EE550 - Image and Video Processing - Lab 2 Report

Baris Sevilmis

Lab Date: 23/10/2019
Due Date: 06/11/2019

1 Overview

Second lab assignment of EE550 focuses various different Dithering Algorithms, where main aim is to reconstruct an grayscale image as close to original grayscale version as possible. Concepts to be implemented are Fixed Threshold Method, Random Threshold Method, Ordered threshold Method, Ordered Matrix with centered points, Diagonal Ordered Matrix with balanced centered points & Ordered matrix with dispersed dots. Details and experimentation of these concepts are clearly explained in the corresponding sections. Results in terms of visual representations are mentioned in corresponding sections, and error & complexity analysis of each method and algorithm are discussed in the last section of the report

2 Fixed Threshold Method

Fixed Threshold Method is one of the most basic dithering methods, in which a simple fixed threshold is used. Depending on the pixel intensity values, new pixel values are obtained. In case that a pixel value is greater or equal than fixed threshold color is changed to *255*, namely **white**. Otherwise new pixel value is attained as *0*, namely **black**. Following formulas depict the procedure:

$$Image(:,:) >= Threshold = 255.0$$
$$Image(:,:) < Threshold = 0.0$$

For the sake of simplicity, threshold is picked as *128* in our case, however depending on the properties of image such as brightness, more successful thresholds could be picked. Figure 1 demonstrates our original test images, which are **Lena** & **Wool**. On the other hand, Figure 2 demonstrates reconstructed images with fixed threshold of . **128**. As it can be seen, method fails to reconstruct details of the images even though main properties are demonstrated. Using a specific threshold, even picked after tuning, is not withstanding method to rebuild an image.

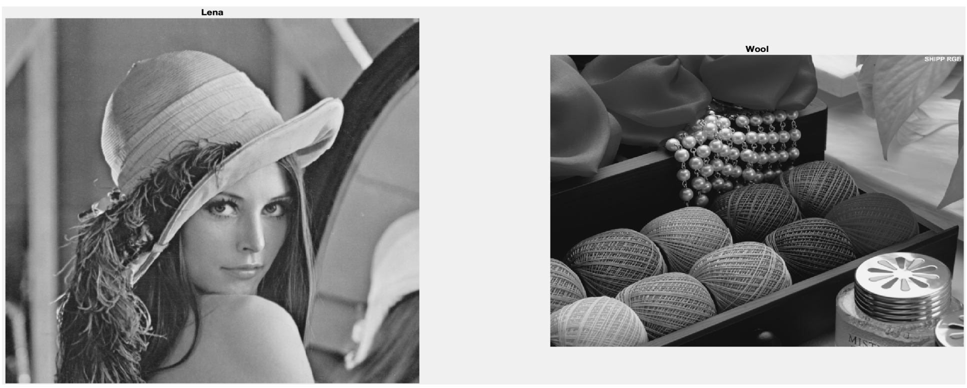

Figure 1: Lena & Wool : Original

Figure 2: Lena & Wool : Fixed Threshold of 128.0

3 Random Threshold Method

In the second section purpose is to implement Random threshold method, which is known as an improved version to Fixed Threshold Method. In addition to the constant threshold, image is preprocessed before threshold operation, in which a noise is added to the intensity values of image. Therefore, image becomes more robust to threshold operation. Thresholding after noise addition removes false contours. Noise is chosen according to a uniform distribution., given by **unidrnd** function in MatLab. Figure 3 provides an flow of the random threshold method.

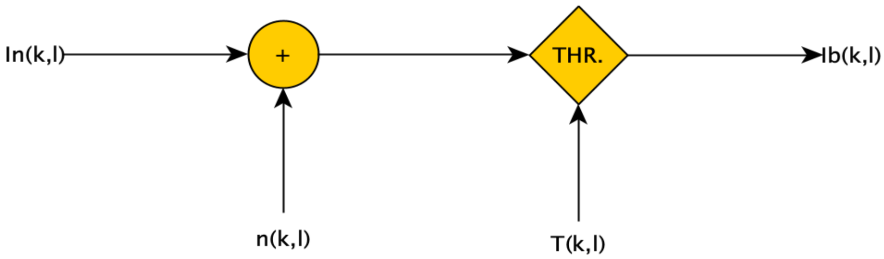

Figure 3: Random Threshold Method

Noise amplitude is the main variant in this method, for this reason various noise amplitude have been chosen as they are provided to uniform distribution function to create an overall noise for the image. Figure 4 provides again the original *lena* & *wool* images, as Figure 5, 6 & 7 provides these images with certain amount of noise. Figure 5 has the noise amplitude of 50, Figure 6 has the noise amplitude of *130* and lastly Figure 7 has the noise amplitude of *250*. It can be seen that adding noise helps reconstructed image quality to improve compared to fixed thresholding. However, after a certain noise amplitude, rebuilt image quality drops incrementally and noise can be seen easily in image as salt-pepper type of noise.

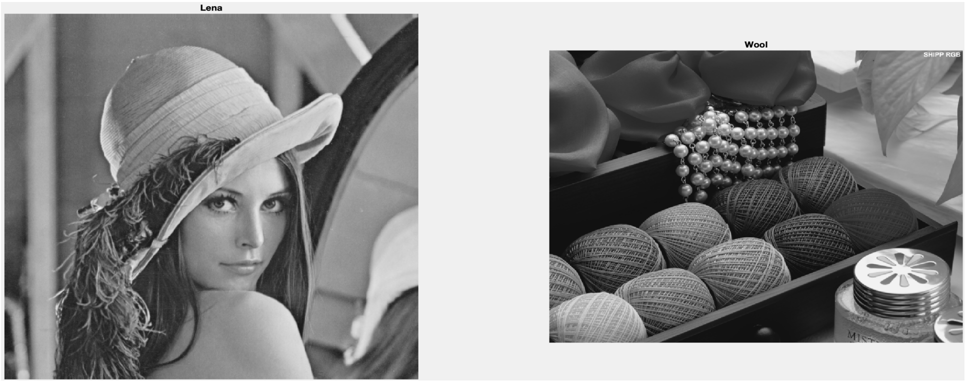

Figure 4: Lena & Wool : Original

Figure 5: Lena & Wool : Fixed Threshold of 128.0 & Noise of 50.0

Figure 6: Lena & Wool : Fixed Threshold of 128.0 & Noise of 130.0

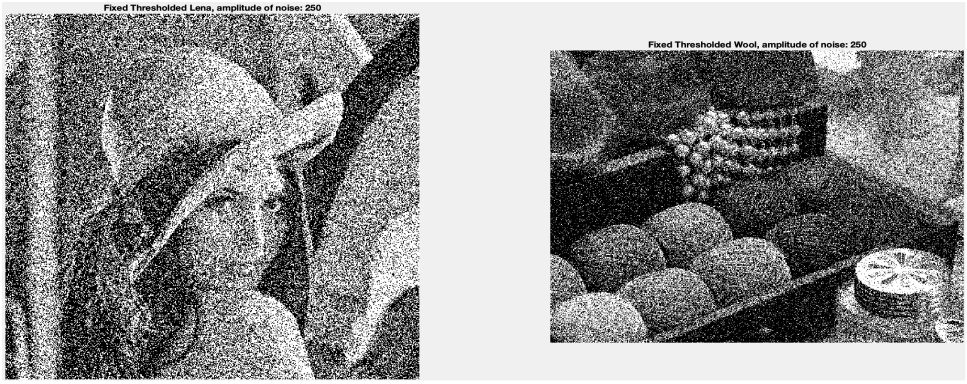

Figure 7: Lena & Wool : Fixed Threshold of 128.0 & Noise of 250.0

4 Ordered Threshold Method

Ordered Threshold Method is the third dithering method in the exercise, which is based on image quantization and threshold. Image is quantized to certain amount of grey levels, and then thresholded with a threshold matrix instead of a constant value. Threshold matrix can be chosen as various matrices, but in our case clustered dots matrix is used, in which there are 37 grey levels. Matrix is size of $6x6$ and it is obvious that values to the middle of the matrix are smaller compared to the borders. This matrix is depicted below as $S = D^{36}_{i,j}$:

$$S = D^{36}_{i,j} = \begin{bmatrix} 34 & 29 & 17 & 21 & 30 & 35 \\ 28 & 14 & 9 & 16 & 20 & 31 \\ 13 & 8 & 4 & 5 & 15 & 19 \\ 12 & 3 & 0 & 1 & 10 & 18 \\ 27 & 7 & 2 & 6 & 23 & 24 \\ 33 & 26 & 11 & 22 & 25 & 32 \end{bmatrix}$$

Grey levels of threshold matrix are in a normalized scale of $[0, 1]$, are $i/(N-1), i = 0, 1, 2, .., N-1$. After the quantization is done, dithering is done by the following operation($\hat{x} \in [0, N)$):

$$x(k,l) := Image(k,l)$$
$$\hat{x}(k,l) := Q(x(k,l)) = \left\lfloor x(k,l)\frac{N-1}{255} \right\rfloor$$
$$(\hat{x}(k,l) >= s(k,l)) := 255.0$$
$$(\hat{x}(k,l) < s(k,l)) := 0.0$$

Comparing image with this matrix is not straightforward as the size usually don't match. Therefore, matrix was applied to every $6x6$ part of the image one by one. Values greater than compared matrix value are turned into 255 as before, otherwise they are turned to 0. Figure 8 demonstrates original *lena* and *wool* images. Image quality increased tremendously compared to previous methods. However, reconstructed images are not as perfect as the original image although they have an improved sense of depth in terms of grey level compared to previous methods.

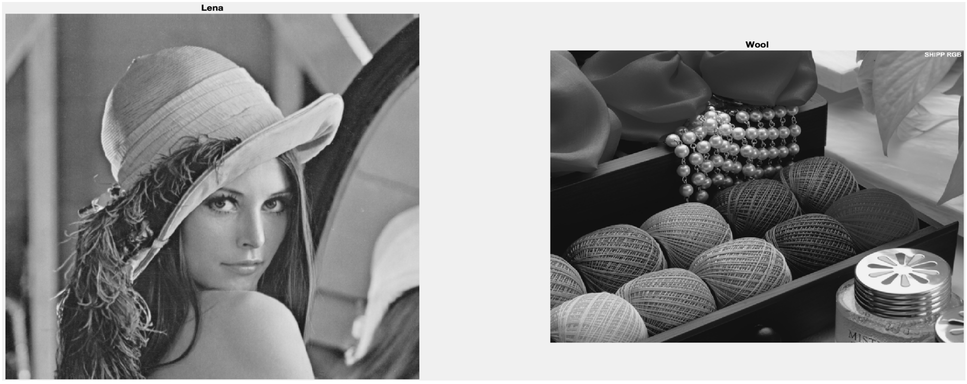

Figure 8: Lena & Wool : Original

Figure 9: Lena & Wool : Ordered Threshold Matrix $D^{36}_{i,j}$

5 Ordered Matrix with Centered Points

In this section of lab assignment, main aim is to reconstruct image as in Ordered Threshold Method(Section 4). In addition, new threshold matrices are used as threshold matrices. These matrices can be seen below as C_6 & E_6:

$$C_6 = \begin{bmatrix} 34 & 25 & 21 & 17 & 29 & 33 \\ 30 & 13 & 9 & 5 & 12 & 24 \\ 18 & 6 & 1 & 0 & 8 & 20 \\ 22 & 10 & 2 & 3 & 4 & 16 \\ 26 & 14 & 7 & 11 & 15 & 28 \\ 35 & 31 & 19 & 23 & 27 & 32 \end{bmatrix}$$

$$E_6 = \begin{bmatrix} 30 & 22 & 16 & 21 & 33 & 35 \\ 24 & 11 & 7 & 9 & 26 & 28 \\ 13 & 5 & 0 & 2 & 14 & 19 \\ 15 & 3 & 1 & 4 & 12 & 18 \\ 27 & 8 & 6 & 10 & 25 & 29 \\ 32 & 20 & 17 & 23 & 31 & 34 \end{bmatrix}$$

These matrices are really similar to $D^{36}_{i,j}$. They are same in terms of their size of $6x6$ and matrix balancing. Small values in the middle, and values get larger as they get near the border of the matrix. Base point of the matrix , meaning the *0* value, slightly changes its location. Results of these new threshold matrices are depicted in Figure 11 & 12, and Figure 10 can used for comparison. Although it is hard to see differences of these filter operations directly, zooming in on images provides much reasonable intuitions. Figure 13 demonstrates these specific details to understand differences between these methods.

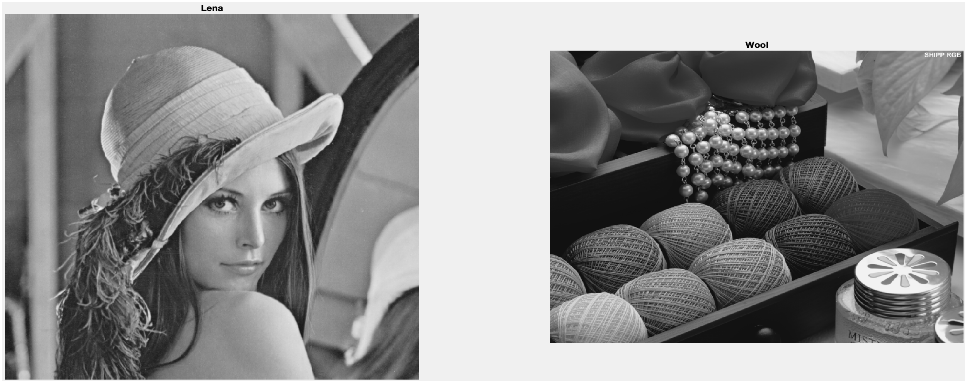

Figure 10: Lena & Wool : Original

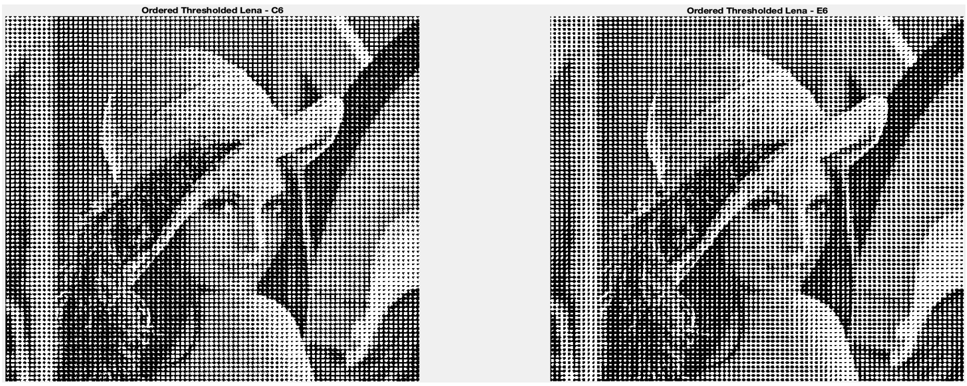

Figure 11: Lena : Ordered Threshold Matrix with Balanced Centered Points C_6 & E_6

Figure 12: Wool : Ordered Threshold Matrix with Balanced Centered Points C_6 & E_6

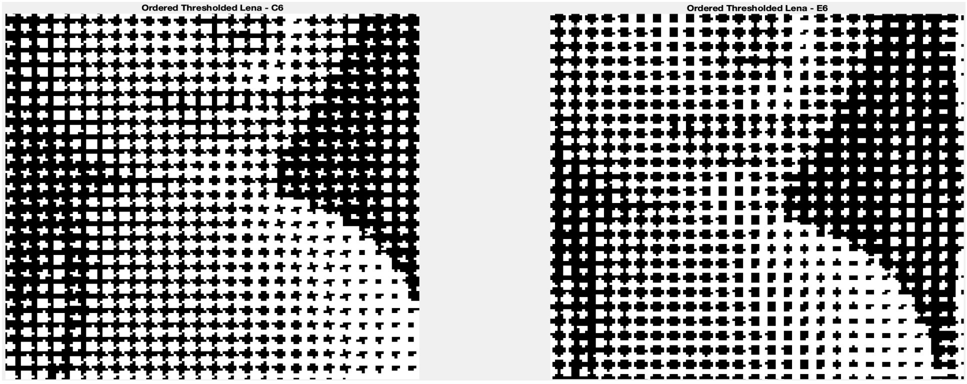

Figure 13: Lena : Ordered Threshold Matrix with Balanced Centered Points C_6 & E_6 - Zoom

Shapes of white and black points mostly differ from each other in both of the images. C_6 mostly results in diamond - cross shaped black and white points. On the other hand, E_6 ensures more symmetric circle-disk shapes which are more symmetric, namely balanced, and therefore, improves visual reconstruction of the image slightly compared to E_6.

6 Diagonal Ordered Matrix with Balanced Centered Points

In this section, methods in Section 4 & 5 are extended. New matrices are used in addition to quantization, nevertheless improving the reconstruction quality. O_8 is used as the threshold matrix, which consists of the different $4x4$ matrices, namely $O_{8,1}$ & $O_{8,2}$.

$$O_8 = \begin{bmatrix} O_{8,1} & O_{8,2} \\ O_{8,2} & O_{8,1} \end{bmatrix}$$

$$O_{8,1} = \begin{bmatrix} 13 & 9 & 5 & 12 \\ 6 & 1 & 0 & 8 \\ 10 & 2 & 3 & 4 \\ 14 & 7 & 11 & 15 \end{bmatrix}$$

$$O_{8,2} = \begin{bmatrix} 18 & 22 & 26 & 19 \\ 25 & 30 & 31 & 23 \\ 21 & 29 & 28 & 27 \\ 17 & 24 & 20 & 16 \end{bmatrix}$$

Additionally, the rhombus shape of the element paving the image plane is depicted below by Rhombus Shape matrix:

$$Rh.Shape = \begin{bmatrix} 0 & 0 & 0 & 18 & 0 & 0 & 0 \\ 0 & 0 & 8 & 25 & 30 & 0 & 0 \\ 0 & 3 & 4 & 21 & 29 & 28 & 0 \\ 7 & 11 & 15 & 17 & 24 & 20 & 16 \\ 22 & 26 & 19 & 13 & 9 & 5 & 12 \\ 0 & 31 & 23 & 6 & 1 & 0 & 0 \\ 0 & 0 & 27 & 10 & 2 & 0 & 0 \\ 0 & 0 & 0 & 14 & 0 & 0 & 0 \end{bmatrix}$$

Figure 14 again provides the original images that are to be reconstructed, and Figure 15 demonstrates much the application of 0_8 on both of the images. It can easily noticed that resulting reconstruction has a much better visual quality compared to results of previous section, namely horizontal and vertical lines are reduced in terms of visual representation. Threshold matrix is balanced, and it is oriented diagonally. It is known that eye is less sensitive to changes in non horizontal or non vertical directions, therefore human visual system can observe diagonally thresholded image with higher quality compared to previous threshold methods.

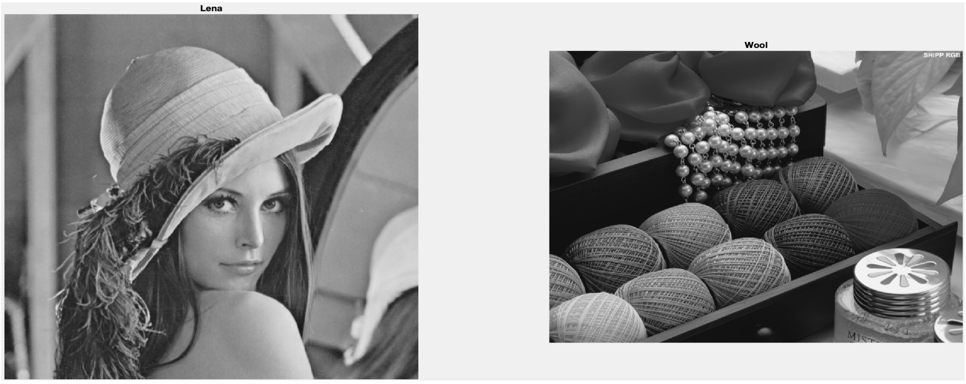

Figure 14: Lena & Wool : Original

Figure 15: Lena & Wool : Diagonal Ordered Threshold Matrix O_8

7 Ordered Matrix with Dispersed Dots

In this section, although previous sections methods are extended, there is additionally a huge change in threshold matrix representation. By the proposition of Bayer, a recursive algorithm was suggested to create a threshold matrix with sparse pattern. A matrix of size nxn,namely D_n, is used to conceive a matrix of size $2nx2n$, namely D_{2n}. Form of this matrix, and recursive submatrices can be seen below.

In our experimentation, two different threshold matrices of size $4x4$ & $6x6$ are used, to be more specific D_4 and D_6 matrices. U_n matrix on the other hand, are unit matrices that is filled with *1*'s. D_{2n} obtained by this algorithm minimize the occurrence of low frequencies, making reconstructed images preferable for human visual system.

$$D_{2n} = \begin{bmatrix} 4D_n & 4D_n + 2U_n \\ 4D_n + 3U_n & 4D_n + U_n \end{bmatrix}$$

$$D_2 = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix}$$

$$D_3 = \begin{bmatrix} 8 & 4 & 5 \\ 3 & 0 & 1 \\ 7 & 2 & 6 \end{bmatrix}$$

Figure 16 depicts original images, and Figure 17 & 18 depict results of D_4 & D_6. Resulting images, as mentioned before, are detected with higher visual quality by the human visual system. As the low frequency components disappear during the reconstruction phase, meaning the threshold phase, reconstructed images in both cases(*Lena* & *Wool*) get rid of their horizontal, vertical and noisy pixel distinctions on high level.

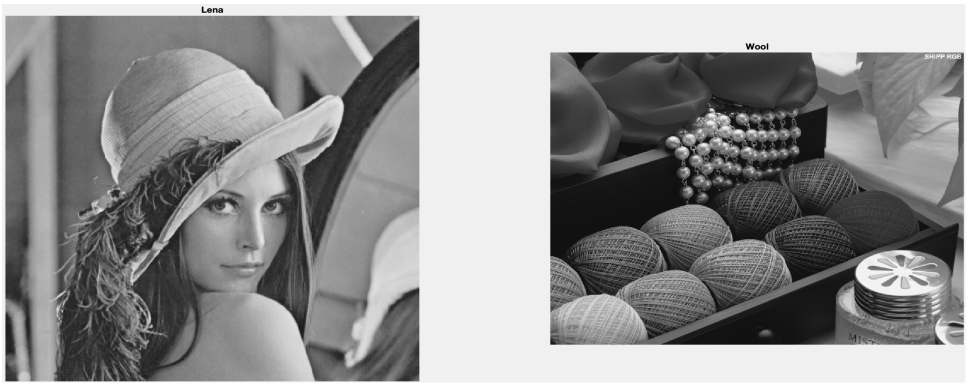

Figure 16: Lena & Wool : Original

Figure 17: Lena & Wool : Ordered Threshold Matrix with Dispersed Dots D_4

Figure 18: Lena & Wool : Ordered Threshold Matrix with Dispersed Dots D_6

8 Error Diffusion Method

In this section of the lab exercise, last method for image dithering will be discussed, which produces by far the best results in terms of visualization. This method is called Error Diffusion Method. Although, image quantization and recursive diagonal balanced thresholds provide astonishing visual results, there exist still noticeable error remaining from the reconstruction phase. Nevertheless, error diffusion method extends previous approaches by a *multi-step* approach. Each pixel in the image is thresholded, and the error of thresholding operation is used to decrease reconstructed images error by distributing the error to neighboring pixels. Diffusion filter is responsible of this distribution. This error is written below as a formula & Figure 19 visualizes the work flow mentioned above:

$$error(i,j) = in(i,j) - T(in(i,j))$$

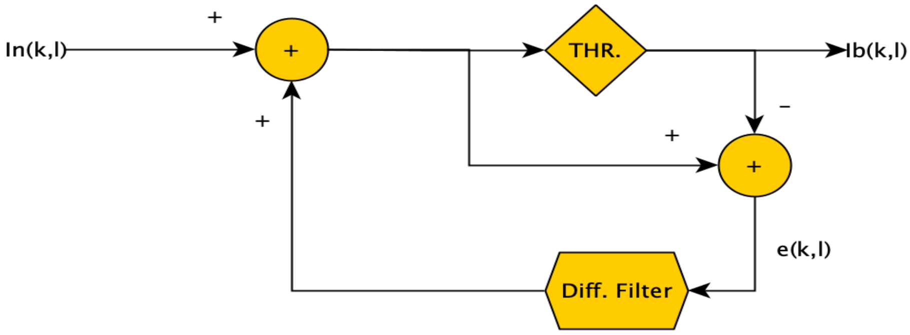

Figure 19: Error Diffusion Method

Diffusion filters to distribute errors to neighboring pixels are depicted below. They are called *Floyd&Steinberg* filter & *Stucki* filter. They are divided by the sum of their elements, therefore both filters are normalized. Lastly, * indicates the pixel that is computed at that moment.

$$Floyd\&Steinberg = \frac{1}{16}\begin{bmatrix} 0 & 0 & 0 \\ 0 & * & 7 \\ 3 & 5 & 1 \end{bmatrix}$$

$$Stucki = \frac{1}{42}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & * & 8 & 4 \\ 2 & 4 & 8 & 4 & 2 \\ 1 & 2 & 4 & 2 & 1 \end{bmatrix}$$

For the sake of comparison, Figure 20 depicts original images. Figure 21 demonstrates the resulting reconstruction of both images when *Floyd&Steinberg* filter is used and Figure 21 demonstrates when *Stucki* is used as the differing filter. Obviously, reconstructed images do have a higher visual quality, having lower error rates, in comparison to the previous section results. Considering results of ordered matrix with dispersed dots that had the highest quality visual representation, and error diffusion method ensures even better visuals in terms of human visual system.

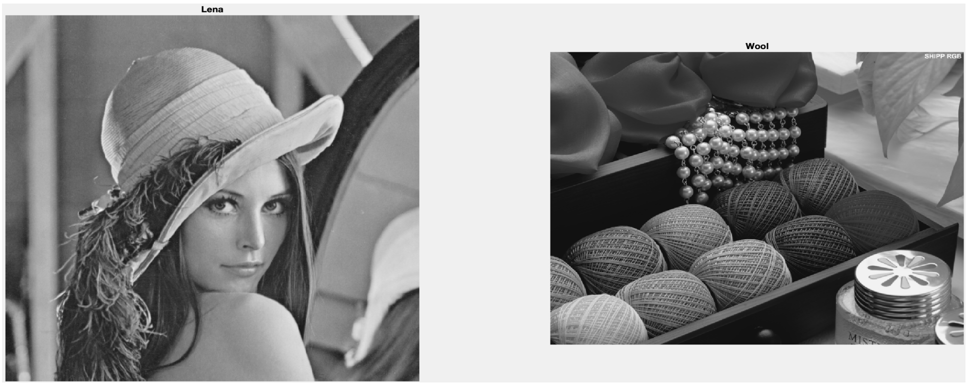

Figure 20: Lena & Wool : Original

Figure 21: Lena & Wool : Error Diffusion Method *Floyd-Steinberg*

Figure 22: Lena & Wool : Error Diffusion Method *Stucki*

9 Error & Complexity Analysis

After the reconstruction of images, mean squared error and complexity analysis might give us another perspective on the reconstruction itself. Results in Table 1, should be considered as a conflict between the human visual system and computational approach. Even though methods to the end were producing better visual quality representations, their mean squared errors were worse than the first two approaches, where we used a fixed or random threshold.

Table 1: Mean Squared Errors: $(\frac{1}{N^2})\sum_{k,l\in[0,N)}(I_b(k,l)-I_n(k,l))^2$

T=128		Lena	Wool
Fixed		8.470286598205566e+03	6.170838956404321e+03
Random	Ns. = 10	8.488934173583984e+03	6.183145018325618e+03
Ordered	S	1.387339280700684e+04	1.170464773823303e+04
Centered	C6	1.386564534759522e+04	1.170315774739583e+04
	E6	1.386272079086304e+04	1.170255040750386e+04
Diagonal	O8	1.384433341598511e+04	1.166958369984568e+04
Dispersed Dots	D6	1.386200266647339e+04	1.170680640673225e+04
	D4	1.373381165695190e+04	1.167254887635031e+04
Err. Diff.	Fl&St.	1.376322423231131e+04	1.357994762099595e+04
	Stucki	1.143550834399609e+04	1.108430279974937e+04

For convention, let us denote total amount of pixels as N^2. Therefore, doing computation on each pixel distinctly corresponds to $\Theta(N^2)$. Therefore, we can say Fixed and Random Threshold complexities are $\Theta(N^2)$. Ordered Threshold has $\Theta(N^2+M)$ as well, because there is a filter that is transformed to same size with the main image and each corresponding pixel are compared. Quantization step is also taken into consideration as it stands for the M in the complexity analysis. In some cases, where N is small, M may be complexity-wise greater than N. Centered, Diagonal and Dispersed Dots methods are using different filters but utilising the same approach as Ordered Threshold, therefore are complexity of $\Theta(N^2+M)$ as well. Lastly, Error Diffusion Method has the highest complexity, because even though there is a pixel-wise comparison, there exists also a error distribution around neighboring pixels. Therefore, let us denote window size as W, which is the size of filter. Complexity for the error diffusion algorithm would be $\Theta(N^2W+M)$.