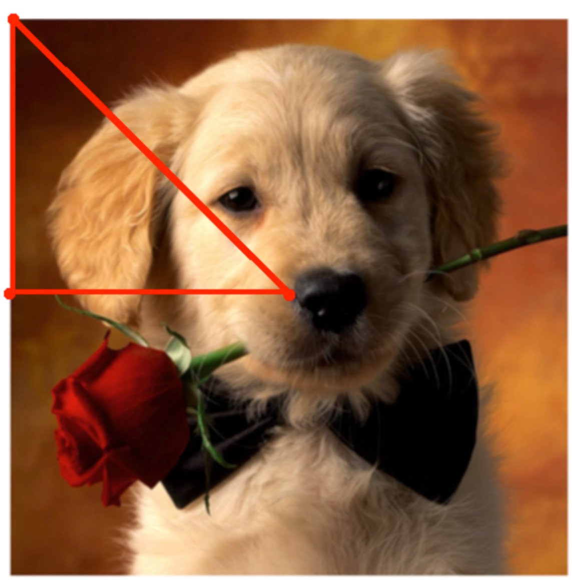
EE569 Report

Problem 1: Geometric Image Modification

(a) Geometrical Warping

The intent of this problem is to duplicate the warping as shown in the following images.

This result can be achieved by employing triangle mapping as described in the Discussion section. For a section, a triangle can be imposed over the image and then given its original and destination positions, can solve for a linear transformation to achieve this result. This process is repeated for all triangle sections such that the end diamond result is achieved. There should be a total of 8 triangle mappings.



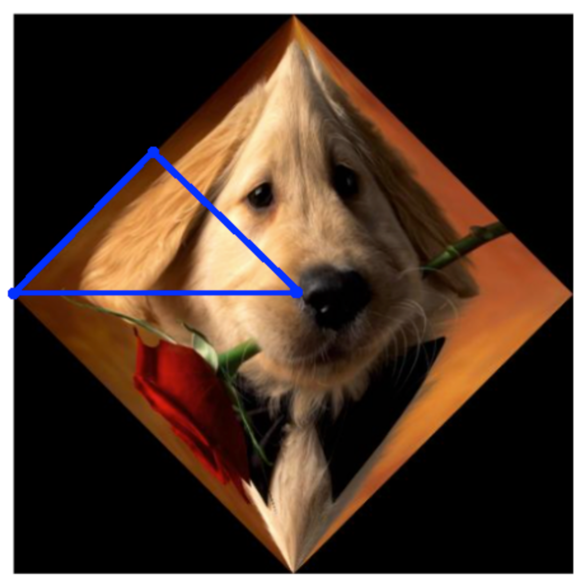


Figure 1-1 and 1-2 show the results of using triangle mapping in this fashion.

Figure 1-1. Triangle mapping transformation result on Kitten_1

The effects of triangle mapping distortion can can be seen on both the images. Because there is only warping on the diagonal axes, this is where there is the most effect. The results do not give a necessarily ‘fish-bowl’ effect, although it does linearly transforms each triangle segment proportionally by each triangle, i.e. each triangle mapping transforms the image segment equally but in different directions. This symmetry can be seen in figure 1-1.

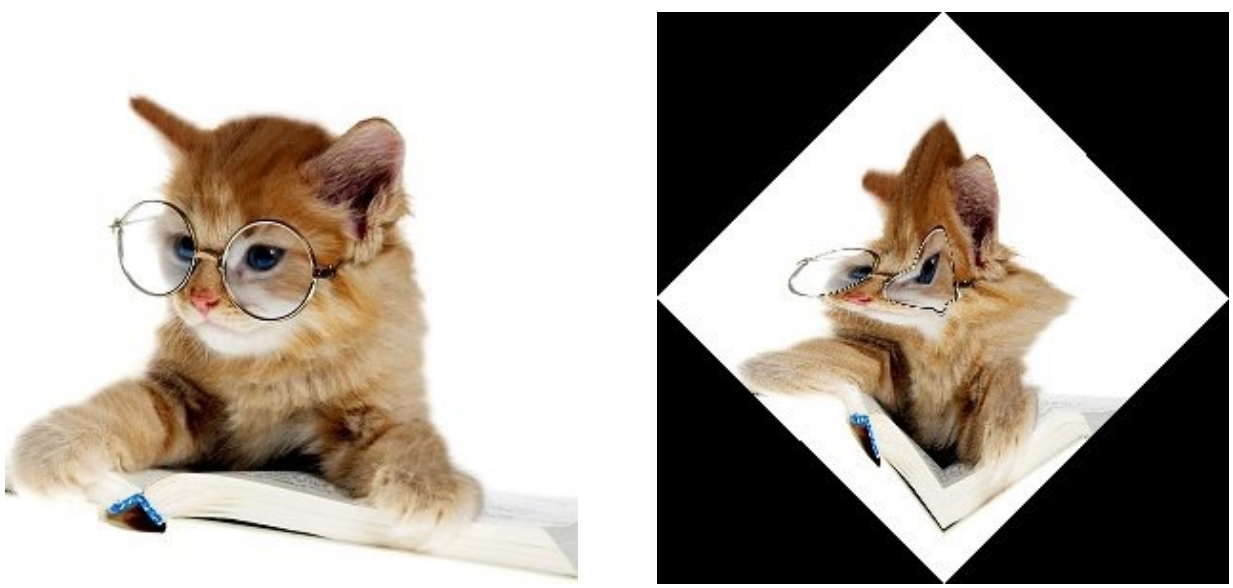


Figure 1-2. Triangle mapping transformation result on Kitten_2

(b) Puzzle Matching

For puzzle matching, two ‘original’ images are given with squares cut out. These squares are then placed in another image file at a random position, resized, rotated. The purpose is to locate these missing pieces in the image file and perform geometric transformations to place them back in the original images. The given images for the problem are shown as below.

For both images, the coordinates of the corners of the pieces as well as the gaps are found programmatically. Separate transformations are performed on the pieces in order to match the missing squares in the original images. The process for both Hillary and Trump are outlined below. The results of the puzzle matching are shown in figure 1-3.

Figure 1-3. Puzzle matching result for Hillary and Trump

For the Hillary image, the eye is rotated counterclockwise at 15 degrees, then scaled down to the dimensions of the missing piece in the original image, which happens to be 100x100. Scaling down causes a redundancy of pixels, these extra pixels are simply discarded. The image is then iteratively copied and placed at the positions of the missing square as found by the program.

The process for the Trump image is similar, although the rotation involves a small rotation of 4.5 degrees clockwise and then another 90 degree rotation clockwise. These two steps are distinct in the program, as the performance suffered when larger angles were used in the rotation transformation. Instead, the 90 degree rotation was achieved by iterating through the rows of the image from top to bottom and placing them into columns from right to left. Further, the image is then scaled up. Missing pixels are filled in with bilinear interpolation. And finally, the image is copied and placed in the missing square of the original Trump picture.

Without close inspection, the resulting images in figure 1-3 look decent. However, finer details, especially at the boundaries, show some discrepancies. Also the image quality is degraded and the resulting pieces are slightly blurred due to the effects of resizing.

(b) Homographic Transformation and Image Overlay

Homographic transformation is the process of creating a generalized transformation from a certain set of starting positions to end positions, linearly transforming the image as it does so. The image can then be placed over another image to create a realistic. combination image.

The homomorphic images used are shown below.

The text logos are to be placed over the football field. The results are shown in figure 1-4 for the Tartans logo and figure 1-5 for the Trojans logo.

Figure 1-4. Tartans image on Football field overlay

Figure 1-5. Trojans image on Football field overlay

The general process includes solving a matrix H that performs the linear transformations given in the following manner. Since there are only 4 points, but 9 parameters, the H_{33} variable is simply set to 1.

$$\begin{bmatrix} x_2' \\ y_2' \\ w_2' \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \ \ and \ \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \frac{x_2'}{w_2'} \\ \frac{y_2'}{w_2'} \end{bmatrix}$$

The problem is then rewritten into an 8x8 matrix linear transform problem and all the H variables can then be solved.

After the homographic transformation and before overlaying, the image contains some outliers and missing pixels. These are filled in with 3x3 median filtering before overlaying on the football field image.

The results look pretty good, but perhaps the text logo images could adapt the energy of the football field image before overlaying to more smoothly fit the image.

Problem 2: Digital Halftoning

Digital halftoning is the process of representing an image with pixel values of only 1 bit. The naive case of digital half toning is setting a constant threshold and binarizing the image based on this threshold. Other techniques involve using pixel density to create the illusion of color depth. The color depth is represented through density, such that a dense pixel area is equivalent to a high pixel value and a sparse pixel area is equivalent to a low pixel value. This idea is the basis of most digital half toning techniques.

The two procedures used to employ digital half toning is dithering and error diffusion.

(a) Dithering Matrix

Dithering is a process that involves a ‘threshold filter.’ This filter is moved across an image and acts as a threshold for pixel values of the image. The Bayer matrix is used as a dithering matrix. The initial matrix is as shown below, and higher order Bayer matrix can be constructed with the following recursion formula.

$$I_2(i,j) = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix}$$

$$I_{2n}(i,j) = \begin{bmatrix} 4 * I_n(x,y) & 4 * I_n(x,y) + 2 \\ 4 * I_n(x,y) + 3 & 4 * I_n(x,y) + 1 \end{bmatrix}$$

In order to construct a proper matrix, the matrix is normalized with the following formula. Where N is the order.

$$T(x,y) = \frac{I(x,y) + 0.5}{N^2} \times 255$$

The resulting Bayer matrix of order 2 is as follows.

$$\begin{bmatrix} 32 & 159 \\ 223 & 96 \end{bmatrix}$$

Below are normalized Bayer matrices up to order 8.

$$\begin{bmatrix} 8 & 135 & 40 & 167 \\ 199 & 72 & 231 & 104 \\ 56 & 183 & 24 & 151 \\ 247 & 120 & 215 & 88 \end{bmatrix}$$

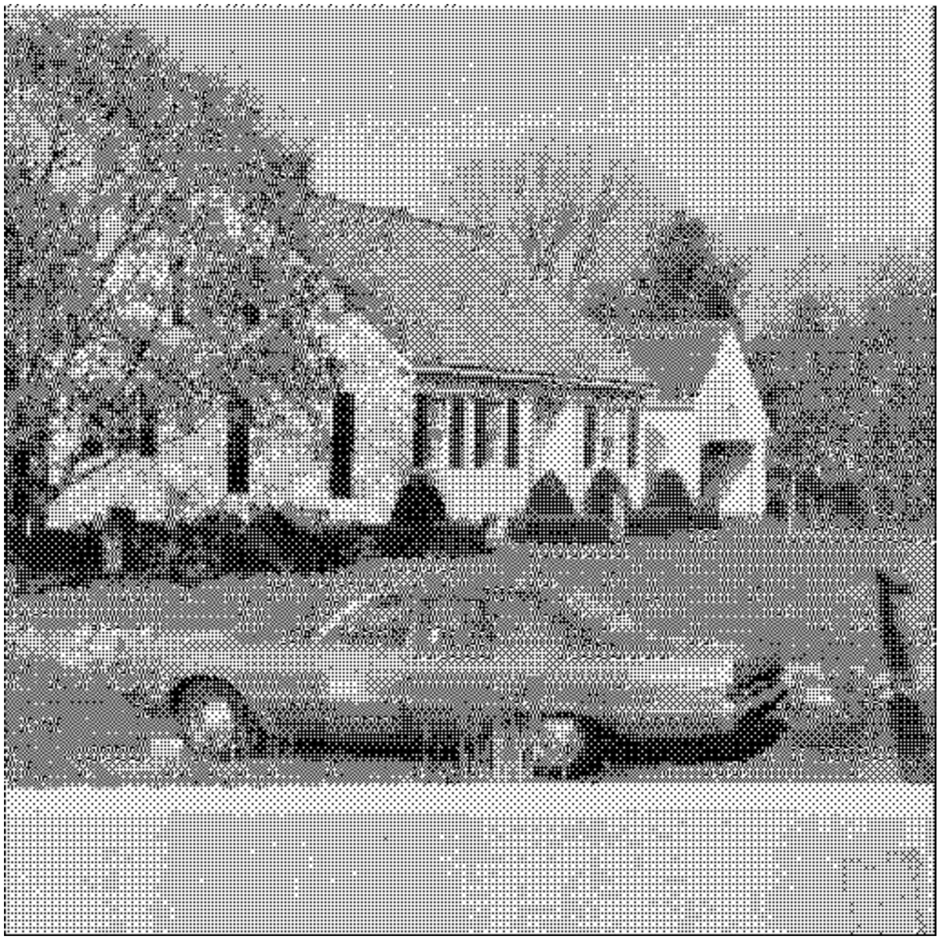


Figure 2-4. I4 dithering on the House image

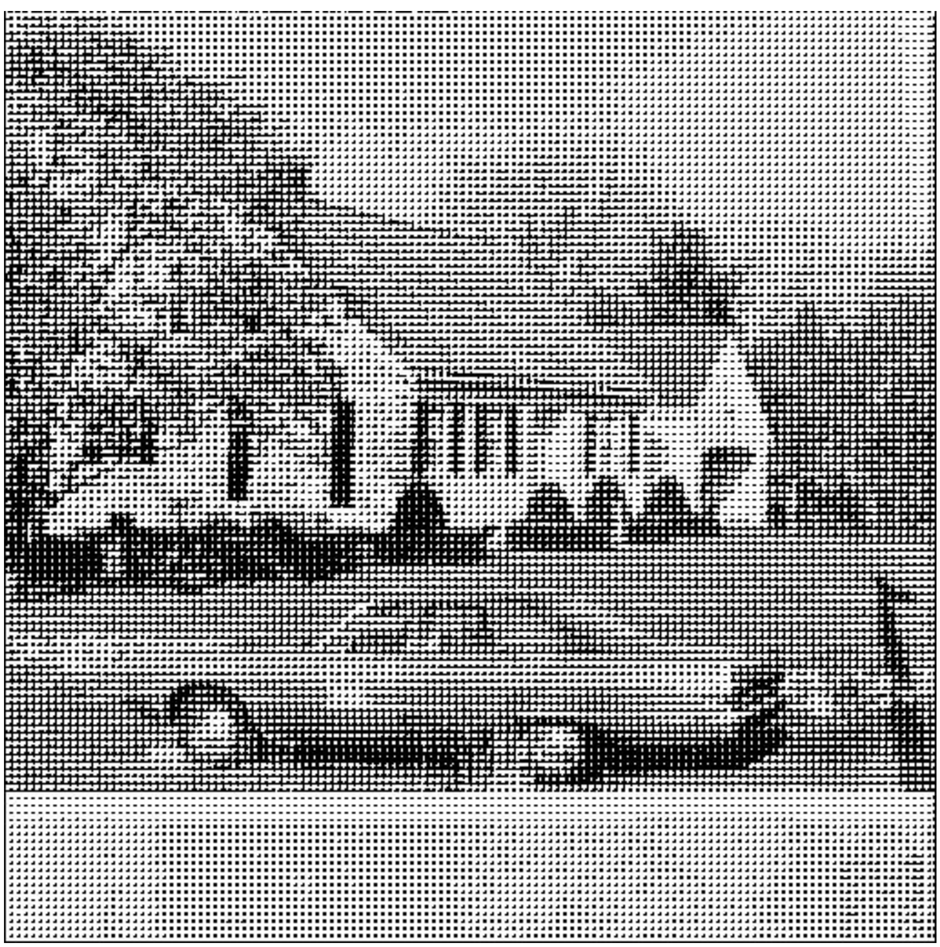


Figure 2-5. A4 dithering on the House image

$$
\begin{bmatrix}
2 & 129 & 34 & 161 & 10 & 137 & 42 & 169 \\
193 & 66 & 225 & 98 & 201 & 74 & 233 & 106 \\
50 & 177 & 18 & 145 & 58 & 185 & 26 & 153 \\
241 & 114 & 209 & 82 & 249 & 122 & 217 & 90 \\
14 & 141 & 46 & 173 & 6 & 133 & 38 & 165 \\
205 & 78 & 237 & 110 & 197 & 70 & 229 & 102 \\
62 & 189 & 30 & 157 & 54 & 181 & 22 & 149 \\
253 & 126 & 221 & 94 & 245 & 118 & 213 & 86
\end{bmatrix}
$$

The target image to be dithered is shown in figure 2.1 and the results of dithering using the Bayer matrices of order 2 and order 8 are shown in figure 2.2 and 2.3, respectively.

Figure 2-1. Original House image

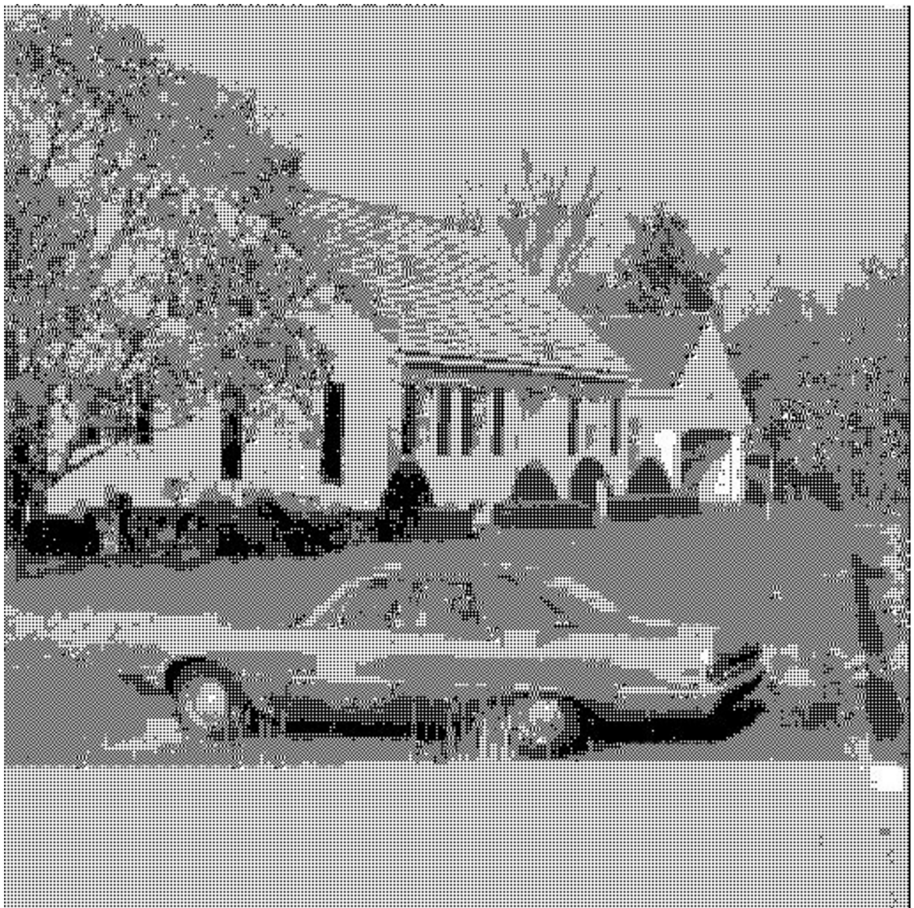


Figure 2-2. I2 dithering on the House image

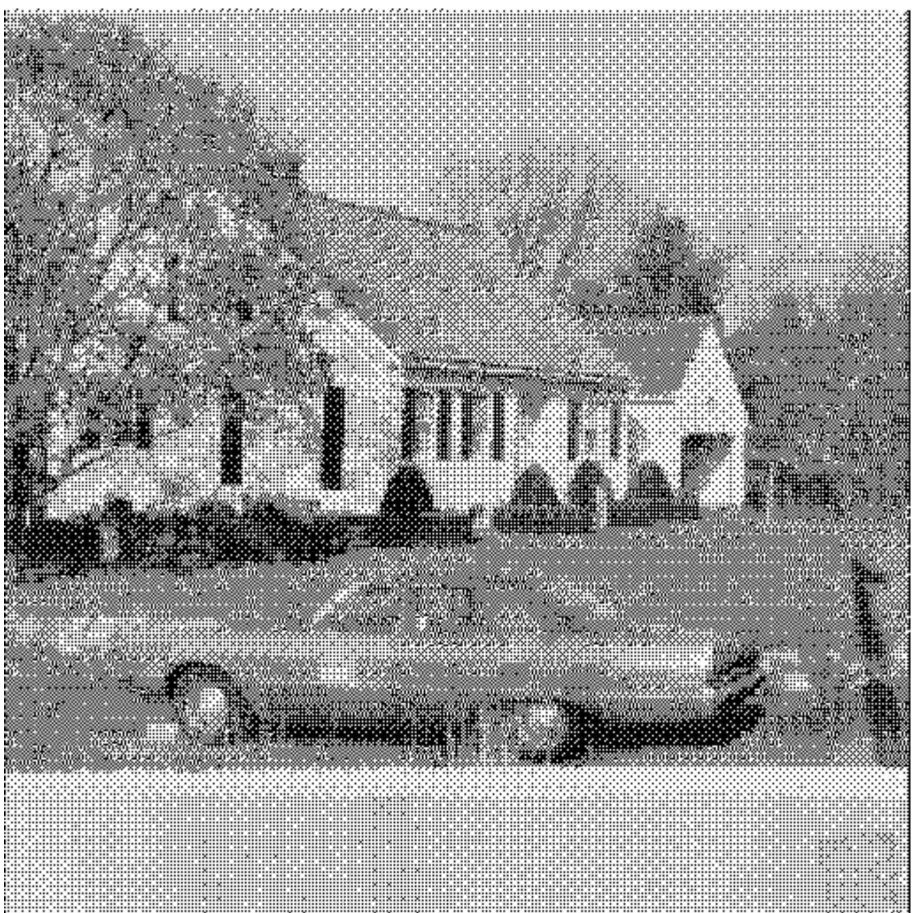


Figure 2-3. I8 dithering on the House image

Below is another custom matrix as an alternative, let us name it A4.

$$\begin{bmatrix} 231 & 167 & 183 & 247 \\ 151 & 56 & 8 & 72 \\ 135 & 40 & 24 & 88 \\ 215 & 120 & 104 & 199 \end{bmatrix}$$

Compare this with the order 4 Bayer I4 matrix.

$$\begin{bmatrix} 8 & 135 & 40 & 167 \\ 199 & 72 & 231 & 104 \\ 56 & 183 & 24 & 151 \\ 247 & 120 & 215 & 88 \end{bmatrix}$$

One can see that from the matrix that the Bayer matrix avoids placing thresholds of similar values close together. It's effect is creating a pixel density that is smooth and not concentrated in small areas. Whereas the alternative A matrix does the opposite and concentrates threshold values of similar magnitude and places them together. The result should be more concentrated pockets of pixels as the majority of pixels would have a value larger than the center 4 values for matrix A. The results for I4 in figure 2-4 and A4 in figure 2-5 exhibit this behavior. Figure 2-5 shows larger 'dots' representing the image. These dots aren't as apparent in figure 2-4. The overall effect, however, is that most high frequency details are filtered from the A4 dithering and so the general picture looks more accurate than I4 dithering, since I4 dithering captures smaller details that may or may not be there. This is perhaps most obvious in the representation of the car in the images. I4 dithering attempts to capture the smaller details of the car, hampering the resolution, whereas A4 ignores this and captures the general contour of the car, resulting in a better representation.

If instead of using 1 bit of information for halftoning pixel values (0, 255), four pixel values are allowed (0, 85, 170, 255), then a different approach may be necessary for ‘quartertoning.’ The proposed method is an adoption of the 4th order Bayer matrix, and separate into three threshold values so a total of 4 values can be achieved. So a 4x4x3 matrix is used as a dithering matrix to threshold each pixel value. The given matrix can be expressed in the following expression.

$$T(x, y, z) = \frac{I(x, y, z) + 0.5}{N^2} \times 255 \times \frac{z}{3}, \text{ where } z = 1, 2, 3$$

Thus, the values of the resulting image are assigned by the following expression.

$$G(i, j) = \begin{cases} 255 & \text{if} \quad F(i, j) > T(i, j, 3) \\ 170 & \text{if} \quad T(i, j, 2) < F(i, j) \leq T(i, j, 3) \\ 85 & \text{if} \quad T(i, j, 1) < F(i, j) \leq T(i, j, 2) \\ 0 & \text{if} \quad F(i, j) < T(i, j, 1) \end{cases}$$

The resulting image is shown in figure 2-6.

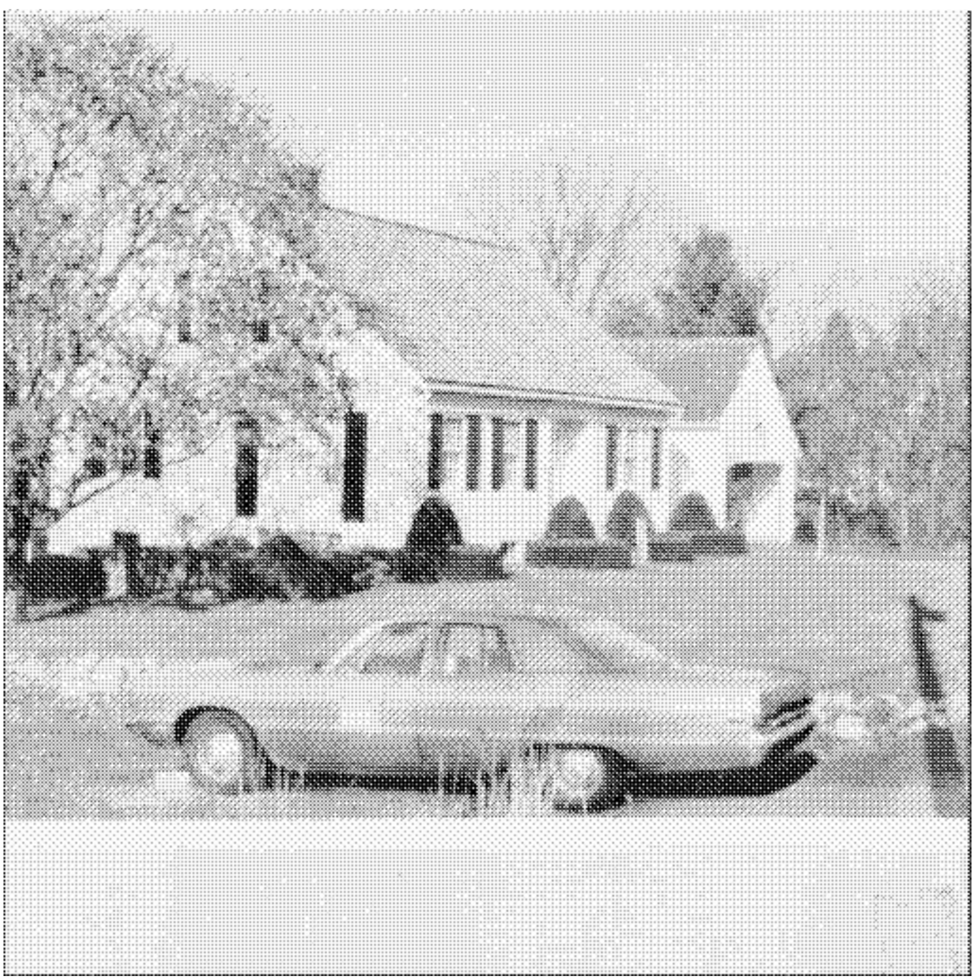


Figure 2-6. 4 value I4 dithering on the House image

The image as shown in figure 2-6 shows a definite improvement from figure 2-4. The structural process is inherently the same as the relative weights of the filter values are the same. This can be seen in the white strip at the bottom of the image, i.e. the dithering pixels have the same structure in figure 2-6 and 2-4. However, the image in 2-6 is qualitatively better due to the extra dimension of precision added to each pixel value. A better use of contrast is used, which is apparent in the areas of shadows. Also, note that the overall energy of figure 2-6 is lower than in figure 2-4, i.e. figure 2-6 looks less dark. This is because there is a larger range of representing information of the image. Values that are 255 for figure 2-4 instead are able to be 170, 85 in addition to 255. This causes the overall energy to decrease, but with more information.

(b) Error Diffusion

Error diffusion is a half-toning method that adds error to future pixels not yet considered. This error corresponds to the quantization error of the current pixel, but is subtracted from future pixels such that they are likely to offset the error of that pixel. For instance, if a pixel with value 188 is halftoned to 255, this error of 67 is multiplied by some constant and then subtracted by its neighboring pixels (that are not halftoned) such that those pixels are more likely to be halftoned down to 0 to offset the initial error. The variable to be considered here is the constants for the errors. These constants are represented in a filter. Three matrices are used to perform error diffusion in this problem.

Floyd-Steinberg matrix (implemented using serpentine scanning, results shown in figure 2-7)

$$\frac{1}{16}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 7 \\ 3 & 5 & 1 \end{bmatrix}$$

Jarvis, Judice and Ninke (JJN) matrix (figure 2-8)

$$\frac{1}{48}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 5 \\ 3 & 5 & 7 & 5 & 3 \\ 1 & 3 & 5 & 3 & 1 \end{bmatrix}$$

Stucki matrix (figure 2-9)

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

Figure 2-7. Floyd-Steinberg Error Diffusion with Serpentine scanning on House image.

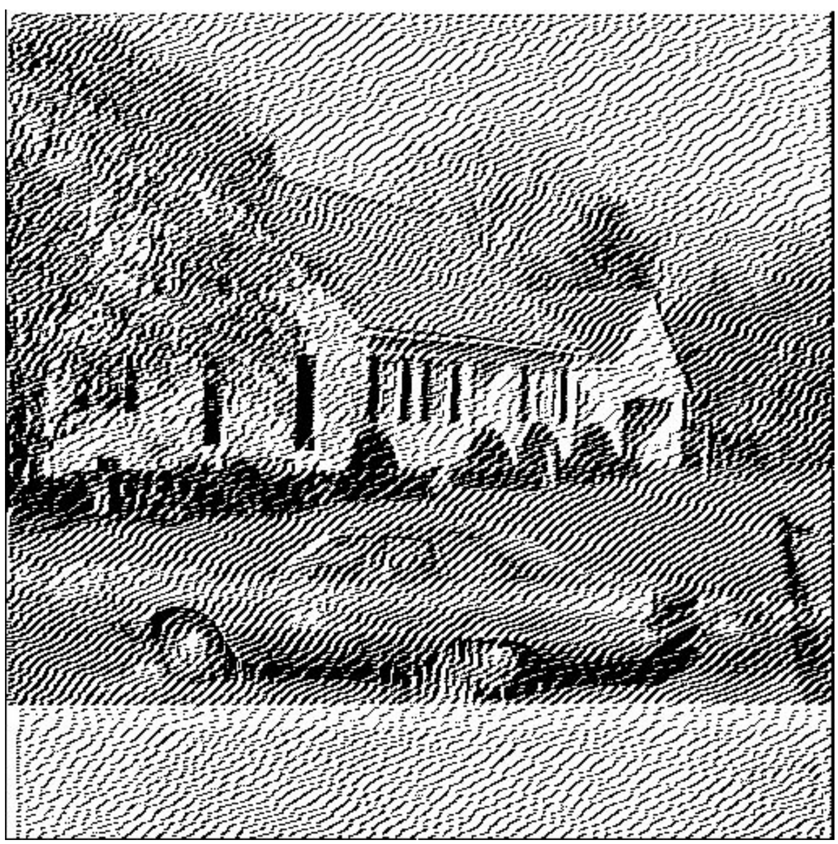


Figure 2-8. JNN Error Diffusion on House image.

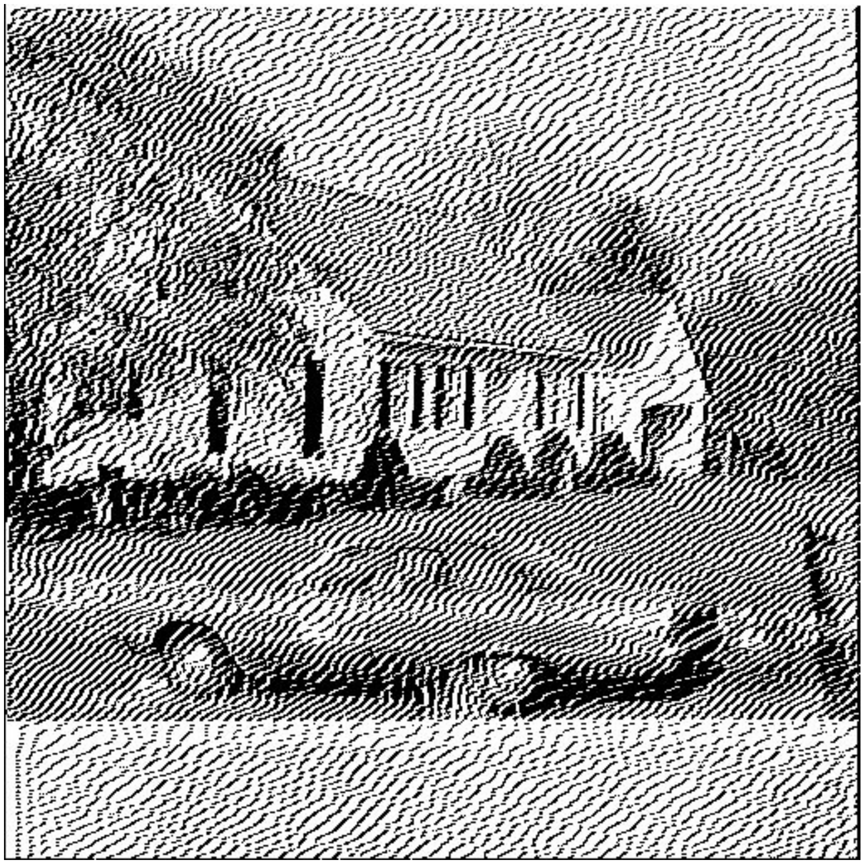


Figure 2-9. Stucki Error Diffusion on House image.

Note that the images from figure 2-9 and 2-8 look very similar. This is because the error diffusion matrices are the same shape and have similar magnitude structure. However, it can be seen that the images are not the same when observing the houses. The black lines appear more numerously in Stucki than in JNN because the propagated error is more for Stucki than for JNN.

While the error diffusion matrices maintain the overall energy of the image, the resulting images appear less qualitatively accurate than that from the dithering process. The dithering operations seem to retain a better representation of the original image, despite not considering quantization errors.

Problem 3: Morphological Processing

(a) Rice Grain Inspection

This problem is given as follows, given the below image, count the number of rice grains, compare the sizes and then detail a way to categorize the rice grains.

By inspection, it is clear to see that there are 11 clusters of rice grains each with 5 rice grains for a total of 55 rice grains.

The procedure employed in the program is as follows.

1. Convert the RGB image to grayscale
2. Binarize image
3. Filter the image
4. Count the number of rice grains
5. Sort the results to compare sizes.

The conversion to greyscale is achieved with the following formula. The result can be seen in figure 3-1.

$$Greyscale = 0.3 \times R + 0.59 \times G + 0.11 \times B$$

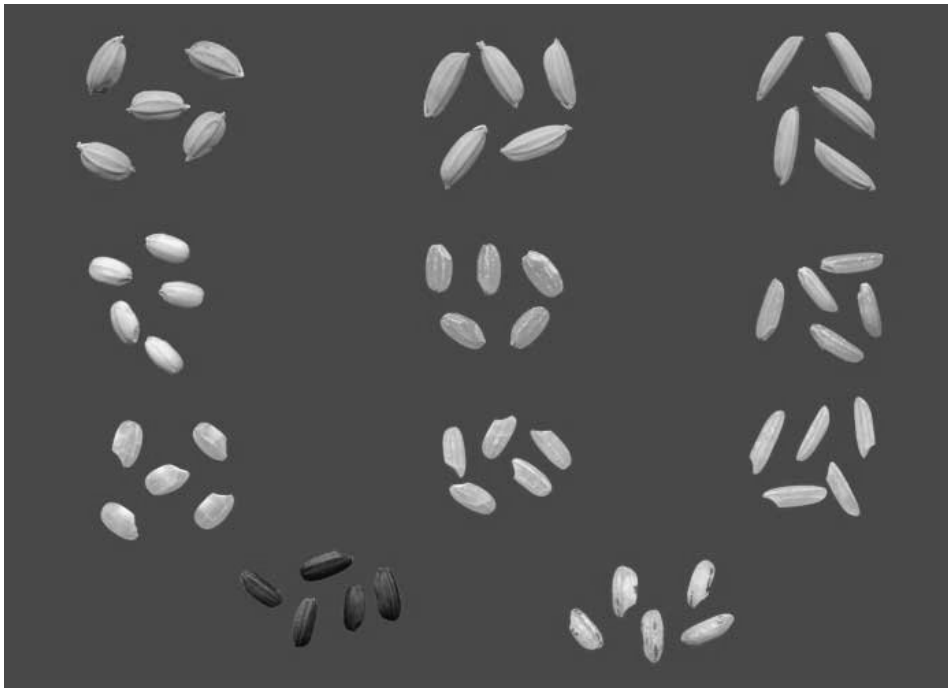


Figure 3-1. Grayscale conversion of Rice image

The image is then binarized using a global mean threshold. The threshold is calculated by considering the mean value of the entire grayscale image. In the future, adaptive local mean calculations for thresholds are likely to give a better results. The result can be see in figure 3-2.

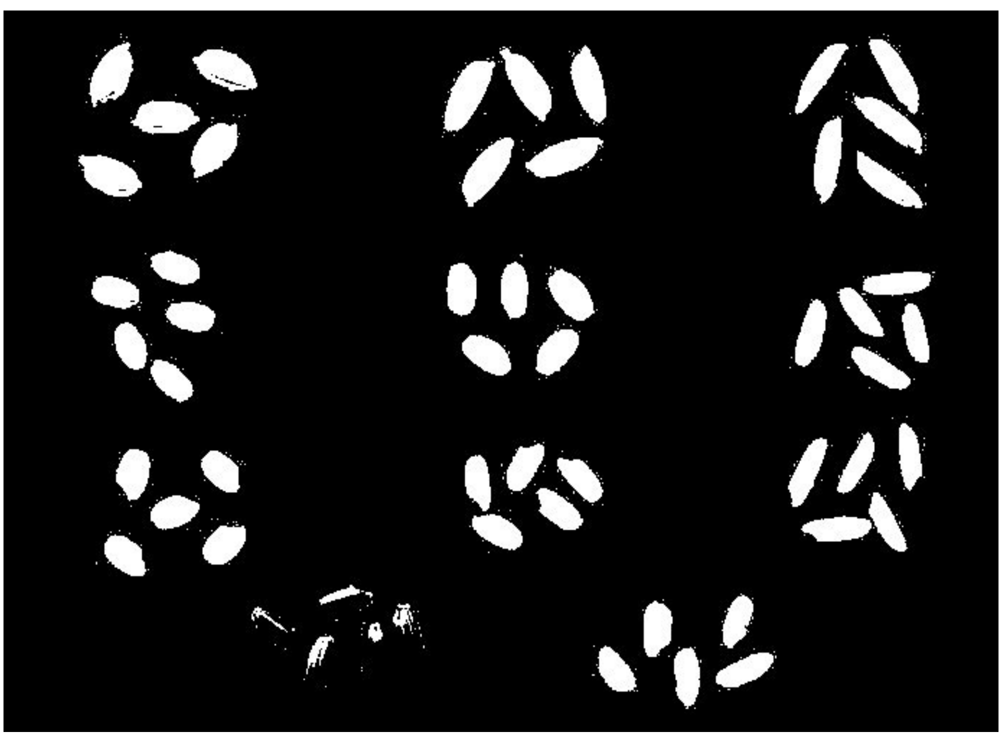


Figure 3-2. Binarization of grayscale Rice image

On close inspection, there are many speckles and dots in the binarization as a result of the binarization algorithm. Median filtering is applied multiple times in order to ensure that no outliers are present, which may affect performance in future steps. The image is shown after filtering in figure 3-3.

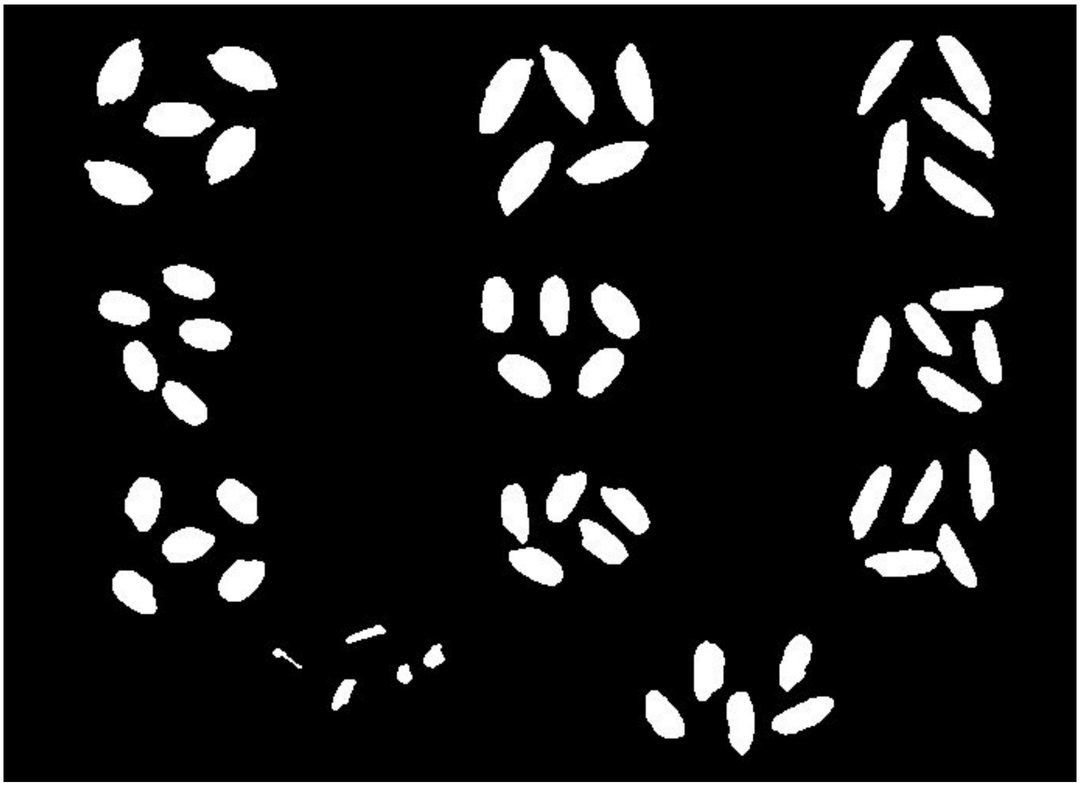


Figure 3-3. Filtering of binarized Rice image

From this point on, no more processing is performed on the image and analysis is performed with connected component labeling.

As printed from the program, the connected component labeling results in a correct output of 55 objects. The algorithm also includes counting the sizes of the objects, which is sorted and then printed to the console. The object number is labelled by counting each object from the top left corner horizontally and then vertically. The size is calculated by counting the total number of pixels associated with a single object. Needless to say, the object with the largest number of pixels is the largest object.

The results printed out by the algorithm that sorts the rice grains by size is relatively accurate although there are some misplaced rice grains. The error associated with this is most likely due to the loss of information from binarization as well as filtering. This is most obvious in the bottom left cluster, which has the most significant loss of information.

A categorization technique would be to consider the average size of each cluster. then consider the shape, i.e. the elongation of each rice grain and elliptical correlation.

Further, if access to color is possible, then this adds another dimension to be able to classify rice grains.

Side note::
I tinkered around with morphological processing a lot but it didn't seem to work. Most likely it is an implementation error. But due to the time constraints I had decided to avoid figuring out what the error was and instead implemented a different algorithm that would get the same results.

Because I did implement morphological processing correctly, I was not able to complete problem 3b.

Jimmy Chen