKONA cannot determine the height and width of a RAW image automatically - it can only make a rough guess. Assuming that the 256×256 sized raw image file BABOON.RAW exists in the current directory, the procedure for loading is the following:

- Choose “**File→Open**” menu option.
- Double click on BABOON.RAW
- The *Open Raw* dialog box appears. The *width* and *height* fields of this dialog box display EIKONA’s guess about the image dimensions. At this point you can either:
 - Press the “Guess” button to allow EIKONA to make a new guess,
 - Press the “Swap” button to swap the values in the dialog box fields
or
 - Enter the image dimensions yourself.

If the RAW image is square (as in our example), EIKONA always makes the right guess.

Upon completion of the above procedure, the RAW image file is loaded and displayed on screen.

Exercise 1.2: Basic image processing

In this exercise, we perform some basic image processing operations using EIKONA. In particular, we see how we can use EIKONA to:

- brighten an image
- compute the negative of an image
- threshold an image
- produce a new image as the result of mixing two other images
- produce a new image as the result of adding two other images
- clip an image.

In order to implement the above operations, we use the image files BABOON256.JPG and PEPPER256.JPG. Both of them are of size 256×256.

1.2.1. Image brightening.

Let us suppose that the image file BABOON256.JPG of size 256×256 has already been loaded into BW Buffer 0. We assume that no other image buffers are currently allocated.

To brighten BABOON, we can add a suitable constant to the grayscale level of every point. This can be done by choosing the “*Black and white*→*Basic*→*Addc*” menu option. The dialog box that appears is very similar to most of the dialog boxes used in EIKONA for user input. Thus, we describe it in some detail. It can be divided in three areas:

- The buffer selection controls are used to specify the *Source* and *Destination* buffers for the operation. EIKONA takes great care to fill these combo-boxes only with valid values for the requested operation. The *Source* buffer control suggests the buffer displayed by the active window. The destination buffer control is filled only with buffers that have suitable dimensions to hold the function result. For most functions there is also an option to create a new buffer for the result.
- The *Buffer thumbnail view* option allows the user to see the contents of the buffer he selects from the combo-boxes. This is a valuable feature especially in cases when many buffers are allocated but are not currently displayed.
- The *Parameter* edit fields where the user can type in the values for various parameters the routine may require.

There are also the usual “OK” and “Cancel” buttons that allow the user to complete or cancel the operation.

Returning to our example, by choosing the buffer where BABOON256.JPG is loaded as the source buffer, requesting a new buffer for output, entering 50 for the constant to be added and pressing the “OK” button, we get the new brightened image of BABOON. Both images are shown in Figure 1. One can easily point out that the average luminance of the brightened image is bigger than that of the original.

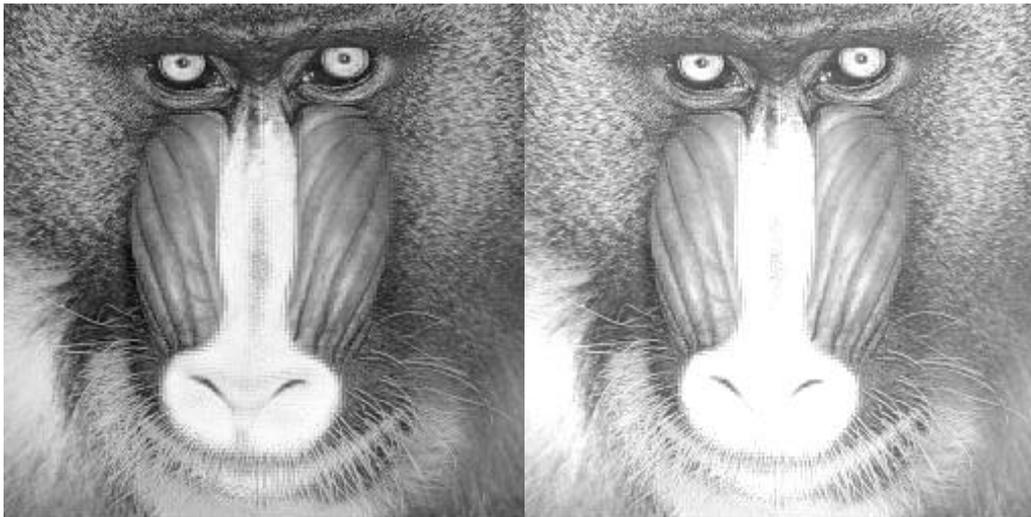


Figure 1: BABOON and its brightened image.

1.2.2. Computing the negative of an image.

Let us suppose that we want to compute the negative of BABOON and store it on the same buffer we stored the brightened BABOON. This can be done by selecting the menu option “*Black and white*→*Processing*→*Neg*”. In the dialog box, we select BABOON and its brightened image as the source and destination buffers of the operation and press the “OK” button. At the end of the operation, the window displaying the brightened BABOON is repainted and shows the negative of image BABOON. This can be seen in Figure 2.

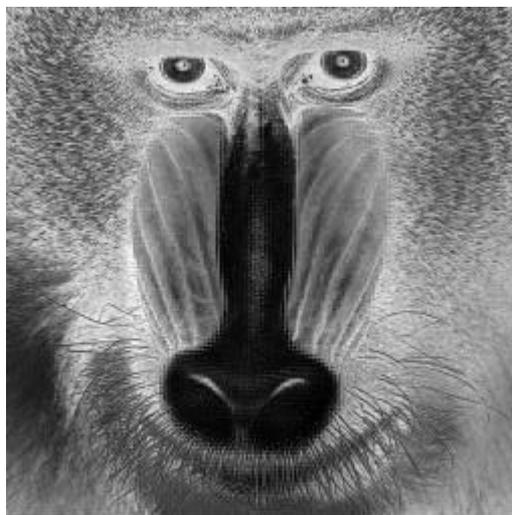


Figure 2: The negative image of BABOON.

1.2.3. Image thresholding.

In order to threshold an image we can choose the menu option “*Black and white*→*Analysis*→*Region Segmentation*→*Threshold*”. To perform thresholding, if image BABOON is currently loaded in BW buffer 0, we choose BW buffer 0 and <NewBuffer> from the source and destination controls respectively, and 150 as the

threshold level. We then press the “OK” button. The result of this operation is the thresholded version of the BABOON image. In the latter, every pixel that has a value greater than or equal to 150 in the original image, appears as white in the output image, whereas every pixel that has a value less than 150, appears as black.

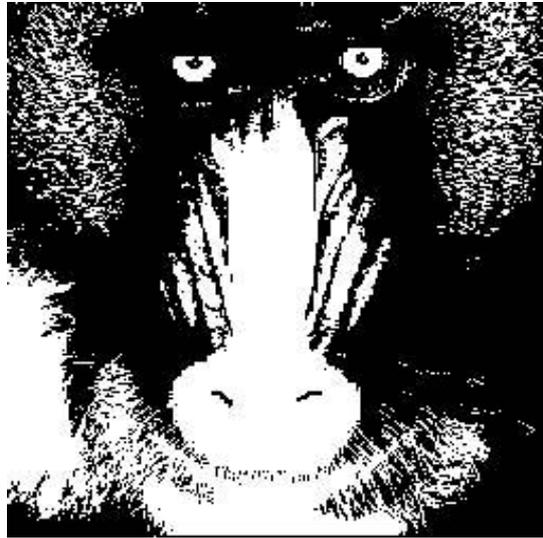


Figure 3: The thresholded version of BABOON.

1.2.4. Image mixing.

Let us suppose that the images PEPPER and BABOON are loaded in the BW buffers 0, 1 respectively. If we want to mix the two images of buffers 0 and 1 and store the result in a new buffer, we select the menu option “***Black and white***→***Basic***→***Mix***”. In the dialog box that appears, we choose buffers 0 and 1 as source buffers, <NewBuffer> as destination buffer and select the mixing percentages of the two images. Finally we press the “OK” button in order for the mixing operation to be performed. The result of the above process is the mixing of the two images. In Figure 4 we present the result of the mixing operation for different mixing parameters.

1.2.5. Image addition.

Let us suppose that the images PEPPER and BABOON are loaded in the BW buffers 0 and 1 respectively. We can perform addition of those two images by selecting the menu option: “***Black and white***→***Basic***→***Add***”. In the dialog box that appears, we choose buffers 0 and 1 as the source buffers, and <NewBuffer> as the destination buffer. After pressing the “OK” button, a new buffer is created containing the result of the image addition. This can be viewed in Figure 5.

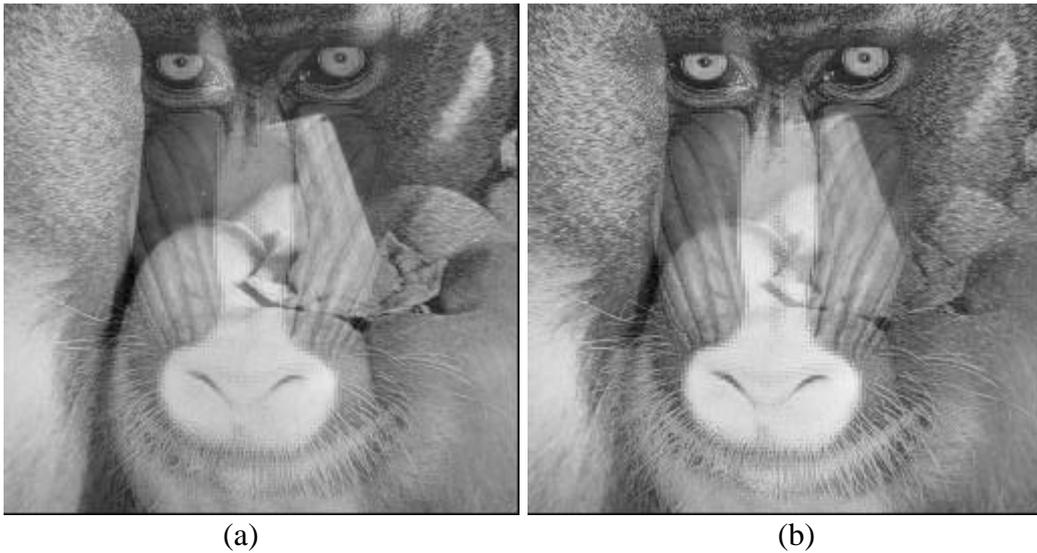


Figure 4. (a) Result of mixing PEPPER and BABOON with mixing percentages 0.5 and 0.5 respectively. (b) Result of mixing PEPPER and BABOON with mixing percentages 0.3 and 0.7 respectively.



Figure 5: Result of adding PEPPER and BABOON.

1.2.6. Image clipping.

If we want to clip an image we should select the menu option “*Black and white*→*Basic*→*Clip*”. By assuming that image BABOON is currently loaded in BW buffer 0, we choose BW buffer 0 and <NewBuffer> as the source and destination buffers respectively, 50 as the minimum clip value and 150 as the maximum clip value. We then press the “OK” button. The clipped version of BABOON is shown in Figure 6.

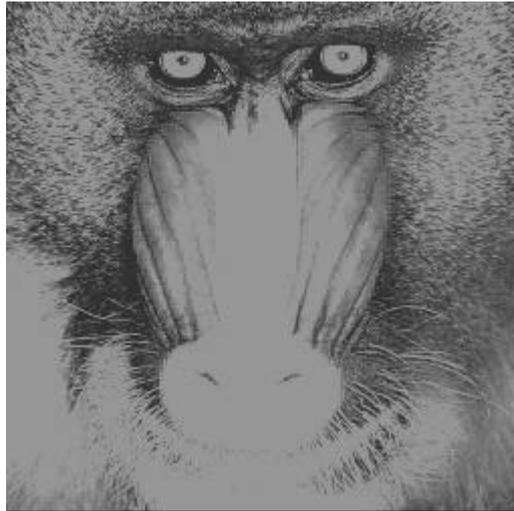


Figure 6: Clipped version of BABOON.

Exercise 1.3: Geometrical transformations.

In this exercise, we use EIKONA to perform examples of geometrical image transformations. The examples we implement are the zooming and the rotation of an image.

1.3.1. Image rotation.

In EIKONA, the image rotation operation can be accessed by selecting the menu option: “*Black and white*→*Basic*→*Rotate*”. Assuming that BABOON has already been loaded in a BW buffer, in the dialog box that appears we select BABOON and <New Buffer> as the source and destination buffers of the operation, 45 as the value of the rotation angle, and we then press the “OK” button. At the end of the operation the new buffer contains the rotated version of BABOON. Note that rotation is performed in an anticlockwise manner. The result of this exercise is depicted in Figure 7.

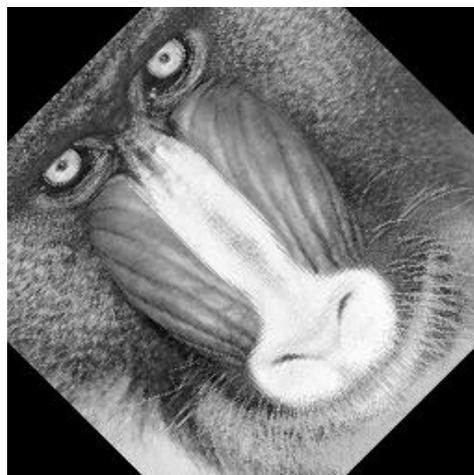


Figure 7: BABOON rotated by 45 degrees.

1.3.2. Image zooming.

In order to access the zoom function in EIKONA, we select the menu option “*Black and white*→*Processing*→*Zoom*” or “*Color*→*Processing*→*Zoom*” to zoom a BW or Color image respectively. Subsequently, we fill in the choices in the dialog box: we enter the source and destination buffers, the zoom factor (which must be a float number greater than or equal to 1, and then we select the interpolation order (Zero order, Linear, Bell shaped or Cubic B - spline). After completing the above, we press the “OK” button to perform the zooming operation.

As an example, zooming a part of BABOON with a zoom factor of 2 and the interpolation order set to *Linear* produces the image shown in Figure 8.



Figure 8: Zoomed region of BABOON .

Exercise 1.4: Noise generation.

In this exercise, we use EIKONA to perform simulations of noise effect on images. Specifically, we see how to add impulsive, additive and multiplicative noise to an image.

1.4.1. Impulsive noise corruption.

If we want to add impulsive noise to an image, we should select the menu option “*Black and white*→*Processing*→*Impulsive*”. In the fields of the dialog box we enter the buffer that BABOON is stored as *Source* and <New Buffer> as *Destination*. Then we need to fill in the rest of the fields. A brief description of the functionality of those fields is given below:

- In the *Noise probability* field, we enter a float number between 0 and 1. The greater the noise probability, the more corrupted the output image is likely to be. This can

be easily understood if we take into consideration the mathematical form of the impulsive sound:

$$f(x, y) = \begin{cases} z(x, y), & \text{with probability } p \\ i(x, y), & \text{with probability } 1 - p \end{cases}$$

Where $z(x, y)$ denotes the impulses, and $i(x, y)$ denotes the original image pixels.

- In the *Minimum and Maximum impulse value* fields, we enter integer numbers. The choices we make for the values of these fields have effect on the output image: The value of $z(x, y)$ is equal to minimum or maximum impulse value with probability $p/2$.

An example of this operation is shown in Figure 9.

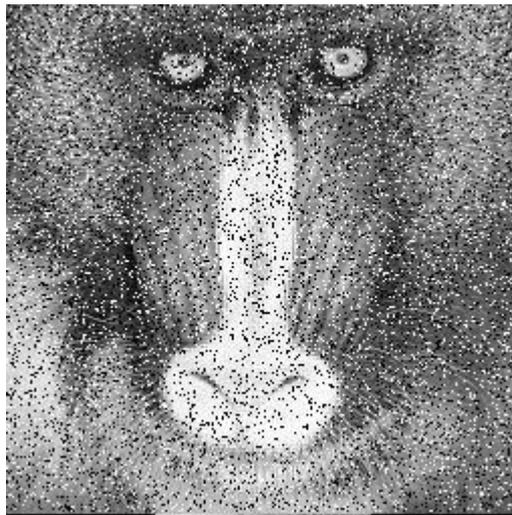


Figure 9: BABOON corrupted by impulsive noise. Noise probability: 0.2, minimum impulse value: 0, maximum impulse value 250.

1.4.2. Additive noise.

To add additive noise to an image, we use the “**Black and white**→**Processing**” submenu where we can choose *Add Uni*, *Add Gauss* or *Add Laplace* to add uniform additive noise, Gaussian additive noise or Laplacian additive noise respectively. Below are some examples of adding additive noise to BABOON.

- *Uniform additive noise.*

After selecting *Add Uni* from the menu option “**Black and white**→**Processing**”, we fill in the fields of the dialog box that pops up. We specify as *Source* and *Destination* the buffer that BABOON is stored in and <New Buffer> respectively. Afterwards, we choose a value for the *Noise Range* field, and finally we press the “OK” button.

Below two examples of this operation can be seen, with the *Noise Range* field set to 100 for the first image (Figure 10a) and 200 for the second one (Figure 10b). The effect the noise range has on the output image is obvious: the higher the noise range, the more corrupted the output image is likely to be.

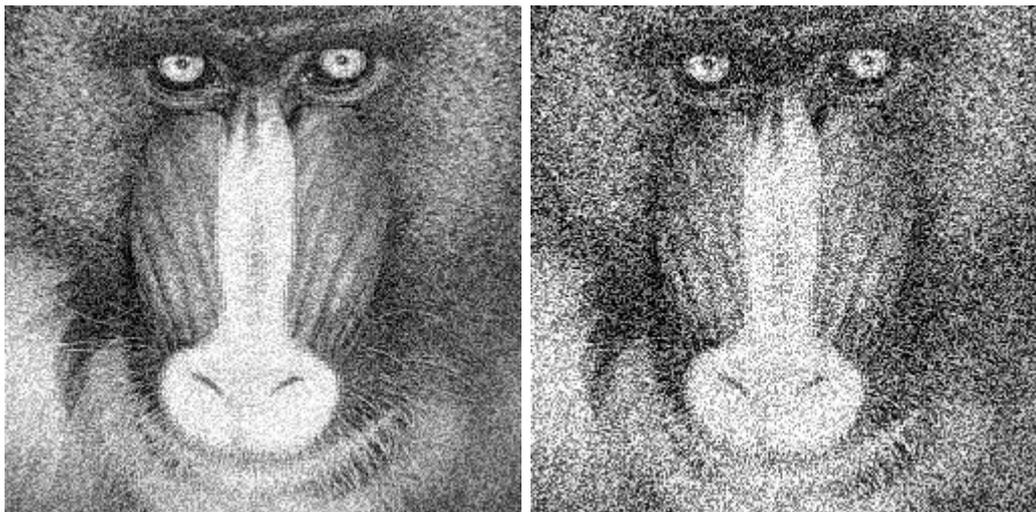


Figure 10: BABOON corrupted by additive uniform noise, (a) noise range = 100, (b) noise range = 200.

- *Gaussian additive noise.*

To add Gaussian noise to an image, we select the option “*Black and white*→*Processing*→*Add Gauss*”. Then we enter as *Source* and *Destination* the buffer that BABOON is stored in and <New Buffer> respectively. Afterwards, we must provide a value for the *Noise standard deviation* field. Finally, we press the “OK” button.

Below two examples of performing this operation on BABOON can be seen, with the *Noise standard deviation* parameter set to 30 for the first image (Figure 11a) and 100 for the second one (Figure 11b). Note the effect that this parameter has on the output images.

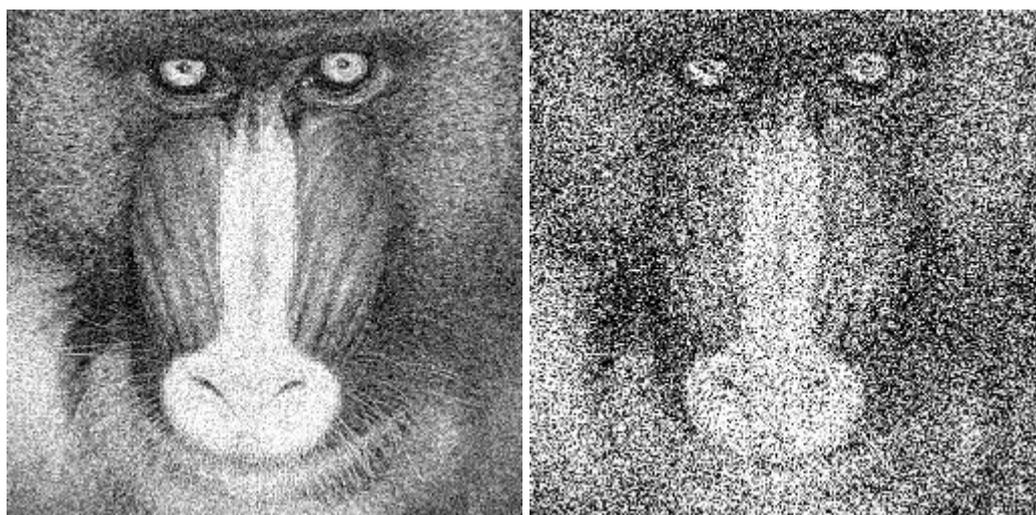


Figure 11: BABOON corrupted by additive Gaussian noise, (a) noise standard deviation = 30, (b) noise standard deviation = 100.

- *Laplacian additive noise.*

To corrupt an image with noise that follows the Laplace distribution, we select the menu option “**Black and white**→**Processing**→**Add Laplace**”. Then we enter as *Source* and *Destination* the buffer that BABOON is stored in and <**New Buffer**> respectively. Afterwards, we provide a value for the *Laplace coefficient* field. Finally, we press the “OK” button.

Below, two examples of performing this operation on BABOON can be seen, with the *Laplace coefficient* field set to 30 for the first image (Figure 12a) and 100 for the second one (Figure 12b).

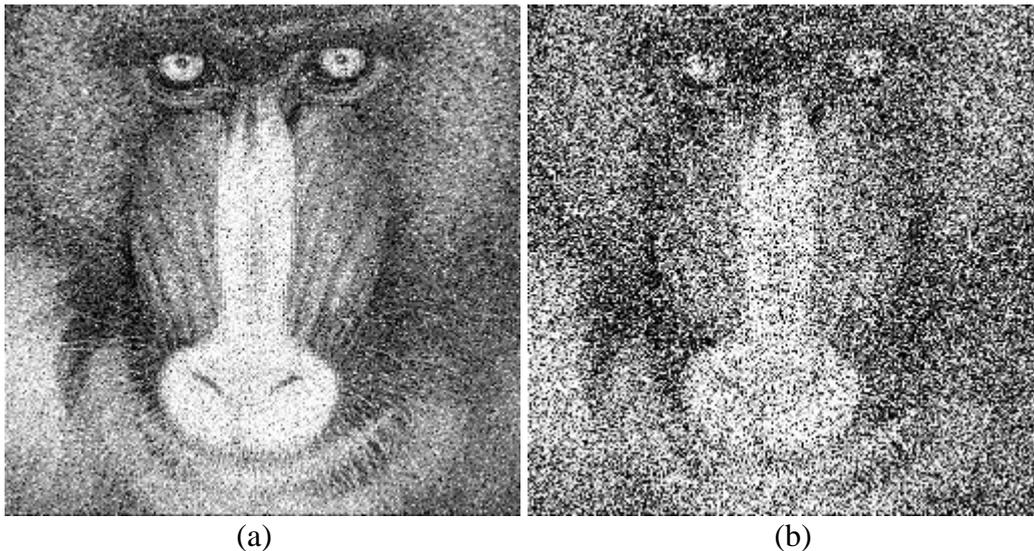


Figure 12: BABOON corrupted by additive Laplacian noise, (a) Laplace coefficient = 30, (b) Laplace coefficient = 100.

1.4.3. Multiplicative noise.

To add multiplicative noise to a BW image, we go to the “**Black and white**→**Processing**” submenu and choose *Mult Uni* or *Mult Gauss* to add uniform multiplicative noise or Gaussian multiplicative noise to the image respectively.

Below, there are some examples of corrupting BABOON with multiplicative noise.

- *Uniform multiplicative noise.*

After selecting *Multi Uni* from the menu option “**Black and white**→**Processing**”, we fill in the fields of the dialog box that pops up. We enter as *Source* and *Destination* the buffer that BABOON is stored in and <**New Buffer**> respectively. Afterwards, we choose a value for the *Noise Range* field, and finally we press the “OK” button.

Below two examples of this operation can be seen, with the *Noise Range* parameter set to 0.5 for the first image (Figure 13a) and 2 for the second one (Figure 13b):

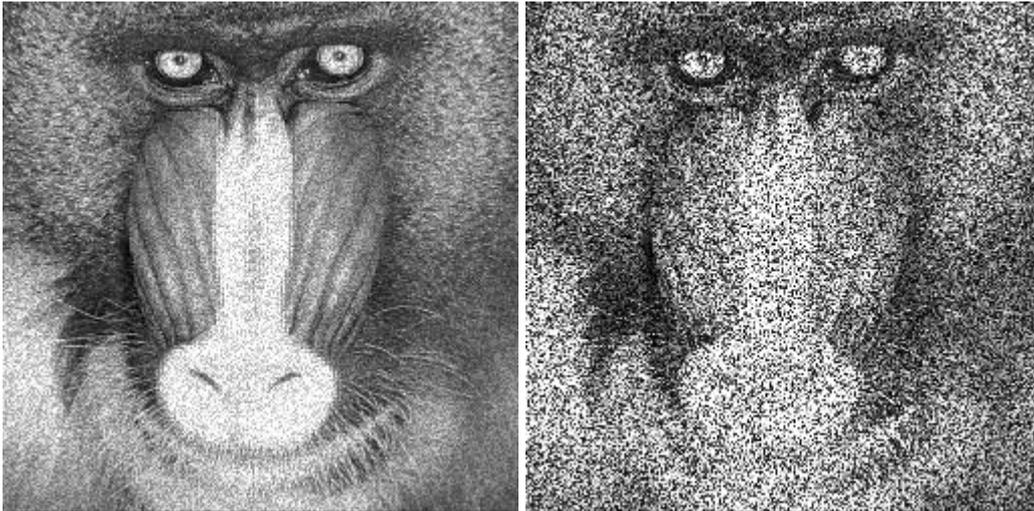


Figure 13: Uniform multiplicative noise, (a) noise range = 0.5, (b) noise range = 2.

- *Gaussian multiplicative noise.*

To add Gaussian multiplicative noise to an image, we select the menu option “**Black and white**→**Processing**→**Mult Gauss**”. Then we enter as *Source* and *Destination* the buffer that BABOON is stored in and <New Buffer> respectively. After this, we must provide a value for the *Noise standard deviation* field. Finally, we press the “OK” button.

Below two examples of performing this operation on BABOON can be seen, with the noise standard deviation parameter set to 0.5 for the first image (Figure 14a) and 2 for the second one (Figure 14b). Note the effect this parameter has on the output images.

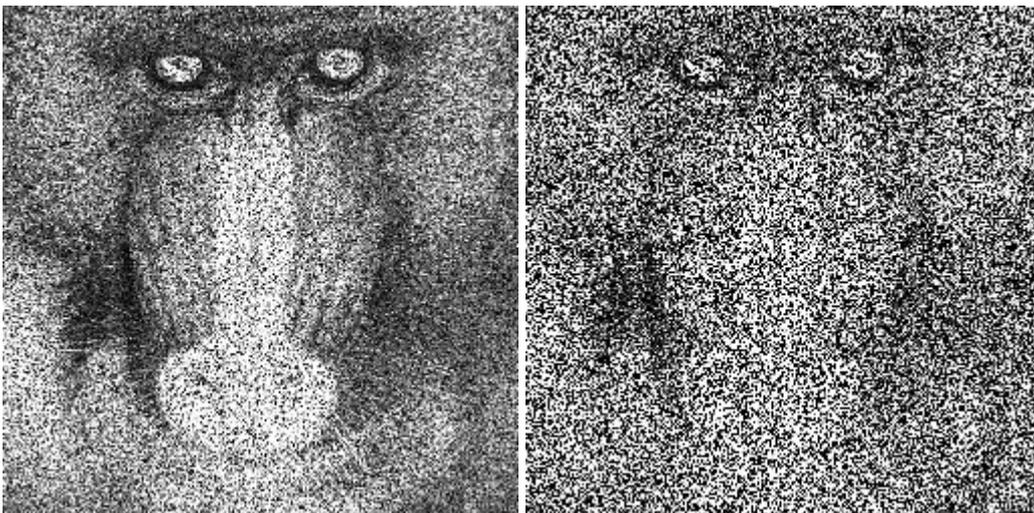


Figure 14: Gaussian multiplicative noise, (a) noise standard deviation = 0.5, (b) noise standard deviation = 2.

Exercise 1.5: Color transformations

In this exercise, we present some examples to familiarize the user with EIKONA's color transformation operations. These operations require a color image as input. This means that the user needs to load a color image in order for the examples to be carried out. Thus, we assume that a color version of BABOON.JPG is stored in the current directory. This is the image we shall use to perform the examples.

1.5.1. RGB to XYZ transformation.

To transform a color image from the RGB to the XYZ color model, we have to follow the menu option "*Color*→*Color Repr*→*To XYZ from*→*CIE RGB*". We enter as *Source* and *Destination* the color buffer that BABOON is stored in and <New Buffer> respectively. The results can be seen in Figure 15.

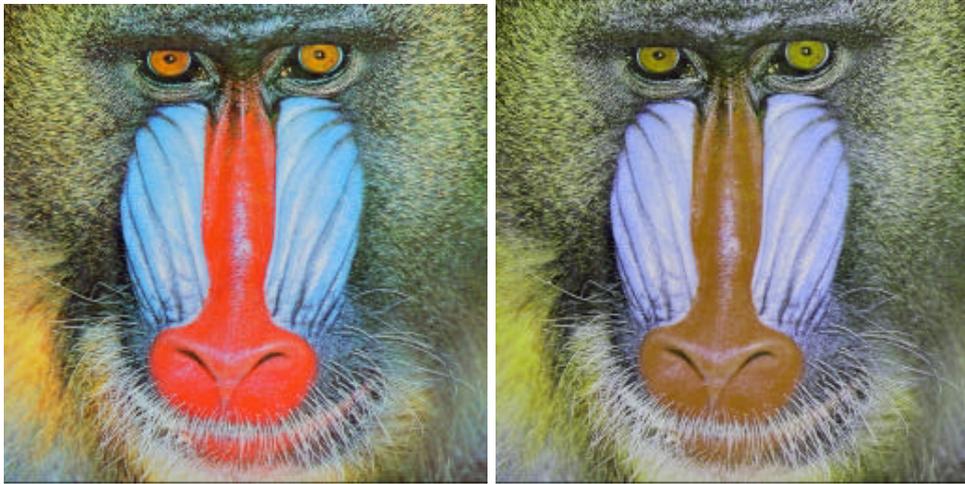


Figure 15: BABOON and its XYZ color model version.

1.5.2. RGB to CMY transformation.

Transforming a color image from the RGB to the CMY model is also a straightforward operation. We need to select the menu sequence "*Color*→*Color Repr*→*From RGB to*→*CMY*". We enter as *Source* and *Destination* the colour buffer that BABOON is stored in and <New Buffer> respectively. The resulting image can be seen in Figure 16. CMY color space is used in the graphics and printing industry.



Figure 16: The CMY version of BABOON.

1.5.3. RGB to HLS transformation.

This color transformation procedure is slightly different than the two examples mentioned above, especially in terms displaying their result on screen. To perform this transform, we select the sequence “*Color*→*Color Repr*→*From RGB to*→*HLS*”. In the dialog box we notice that we have to specify three float buffers, not a colour image, as the operation destination. This is due to the nature of the transform: the value range of its output cannot be stored in an image buffer (an image buffer is an integer buffer which can hold numbers from 0 to 255). If we need to see the transform results we can copy the three float buffers to image buffers using the menu sequence “*Black and white*→*Basic*→*Image to matrix norm*”. This copies a float buffer to an image one while normalizing its value range. The results of the transform to HLS can be seen in Figure 17.



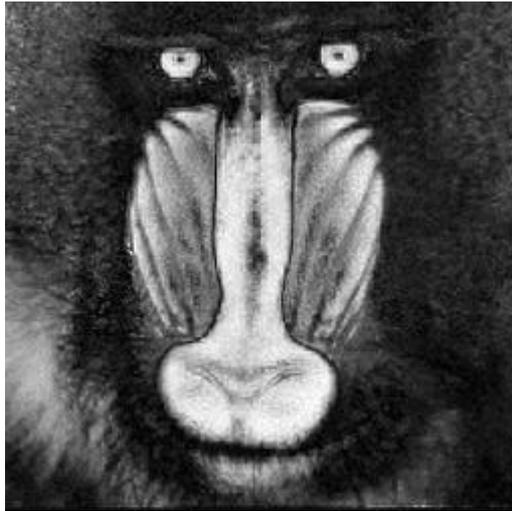


Figure 17: Channels H, L, S of the HLS version of image BABOON.