Aakash Shanbhag

adshanbh@usc.edu

3205699915

AAKASH DHANANJAY SHANBHAG

Contents

Aakash Shanbhag

adshanbh@usc.edu

3205699915

Aakash Shanbhag

adshanbh@usc.edu

3205699915

Problem 1: Geometric Image Modification.

1.1 Motivation and Abstract.

Geometric Image modification and spatial warping techniques are a crucial part of the image processing and computer vision domain in which desired shapes and configurations can be obtained on the basis of the Mathematical calculations. Basic techniques include scaling, translation, rotation, down-sampling and up-sampling using bilinear interpolation along with affine and perspective transforms which are largely used in image homographic techniques where warping and morphing is done precisely. Homographic techniques can be specially used when 2 camera perspectives are taken into consideration and the point of view changes, which increase the degree of freedom (d.o.f). Such modifications can be well adept are generally used in medical imaging, computer graphics , Augmented Reality applications to achieve seem less blending and better visual content.

In the first part of the question- linear interpolation was utilized to obtain the desired output with 4 quarters broken down for interpolation in the horizontal direction only. In the second portion of the problem missing puzzle piece was reinserted in the location desired by using the scaling, rotation and translation techniques with reverse mapping. In the last part of the question the homographic techniques were carried out to overlay an image according to the values of the transform matrix. It is to be noted that the reverse image warping is crucial for all the above methods discussed along with re-sampling by interpolation techniques.

1.2 Approach and Procedure.

There are 3 parts to this problem and each part is dealt with in a different manner as discussed below through examples and figures with detailed discussion on the implementation of the algorithm for each of the cases.

1.2 (a) Geometric Warping

The steps involved in carrying out the desired warping as seen in the dog image is as follows:

1. The original cup image is broken down into 4 quarters are seen below in the figure. As the above image is a square, we can easily obtain its quarters by dividing the height and width into 2 halves.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

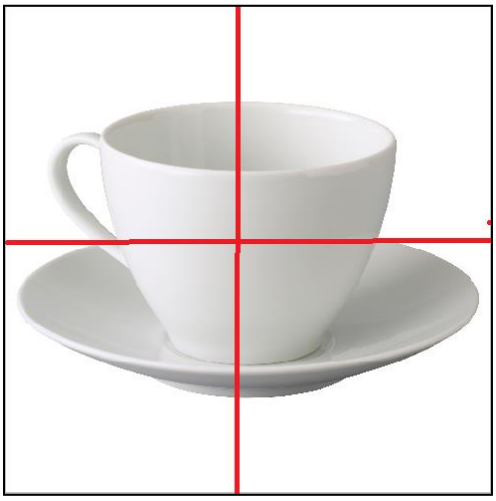
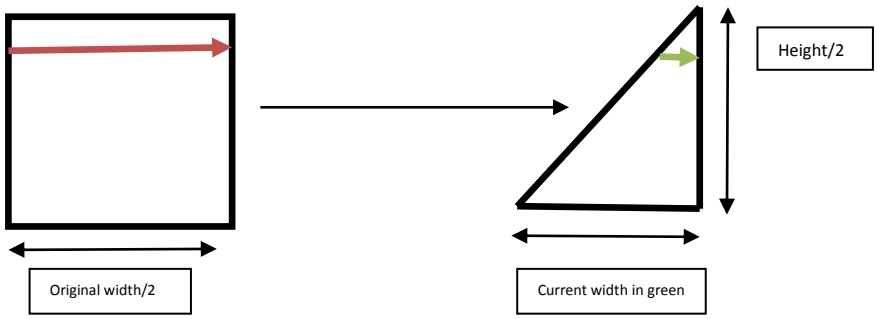


Fig 1.1: Dividing image into 4 quarters.

2. Linear interpolation is carried out in each of the 4 quarters in the horizontal direction and each point is linearly interpolated.
3. Consider the 1st quarter portion of the image, we loop across the entire first quarter, we choose the old width to be equal to image width of the quarter(width/2) which gets remapped to the desired shape as seen below: In this case the red width indicates original width=width/2 getting remapped to green width which is equal to the



4. Similarly the linear interpolation algorithm can be used for the rest of the 4 quarters as well where only the shape of the desired output would change and would give a different width (green).
5. The formula required to match the corresponding points according to the above mentioned method is as follows:

$$g(x,y) = f(x,y) \times (1 - \Delta y) + f(x,y+1) \times (\Delta y)$$

Aakash Shanbhag

adshanbh@usc.edu

3205699915

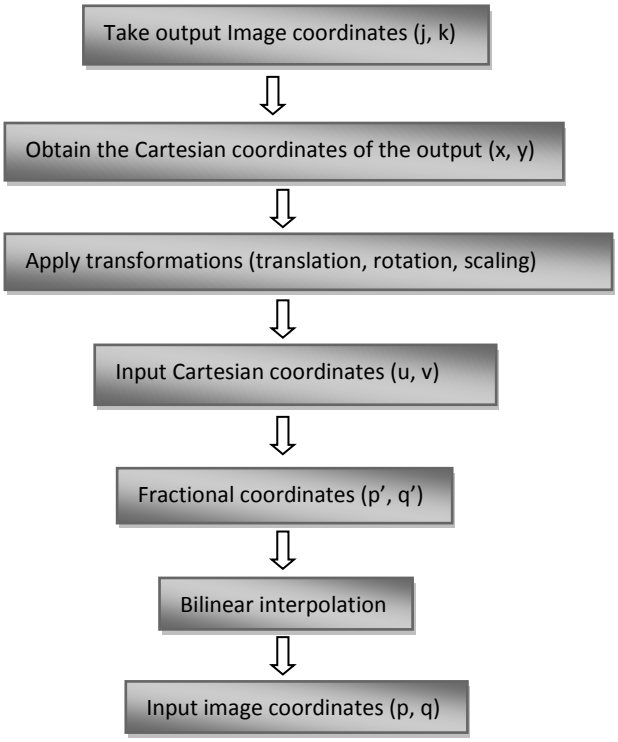
Where $g(x, y)$is the output point and $f(x, y)$ is the input point and Δy is the fractional difference between intermediate points.

6. Similar process is carried out with all the 4 quarters and since only linear interpolation is needed, the points inside the square are interpolated in such a manner that they fit the triangle.
7. All the quarters are worked according to a similar process with linear interpolation in a single direction (horizontal).
8. Each and individual portion is separately dealt with and the remaining portion in the image is set to black (0).
9. General image memory allocation needs to be carried out dynamically and de-allocations also need to be followed correspondingly.

1.2 (b) Puzzle Matching.

The steps involved in puzzle matching of the Hillary and Trump image are as follows:

1. The general reverse mapping techniques is utilized in this problem which involves the conversion from the image coordinates to the Cartesian coordinates from the output image to the input image with the following process:



Aakash Shanbhag

adshanbh@usc.edu

3205699915

2. The image coordinates of the unknown Hillary image can be found by checking the image and finding the 4 coordinates in the first half of the piece image. This is done by first finding the top position of the Hillary piece image and correspondingly checking the neighborhood for white background, the 4 unknown points of the Hillary image can be easily obtained.
3. These points are image coordinates which need to be converted to Cartesian coordinates as follows:

$$x = i + 0.5$$
$$y = imageheight - 0.5 - j$$

   Where (x, y) are the Cartesian coordinates and (i, j) are image coordinates.
4. Translate by a factor tx and ty to reach the origin of the Cartesian plane.

5. Calculate the angle of rotation needed to rotate with respect to the points shown below. The angle obtained needs to be in radians. The angle in blue= π/2-arctan (difference of the y coordinates of blue marked points/ difference in the x coordinates of the blue marked points). Anticlockwise convention is always taken into consideration.

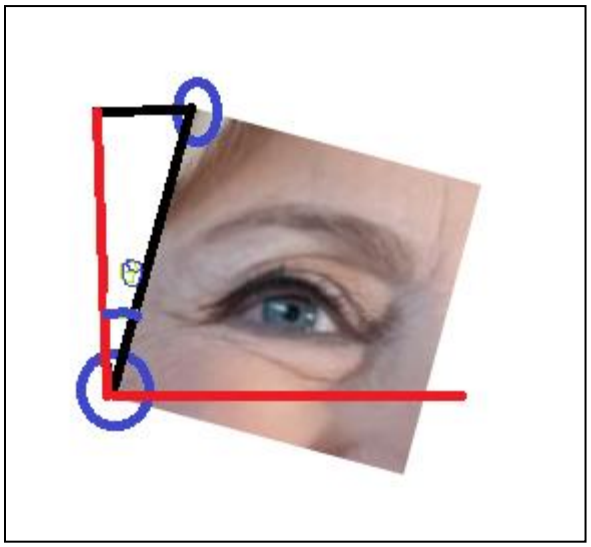


Fig 1.2 : The angle calculation for rotation.

6. Each of the points inside this Hillary eye piece are hence rotated by this angle obtained by the formula below:

Aakash Shanbhag

adshanbh@usc.edu

3205699915

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

7. The above rotation is applied since we are following the reverse mapping concept and each of these points are rotated by that angle in order to obtain the desired rectangle .
8. Scaling is done by the formula discussed below:

$$u = \frac{x}{S_x}, v = y/S_y.$$

9. Each and every point is scaled by the factor mentioned below and hence each point can be set to the size of the missing piece which is equal to 100, 100. Hence the scale factor is calculated by the formula :

$$S_x = (100 + 2)/(width\ of\ the\ piece\ image)$$
$$S_y = (100 + 2)/(height\ of\ the\ piece\ image)$$

10. The addition of the 2 is done in order to avoid the corner blurring present.

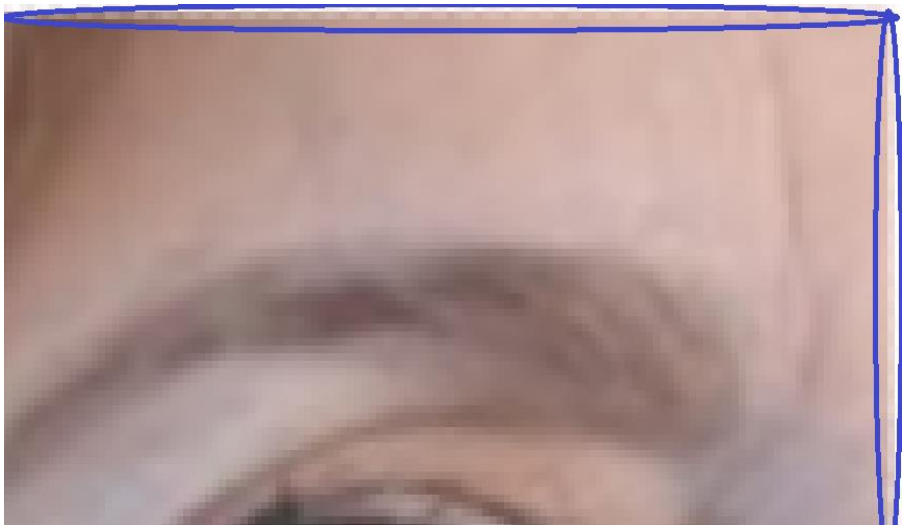


Fig 1.3: Blurring present at edges.

11. Now the Cartesian coordinates are transformed back to the image coordinates by the exact opposite formula which was used to convert vice versa.

$$p' = u - 0.5$$

$$q' = imageheight - 0.5 - v$$

Aakash Shanbhag

adshanbh@usc.edu

3205699915

12. Since the above obtained values can be fractional we need to use bilinear interpolation in order to recover integer image coordinates (p, q) which can be hence mapped back to the locations of the missing puzzle.
13. In order to find the find the missing location of the Hillary and Trump image, we scan through the entire image and find the first white pixel and check for the next 100 pixels if they are white since we know the height and width of the puzzle piece to be matched in 100x100.

14. Once this condition is met, break the loop and set that location as the first point of the puzzle image to which the rotated and scaled image is one to one mapped.
15. The puzzle is matched by the above process and the other locations of the Hillary and Trump image are left unaffected.
16. Hence all the operations are carried out in the piece image and mapping is done on the puzzle of Hillary and Trump image.
17. In case of the Trump Image only difference that occurs in the process is that of angle calculation which turns out to be equal to = -π/2-arctan (difference of the y coordinates of blue marked points/ difference in the x coordinates of the blue marked points.)
18. All the other steps remain same for the Trump image and hence puzzle fitting can be easily carried out for both the cases.

1.2 (c) Homographic transformation and Image Overlay.

The Homographic transformation procedure is stated below. Images of point in a plane , from two different camera viewpoints, under perspective projection (pin hole camera model) are related by Homography as follows:

$$P_2 = HP_1.$$

Where H is 3x3 homographic transformation matrix, P_2 and P_1 denote corresponding image points in homogenous coordinates before and after the transform respectively. Specifically, we have

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

Where $H_{33} = 1$ and so there are 8 parameters to be determined. 4 point pairs in two images are used to build 8 linear equations which solve the above matrix. Backward mapping and bilinear transformation are used in this process to recover the desired output.

According to the above detailed explanation, I approached this question in the following manner:

1. 4 image coordinate points were chosen from the Trojan image as well as the field image for generating a map and developing 8 linear equations.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

2. The image coordinates of both the images were converted to Cartesian coordinates by the formula.

$$x = i + 0.5$$

$$y = imageheight - 0.5 - j$$

Where (x, y) are the Cartesian coordinates and (i, j) are image coordinates.

3. The mapping necessary for the above mentioned process is explained in detail below:

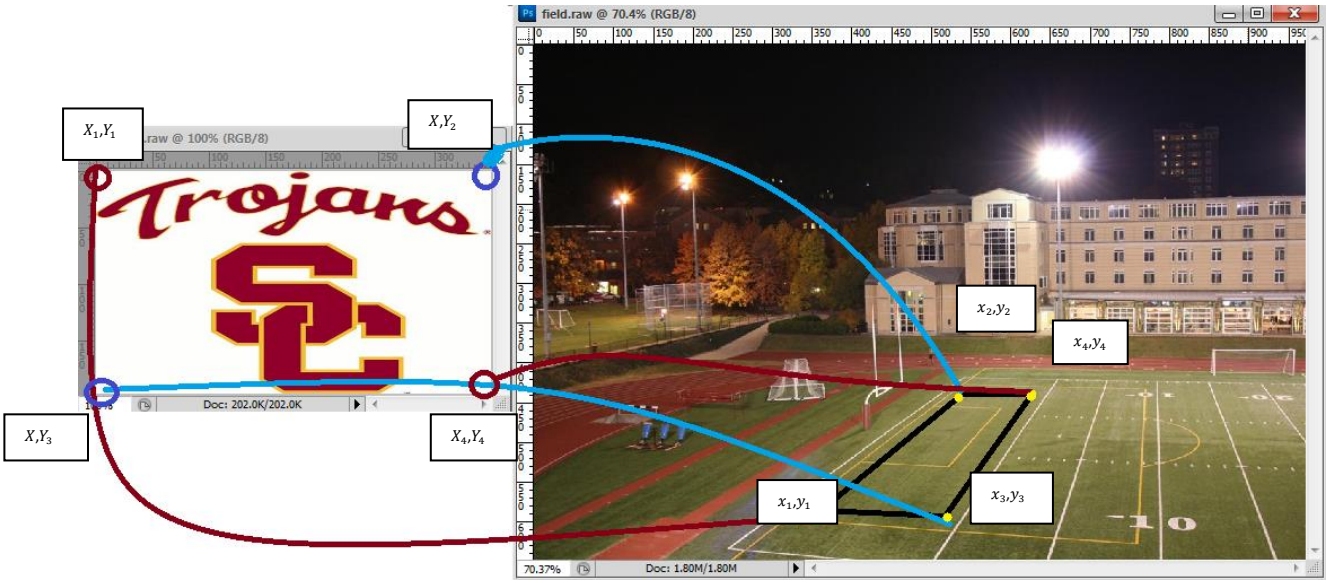


Fig 1.4: Coordinate mapping between the field image and the Trojan Sc image.

The table below shows the transformation of the image coordinates to the Cartesian coordinates and the corresponding 4 points are mentioned below where a=1, 2, 3, and 4:

j_a	i_a	x_a	y_a
356	584	356.5	63.5
541	440	541.5	207.5
522	595	522.5	52.5
630	436	630.5	211.5

Table: Image coordinates (j, i) and corresponding Cartesian coordinates (x, y) for the field image.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

J_a	I_a	X_a	Y_a
1	1	1.5	195.5
349	1	349.5	195.5
1	196	1.5	0.5
349	196	349.5	0.5

Table: Image coordinates (J, I) and corresponding Cartesian coordinates (X, Y) for the Trojan image.

4. The corresponding homographic transformation matrix parameters are found out by solving the 8 equations and 8 unknowns by the matrices defined below:

$$\begin{vmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1X_1 & -y_1X_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2X_2 & -y_2X_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3X_3 & -y_3X_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4X_4 & -y_4X_4 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1Y_1 & -y_1Y_1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2Y_2 & -y_2Y_2 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3Y_3 & -y_3Y_3 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4Y_4 & -y_4Y_4 \end{vmatrix} \cdot \begin{vmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{vmatrix} = \begin{vmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{vmatrix}$$

$$a = H_{11},\ b = H_{12}, c = H_{13}, d = H_{21}, e = H_{22}, f = H_{23}, g = H_{31}, h = H_{32}$$

Source: http://www.corrmap.com/features/homography_transformation.php

5. The Homographic transformation matrix thus obtained is equal to :

$$\begin{bmatrix} X \\ Y \\ W \end{bmatrix} = \begin{bmatrix} 0.0692 & 1.0269 & -88.4776 \\ -1.0478 & 0.7099 & 510.723 \\ 0.005 & -0.0039 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

6. The value to be mapped back is defined by X'=X/W and Y'=Y/W.
7. Once the four points of the field image are known it is important to apply the transformation to points or addresses that fall inside the region specified. This can be done by finding out the equations of lines and checking if each of the corresponding values lies inside the boundary that specifies the 4 points.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

8. According to the condition of the points inside the field boundary, the above transformation matrix is applied to those that lie inside the boundary.
9. Reverse image mapping is again carried out in this case and the Cartesian coordinates are again converted back to image coordinates using the formula:

$$p' = u - 0.5$$
$$q' = trojanheight - 0.5 - v$$

10. Since the above obtained values can be fractional we need to use bilinear interpolation in order to recover integer image coordinates (p, q) which can be hence mapped back to the locations of the field image.
11. The white background of the Trojans image is removed by proper thresholding conditions set up.
12. Dynamic memory allocation and de-allocation is carried out and the Trojan image is set up on the field as desired.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

1.3 Experimental Results.

1.3 (a) Geometric Warping

The input image and the corresponding step by step conversion of the cup.raw images to the desired output are shown below by using the steps discussed in 1.2 (a).

Output for Cup1.raw

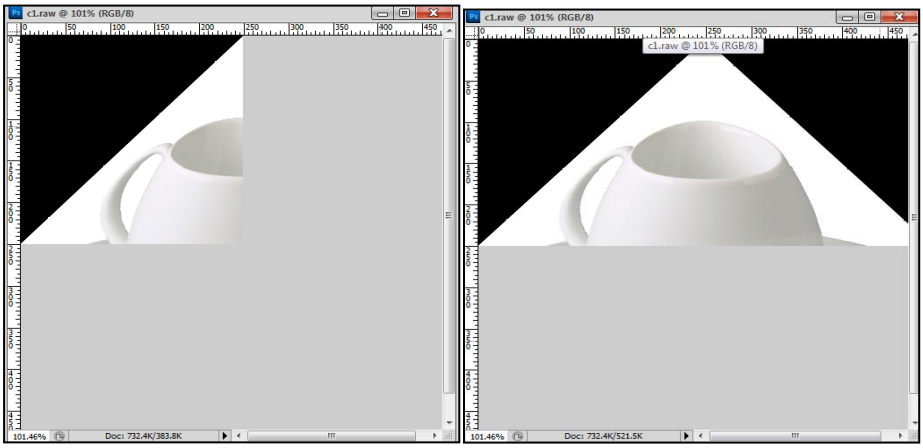


Fig 1.5: Step by step output generation based on the each quarter piece.

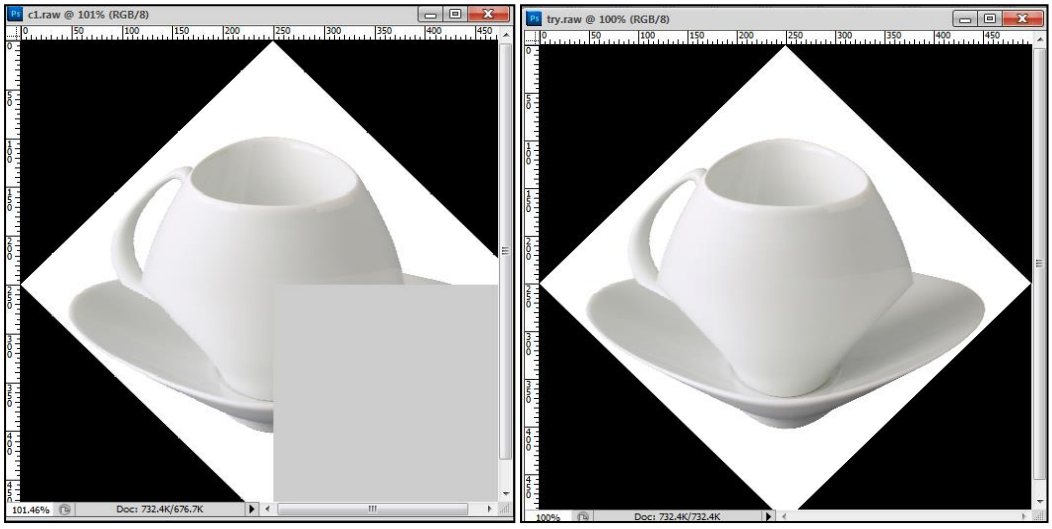


Fig 1.6: Step by Step output generation based on the last quarter piece.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

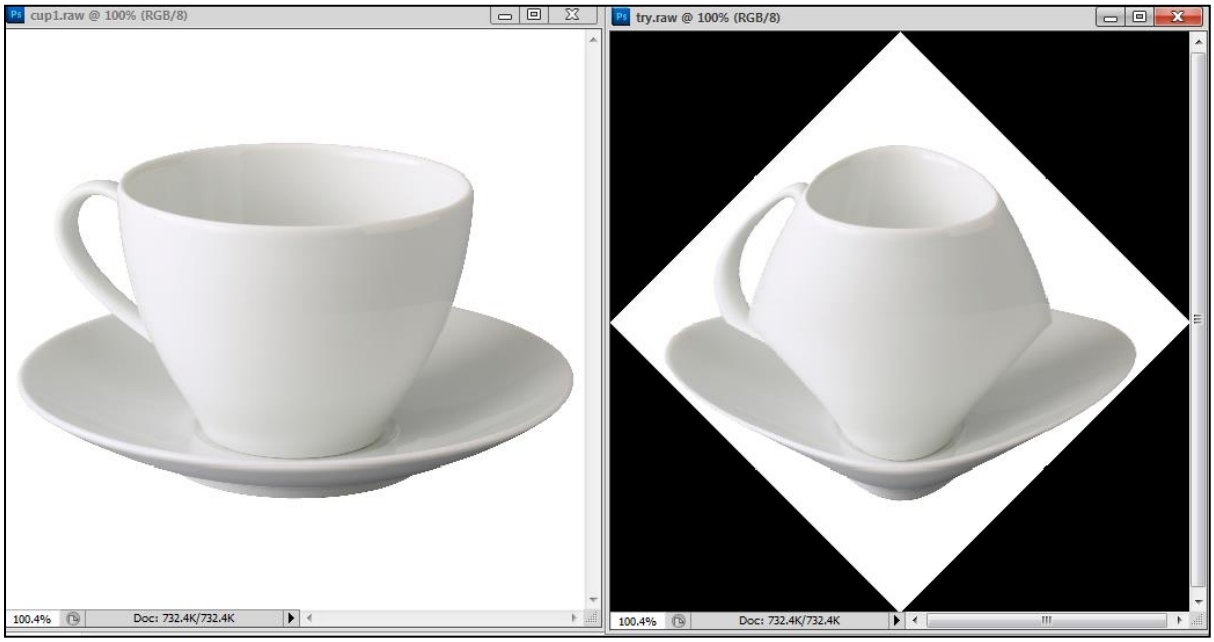


Fig 1.7: Final output generated by linear interpolation based on each of the piece for cup1.raw image .

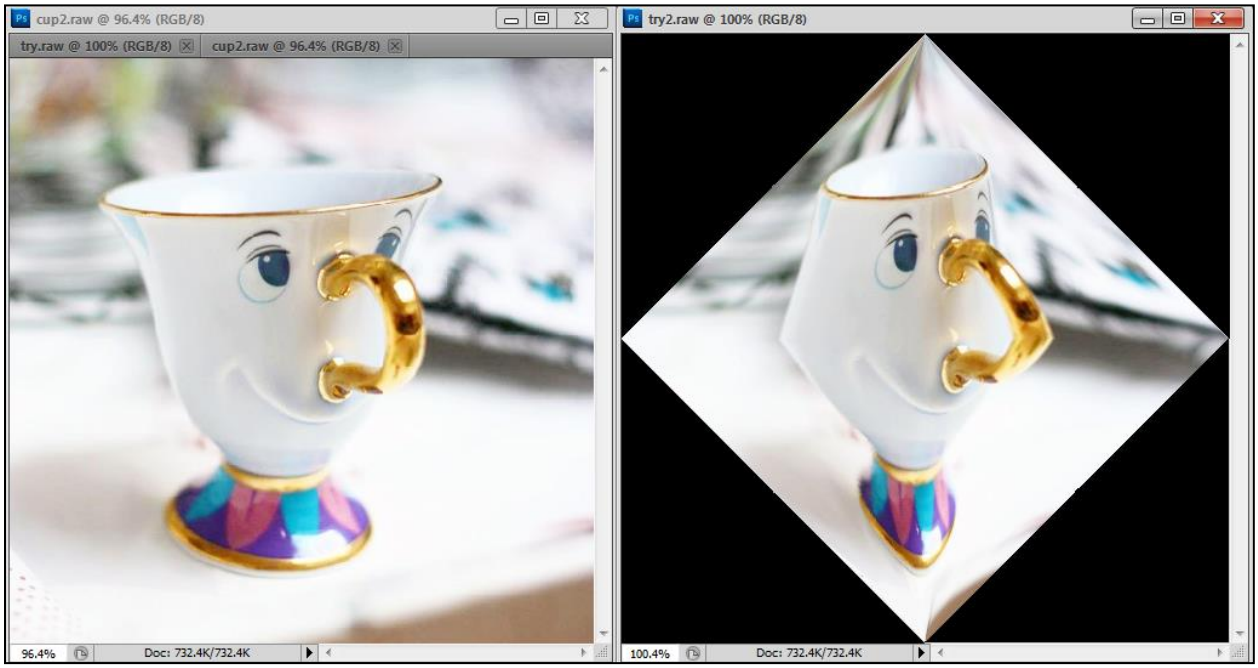


Fig 1.8: Final output generated by linear interpolation based on each of the piece for cup2.raw image.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

1.3 (b) Puzzle Matching.

The input image and the corresponding step by step matching process are shown below by using the steps discussed in 1.2 (b). Intermediate outputs are also displayed.

Case 1: Hillary Image:

According to the steps mentioned in above, the image coordinates of the piece image are obtained and corresponding Cartesian coordinates are also obtained along with the angle specified. Finally the unknown top location of the Hillary Puzzle location is also found marked in red and its rotated and scaled image are also shown below.

```
Locations of Hillary piece image are as follows:
topvalues:96 57
rightvalues:241 96
leftvalues:57 200
bottomvalues:199 240

topvalues Cartesian:96.5 442.5
rightvalues Cartesian:241.5 403.5
leftvalues Cartesian:57.5 299.5
bottomvalues Cartesian:199.5 259.5

Angle in degrees:15.2551

Missing Puzzle top value locations from where one to one mapping is needed to be
 carried out
Puzzle top location x:173
Puzzle top location y:135
```

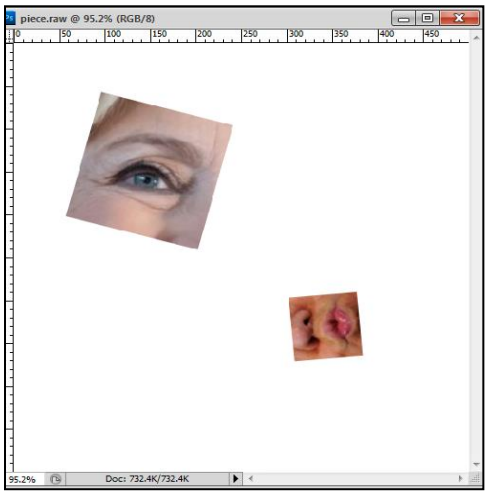


Fig 1.9: Input piece image.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

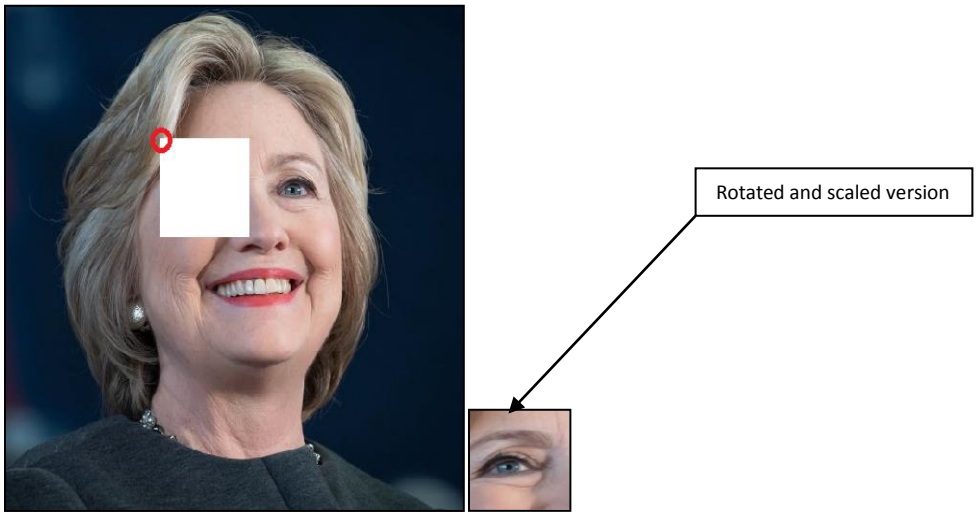


Fig 1.10: Top left position of the puzzle image in red with coordinates x=173 and y=135 mapped to the rotated and scaled version of the piece image with coordinates mentioned above.

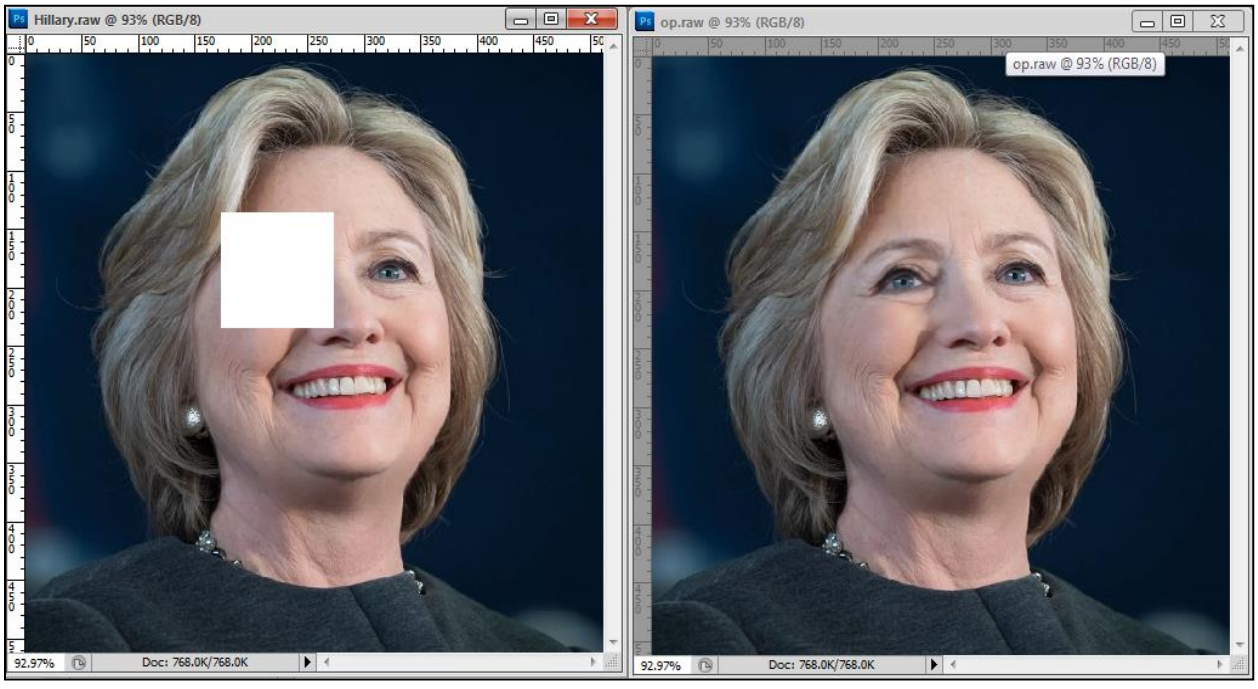


Fig 1.11: The puzzle matched Hillary image.

Case 2: Trump Image:

Aakash Shanbhag

adshanbh@usc.edu

3205699915

The angle of rotation in case of the Trump image is different from that of the Hillary image as well as the coordinates obtained are different in this case. In this case the scale up is required to be carried out in order to match the original size as the original piece image is of a lesser dimension.

```
Locations of Trump piece image are as follows:
topvalues:377 290
rightvalues:383 364
leftvalues:302 296
bottomvalues:308 371

topvalues cart:308.5 128.5
rightvalues cart:302.5 203.5
leftvalues cart:383.5 135.5
bottomvalues cart:377.5 209.5

height:75.326   width:74.2428
Angle in degrees:-95.3322


Missing Puzzle top value locations from where one to one mapping is needed to be
 carried out
Puzzle top location x:163
Puzzle top location y:236
```

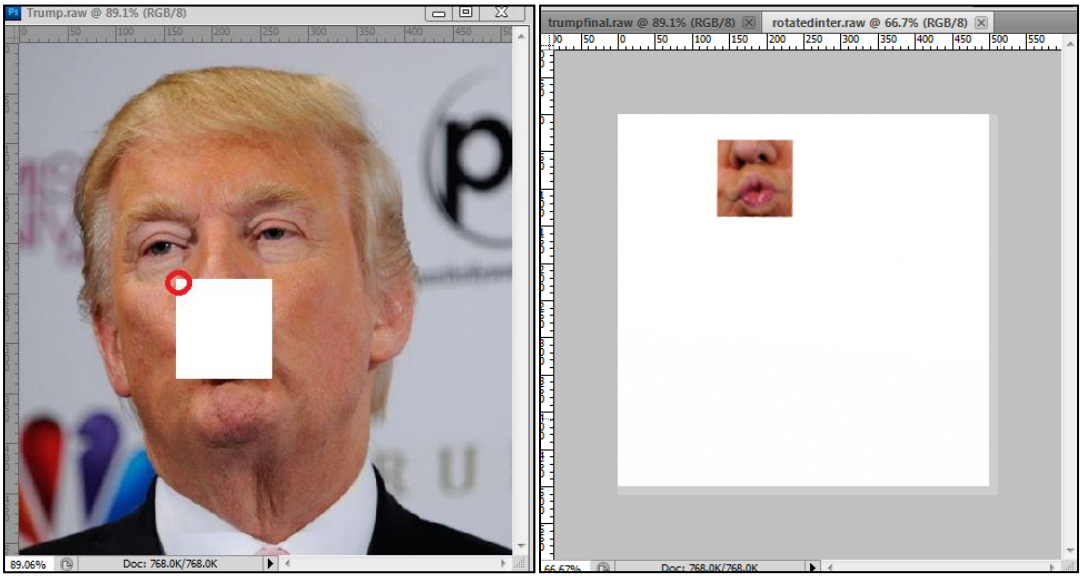


Fig 1.12: Top left position of the puzzle image in red with coordinates x=163 and y=236 mapped to the rotated and scaled version of the piece image with coordinates mentioned above.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

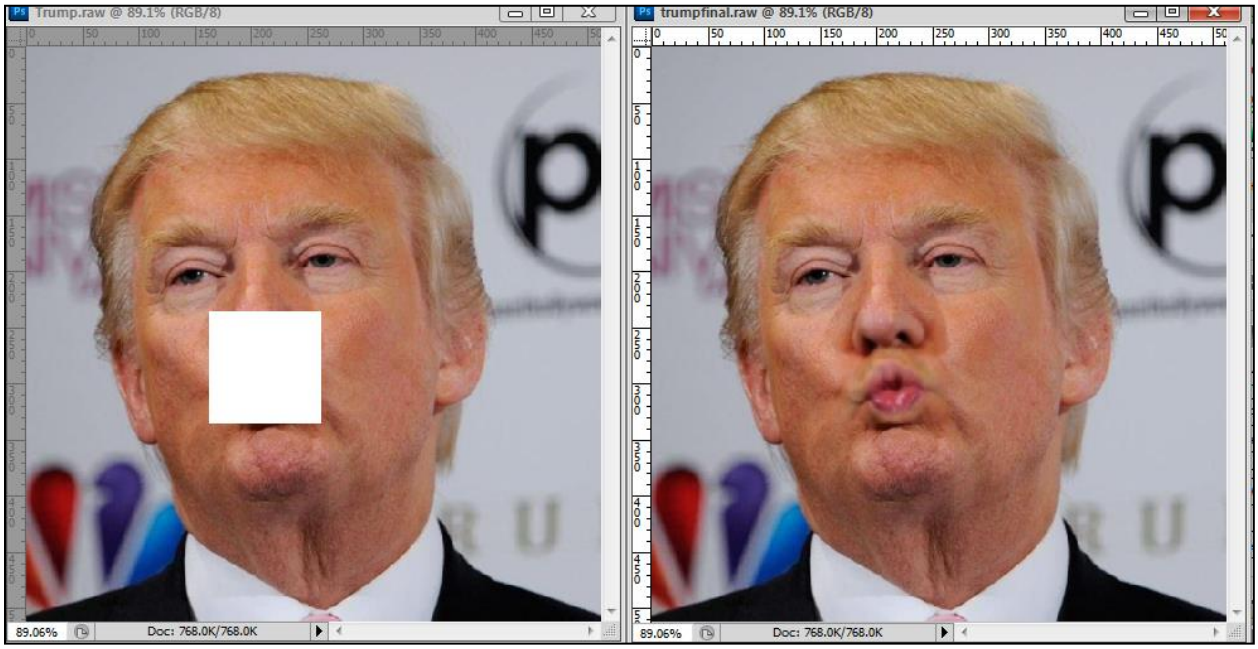


Fig 1.13: The puzzle matched Trump image.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

1.3 (c) Homographic transformation and Image Overlay.

The method of using multiple camera viewpoints is used in homographic transformation and as mentioned in the steps of 1.2 (c), the image coordinates are first converted to Cartesian and then all the reverse mapping occurs. The output is displayed below:

```
image to cartesian of field
 x1:356.5   y1:63.5
 x2:541.5   y2:207.5
 x3:522.5   y3:52.5
 x4:630.5   y4:211.5
image to cartesian of TROJAN
 X1:1.5   Y1:195.5
 X2:349.5   Y2:195.5
 X3:1.5   Y3:0.5
 X4:349.5   Y4:0.5
```

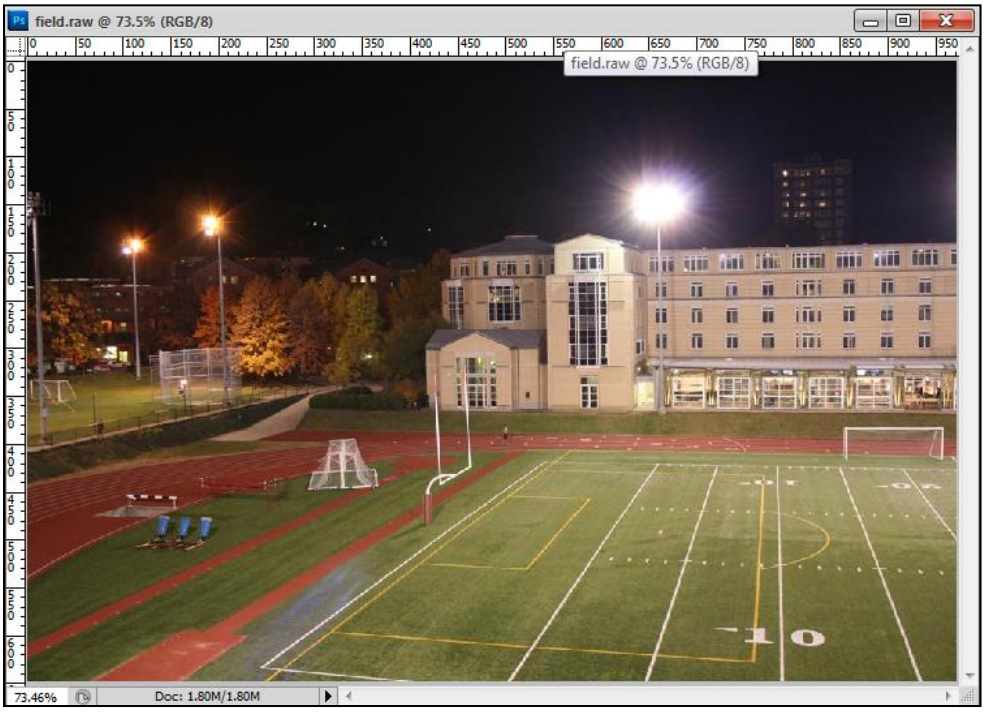


Fig 1.14: The field image.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

Fig 1.15: The Trojan image.

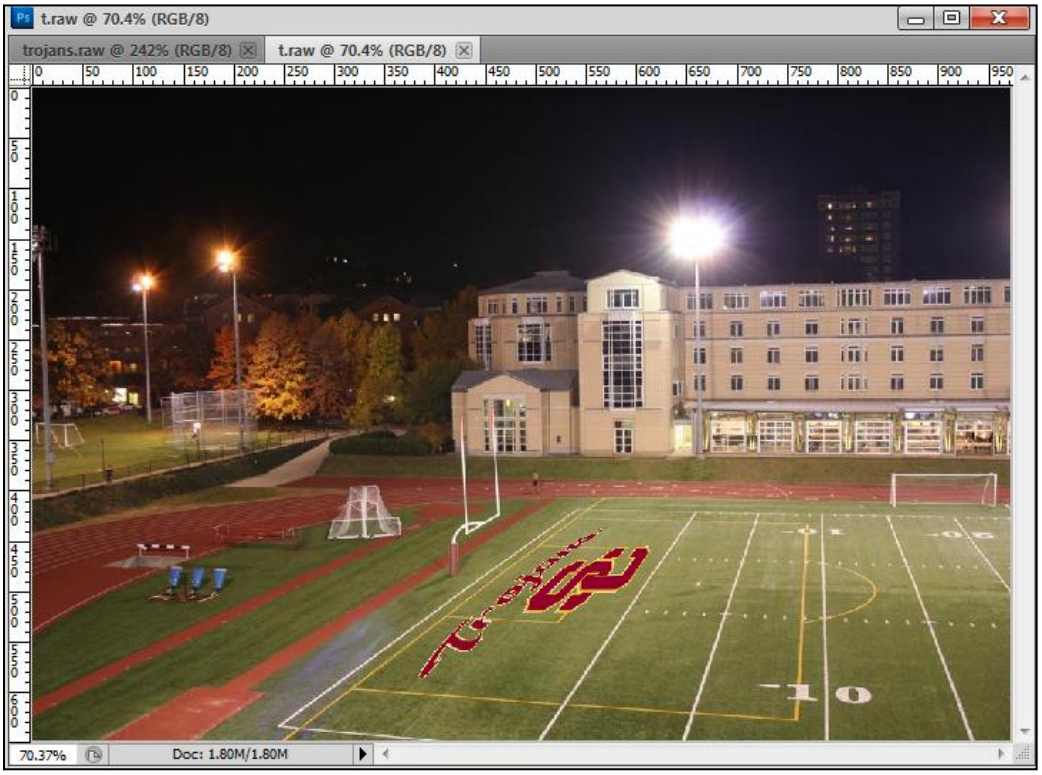


Fig 1.16: The Trojan image overlaid on the field image as desired.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

1.4 Discussion and Conclusion.

According to the requirements all the geometric modifications were made and successful results were obtained in each case. The detailed procedure and method is discussed above and some important conclusions for each of the portions are as follows:

1 (a) Geometric Image Modification by spatial warping.

In this method the desired output was similar to the dog warped image and hence the concept of linear interpolation was used to successfully down sample the image in such a manner that the scaled version did not posses any discontinuity. Fig 1.5 and 1.6 show the step by step change in dealing with the different quarter portions of the image. Figures 1.7 and 1.8 show the outputs of both the cup images and their respective outputs. It is to be noted that in this case forward mapping was used along with linear interpolation. Method of creating triangles and mapping them to certain positions can also done for this example but is too computational complex and takes up larger iterations than desired.

1 (b) Puzzle matching.

In the Puzzle matching problem, reverse mapping along with bilinear interpolation was carried out. Translation, rotation and scaling were carried out by first converting to Cartesian coordinates. In both the cases the pivot point for rotation was taken to be the lower most left point which was translated to the Cartesian origin. In case of Hillary piece matching, the translated and rotated image was larger than the missing puzzle which was scaled down to the desired size by down sampling and applying a reverse mapping along with bilinear interpolation but in case Trump image scaling up by interpolation was needed and hence caused a Blurring effect which is a peculiar feature of up sampling in which a blurring effect is produced. Similar to Hillary image reverse mapping and interpolation produced the desired output.

1 (c) Homographic Image Modification and Image Overlay

The most commonly used model in Aerial Photography, Augmented Reality applications is that of the homographic image transformation. In this process manually the 4 inputs and output points were chosen and overlay corresponding to the perspective transform matrix was utilized. The performance of this algorithm depends on the locations and the up sampling and down sampling process, since large change in the dimensions of the image to be overlaid creates a stretching and blurring effect which is undesired. Hence 4 points in one space can be easily remapped to 4 points on a different plane or space using this homogenous coordinate mapping.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

Problem 2: Digital Halftoning.

2.1 Motivation and Abstract.

Digital halftoning is the process in which the gray level images in the range of 0 to 255 are converted to a range of 0 to 1(binary) image range. This technique is progressively used in printing and scanning along with varied applications of different dots per inches also known as dpi. Different dpi intensities can be used to represent a gray level image into a binary image. For instance the darker portion of the image should incorporate more density of the binary pixels whereas the brighter ones should have lower value of dpi.

Since the equivalent information of a gray level image is being transformed to a binary image, there is visible discontinuity in the image which needs to be minimized drastically and hence two techniques namely Dithering and error diffusion are carried out for improving the quality of the binary images by correct manipulation of the dpi.

2.2 Approach and Procedure.

There are two methods which need to be carried out in this process of digital halftoning. The first section describes the Dithering matrix method which includes the Bayer index and the thresholding matrix and the second approach in which the error is diffused to the neighboring pixels and eradicated in total.

2.2 (a) Dithering Matrix

The following steps need to be carried out for using the Bayer Index matrix to process the gray level image to the dithered equivalent of the image. Instead of adding noise we use an adaptive threshold mark by which dithering can be successfully carried out.

1. Dithering parameters are specified according to the Bayer index matrix. This matrix is as follows:
$$I_2(i,j) = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix}$$
Where 0 indicates the pixel most likely to be turned on and the 3 is the least likely to be turned on.
2. Create a new matrix on the basis of the basic matrix which can be done recursively by the above formula:
$$I_{2n} = \begin{bmatrix} 4 * I_n(x,y) & 4 * I_2(i,j) + 2 \\ 4 * I_2(i,j) + 3 & 4 * I_2(i,j) + 1 \end{bmatrix}$$

For instance the $I_4(i,j)$matrix is as follows:

Aakash Shanbhag

adshanbh@usc.edu

3205699915

$$I_4(i,j) = \begin{bmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{bmatrix}$$

3. Calculate the threshold matrix from the Bayer index matrix as follows where the value of N=4 in the above example denotes the size of the Bayer index matrix:

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

4. Assume the input image is F, and the output image that is being used is G. Once this threshold matrix mask is scanned through the entire image we decide our binary logic as follows:

$$G(i,j) = \begin{cases} 255 & if\ F(i,j) > T(i\%N, j\%N) \\ 0 & otherwise \end{cases}$$

5. Hence the output image is set to either black or white depending on the adaptive threshold method described above.

2.2 (b) Error diffusion method.

The following steps need to be carried out for using Error diffusion technique to process the gray level image to the half toned image. The process is explained below:

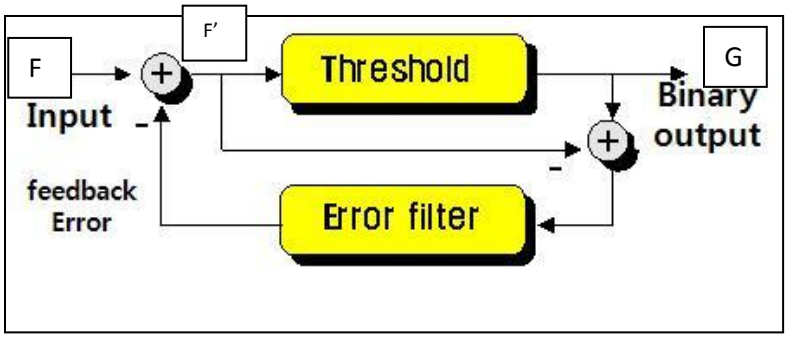


Fig 2.1: Error diffusion process.

Source: cilab.knu.ac.kr

F is the input image and G is the output image with an intermediate F' created which causes the error to be propagated to the neighboring pixels. This F' includes the noise times the weights provided by the different masks to the original pixel.

1. Quantization threshold is set to 0.5 after the image input F pixels are normalized between 0 and 1.
2. If the input pixel value F (i, j)> threshold value, Output pixel G (i, j) is set to 1 or else it is set to 0.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

3. Error is calculated which is the difference between the current pixel value and the output set quantization value.
4. Error hence calculated is diffused to the neighbors with a weight associated according to the matrices defined as follows for Floyd Steinberg:

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

Similarly it can be carried out for JJN, Stucki as well with different weights associated.

$$\frac{1}{48}\begin{bmatrix}0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 7 & 5\\ 3 & 5 & 7 & 5 & 3\\ 1 & 3 & 5 & 3 & 1\end{bmatrix} \qquad \frac{1}{42}\begin{bmatrix}0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 8 & 4\\ 2 & 4 & 8 & 4 & 2\\ 1 & 2 & 4 & 2 & 1\end{bmatrix}$$

5. While calculation of the error for the even row, left to right scanning is done with the above weight times the error being added to the neighboring pixel. While scanning for the odd row, right to left movement is done with the mask weights also being flipped as show below, this is also called as the Serpentine method to avoid accumulation of error in a single direction.

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

6. Accordingly the serpentine technique of scanning can help in changing the neighborhood pixels and hence create a digital half-toned image with compensation of the error among the neighborhood.

To implement 4 grey level halftoning, the threshold matrix can be according to the below method.

1. Define the levels of the threshold in 4 levels viz. 1,$(0.5+0.5 * I_2(i,j)), I_2(i,j), 0.5 * I_2(i,j), 0$. Hence now the new output is given by:

$$G(i,j) = \begin{cases} 255, & if\ 1 > F(i,j) > (0.5 + 0.5\ * I_2(i\%N, j\%N)) \\ 175, & if\ \left(0.5 + 0.5\ * I_2(i\%N, j\%N)\right) > F(i,j) > I_2(i\%N, j\%N) \\ 85, & if\ I_2(i\%N, j\%N) > F(i,j) > 0.5 * I_2(i\%N, j\%N) \\ 0, & if\ 0.5 * I_2(i\%N, j\%N) > F(i,j) > 0 \end{cases}$$

Aakash Shanbhag

adshanbh@usc.edu

3205699915

Where $F(i,j)$ is the normalized value between 0 and 1 and 2is the size of the Bayer Matrix applied. Any size of the Bayer index matrix can be applied.

2. The above method can be carried out with keeping the levels to 1, $0.75 * I_4(i,j)), I_4(i,j), 0.5 * 0.25 * I_4(i,j), 0$ and then applying the step mentioned above with different intermediate values.
3. The output is slightly varied if the levels are changed by an amount described above and arbitrary level setting can lead to different results.

4. Choice of the threshold level can affect the results tremendously.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

2.3 Experimental Results.

2.3 (a) Dithering Matrix

The matrix for $I_2(i,j)$ and $I_8(i,j)$ are as follows:

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

Dividing by 4 would obtain the normalized value.

0	32	8	40	2	34	10	42
48	16	56	24	50	18	58	26
12	44	4	36	14	46	6	38
60	28	52	20	62	30	54	22
3	35	11	43	1	33	9	41
51	19	59	27	49	17	57	25
15	47	7	39	13	45	5	37
63	31	55	23	61	29	53	21

Fig 2.2: $I_8(i,j)\ matrix$. Dividing the above by 64 would obtain the normalized value.

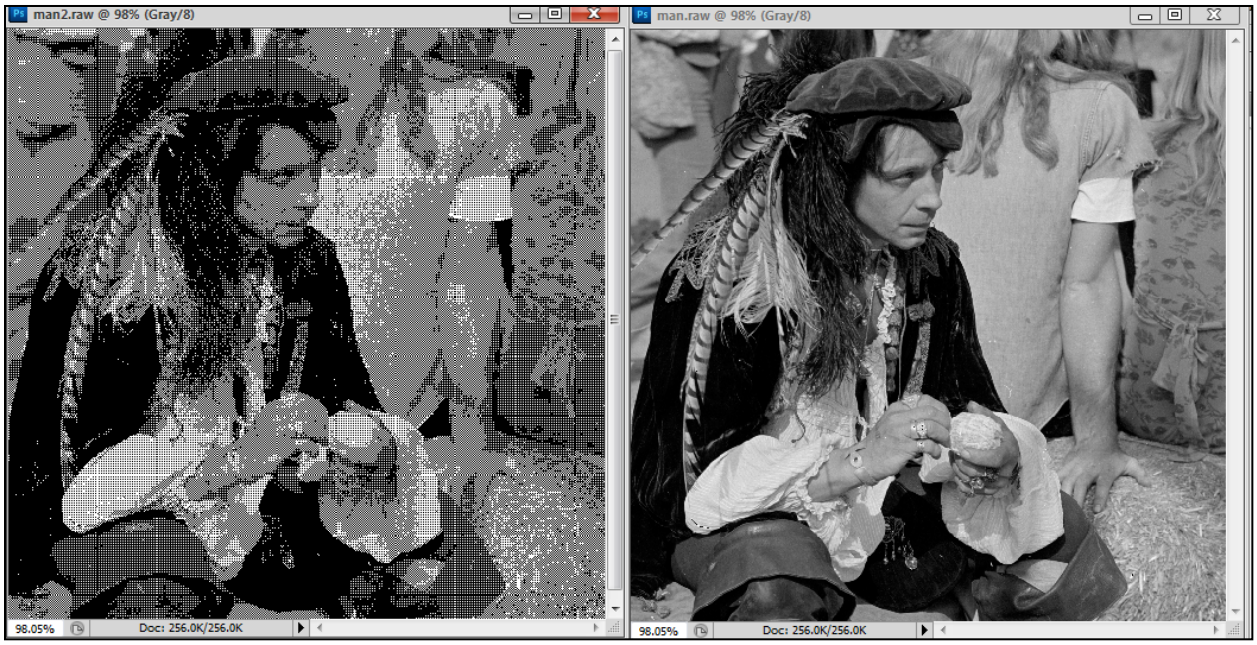


Fig 2.3: $I_2(i,j)$ applied on the man.raw image (right) and output (Left) with similar patterns obtained.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

Fig 2.4: $I_8(i,j)$ applied on the man.raw image (right) and output (Left) with similar patterns obtained

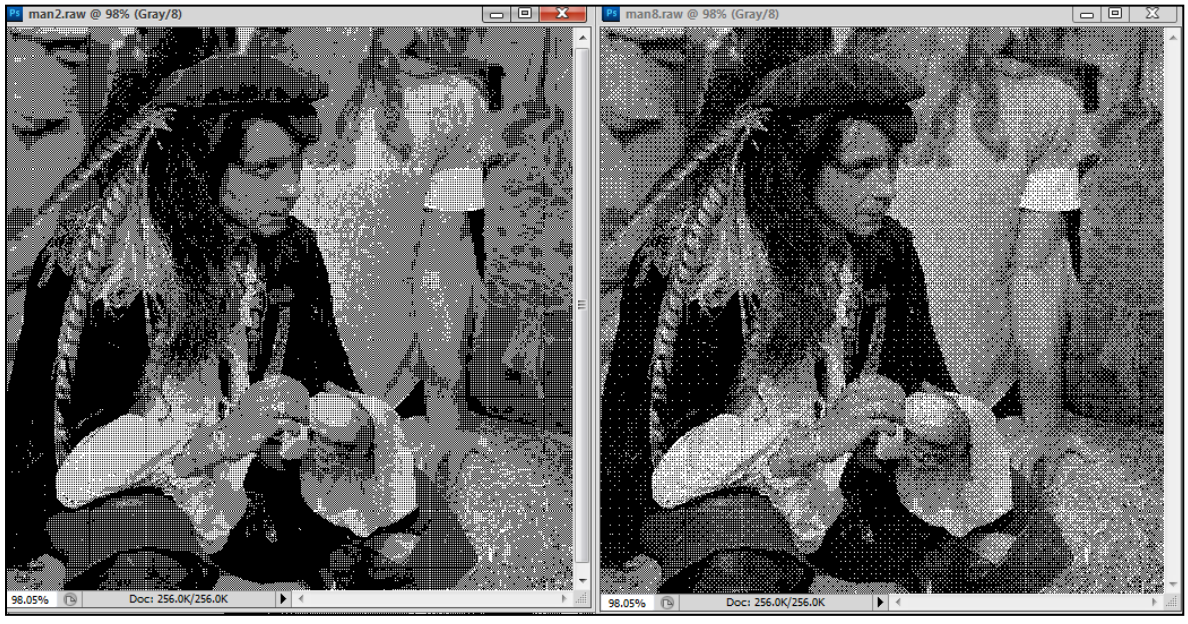


Fig 2.5: $I_8(i,j)$ applied on the man.raw image (right) and $I_2(i,j)$ (Left) with similar patterns obtained. Clearly better oriented image can be seen on the right with more variation than the previous one.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

The $I_4(i,j)$ matrix is as follows which can be normalized by diving by 16:

0	8	2	10
12	4	14	6
3	11	1	9
15	7	13	5

Fig 2.6: The $I_4(i,j)$ matrix.

The $A_4(i,j)$ matrix is as follows:

$$A_4(i,j) = \begin{bmatrix} 14 & 10 & 11 & 15 \\ 9 & 3 & 0 & 4 \\ 8 & 2 & 1 & 5 \\ 13 & 7 & 6 & 12 \end{bmatrix}$$

Fig 2.7: $I_4(i,j)$ applied on the man.raw image (left) and $A_4(i,j)$ (right) with similar patterns obtained.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

Fig 2.8: 4 level equivalent with size 2 on the left (input) and on the right (output).

2.3 (b) Error diffusion method.

The error diffusion techniques using Floyd Steinberg, JJN and Stucki are shown below:

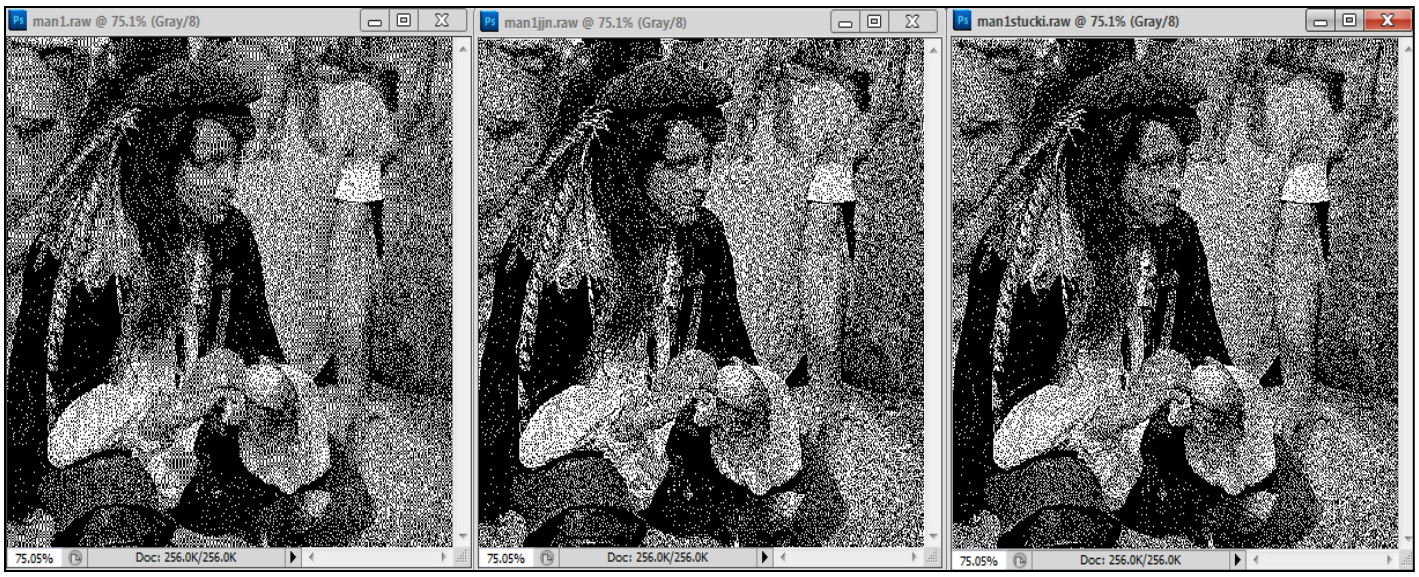


Fig 2.9: Comparison between Floyd Steinberg, JJN and Stucki.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

Fig 2.10: Floyd Steinberg error diffusion method.

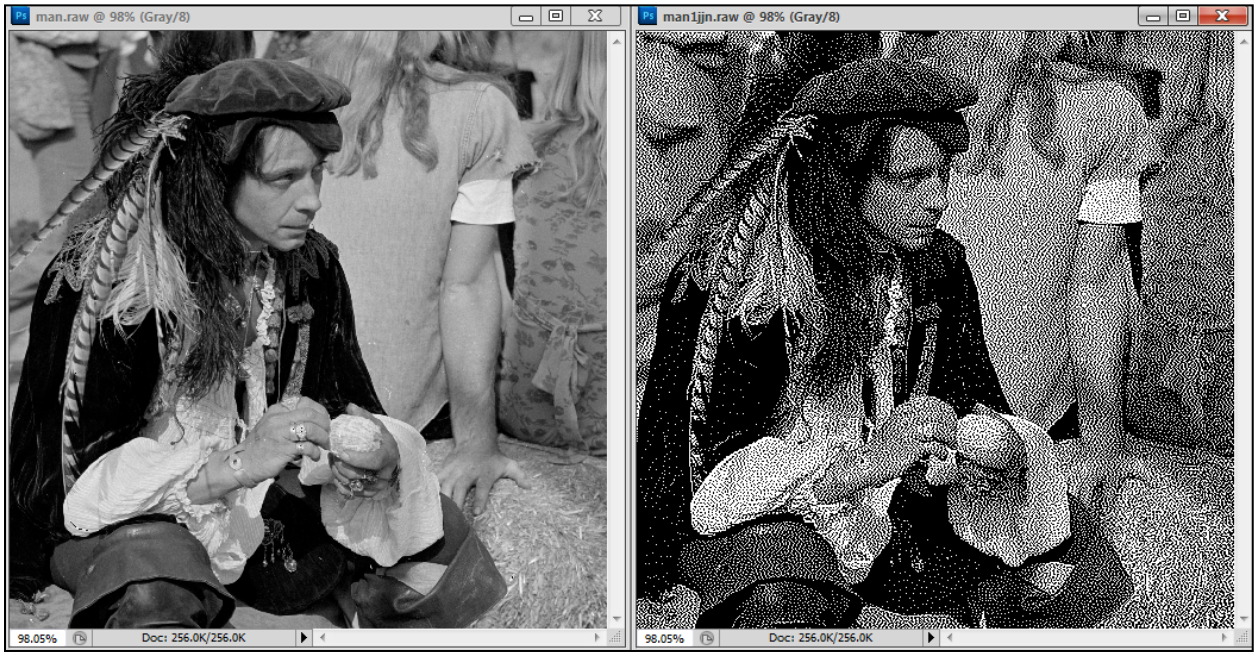


Fig 2.11: JJN error diffusion method.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

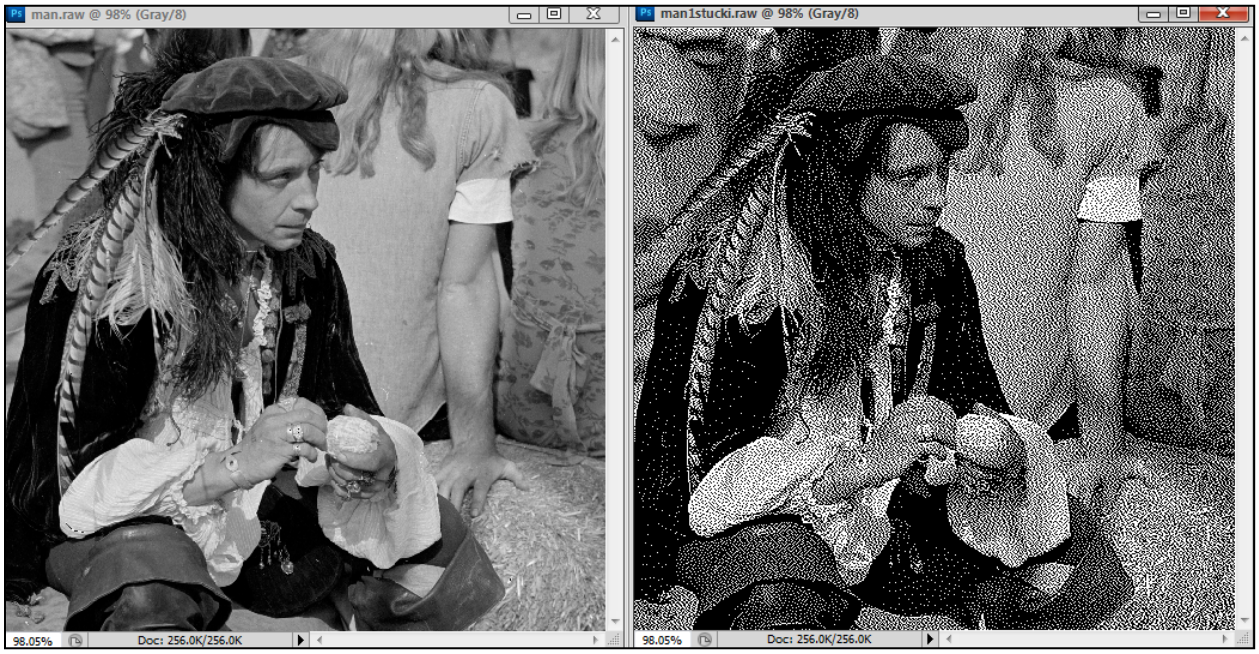


Fig 2.12: Stucki error diffusion method.

2.4 Discussion and Conclusion.

Figures above describe the dithering method and the error diffusion method in brief and state the differences among the methods and also the effect of different Bayer indices and different error diffusion methods.

(a) Application of $I_2(i,j)$ and $I_8(i,j)$ matrices on the man.raw image.

The $I_2(i,j)$ and $I_8(i,j)$ matrices are defined above and it is clearly visible that the variation obtained in a larger Bayer index size is better than the variation in the smaller Bayer index. Repetitive patterns or artifacts are obtained by the recursion principle that is followed and hence we can generate the above figures. These effects are better than the absolute hard thresholding and generate different dpi based on the index and the Size and is equivalent to adding noise to the image. Fig 2.5 clearly delineates this fact and shows the variation precisely. The variation on the hand and the details are better seen on the I8 matrix as compared to the I2 matrix which is partly blurred.

(b) Compare I4 and A4 matrices and their differences.

As it is clearly visible from the I4 matrix which is recursively obtained from the I2 matrix and has 0 as the largest weight as compared to 3 which has the lowest weight. The adaptive threshold matrix pattern of the I4 can be seen clearly in Fig 2.7. The output of I4 is considered to contain specs and output black

Aakash Shanbhag

adshanbh@usc.edu

3205699915

and white dots. The A4 is somewhat different from the I4 and can hence be seen in the SPIRAL pattern arrangement of the matrix. This spiral pattern causes the image dithered to produce a grid like structure passing through the image which is not seen in the I4. The smoothness in the I4 filter is clearly evident and is better because of the distribution around the neighbors and it can be witnessed in the fig 2.7. A4 due to the spiral behavior creates the grid effect which makes it difficult to distinguish sharp edges and convoluted patterns.

(c) 4 gray level display image.

The algorithm for the development of a 4 gray level Image is generated and explained in the procedure section briefly. Here the threshold matrix that is used is updated and hence for different set of threshold values, different threshold matrices can be generated. The choice of the threshold matrix is arbitrary and hence such different patterns and output can be obtained. Larger variation can be obtained if the size of the Bayer matrix is larger. There are more gray levels to represent a gray image and hence better representation of the image can be obtained in this procedure. The process includes the setting up of a threshold boundary and breaking the 0 to 255 range into 4 levels viz. 0, 85, 170, 255 instead of the normal 0,255 digital halftoning scheme for printing. Fig 2.8 shows the closeness of the 4 level image to the gray image generated which is much better than digital half toned image with 2 levels.

(d) Comparison between Error diffusion methods.

The error diffusion techniques are compared in the fig 2.9 and the following images in fig 2.10 to 2.12 provide evidence regarding about the error diffused and no propagation in only one direction. Since serpentine type of scanning technique is applied the error is diffused and no accumulation occurs in any of the cases; the filters are also flipped while moving from left to right and right to left respectively. Since high frequency noise called as Blue noise is added in this technique, all the approaches show large amount of noise being introduced in the system.

Floyd Steinberg is a 3x3 mask whereas JJN and Stucki error diffusion are 5x5 and hence affect a larger neighborhood as compared to Floyd Steinberg. Hence in a general sense the JJN and Stucki tend to provide a visually better output than Floyd Steinberg. It is difficult to make out the difference between the Stucki and the JJN as they tend to provide a similar output, just that the weights associated are larger in case of Stucki than JJN. The JJN method could provide stronger edges as compared to the other methods and hence can be used when edge information is more importance as compared to other portions of the image. Blur is the largest for Floyd Steinberg and is relatively less for the other two techniques.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

(e) Propose a technique that can be utilized.

According to me, the advantages of both the error diffusion and the dithering can be combined in a single hybrid process. This hybrid process would incorporate the adaptive thresholding technique. In such a process instead of calculating the error as the round off to the closest intensity in error diffusion the mask can be iteratively used to decide the threshold of the pixel concerned and set it adaptively instead of rounding the palette to the nearest value. In this method the error generated can be propagated ahead in the same way as error diffusion techniques and restricted from being accumulated by using the serpentine manner of scanning. Hence the advantage of dithering (variation increase) and error diffusion can be incorporated simultaneously.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

Problem 3: Morphological Image Processing

3.1 Motivation and Abstract.

The most commonly used techniques on binary images that give detailed information about the shape cover the class of morphological image processing. These techniques are very essential since they find applications in Optical character recognition, finger print scanning, word recognition and many more. It is crucial that the image given is a binary image else it needs to be converted to a binary image by various methods. Shape processing is then carried out on such binary images. Some fundamental morphological processing includes shrinking, thinning, skeletonizing which cover the class of subtractive masks while some additive masks such as flood-fill, dilation are also implemented in this problem.

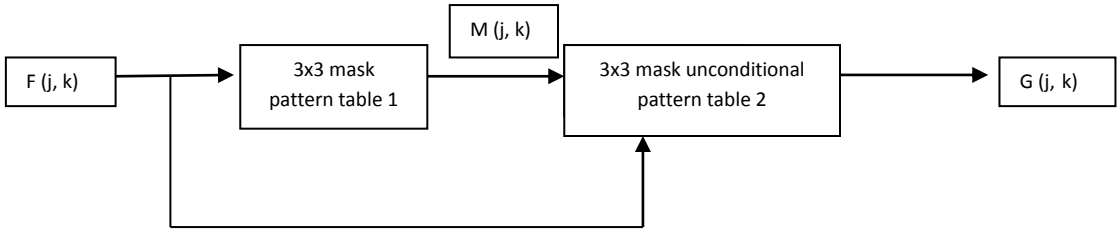
3.2 Approach and Procedure.

In this section of the problem, initially the subtractive class of morphological image processing is carried out. Shrinking causes the binary image to shrink to a small single pixel which van be sometimes useful for counting the number of objects in the foreground as well as knowing the sizes of figures in the foreground. Thinning can be used to remove the bold structure of the image in a subtractive manner to produce single lines and Skeletonising can be used to decipher the exact edges in the image. Each of these cases has specific applications. Hybrid mixture of additive and subtractive masks is seen in the last part of the problem which enhances the capability of counting.

3.2(a) Shrinking

The process that I followed to obtain to shrinking is as follows:

1. Generate the first and second stage pattern tables that are necessary to check whether a hit or miss occurs in the entire process.
2. The flow diagram of the process of Shrinking is as follows:



3. A 5x5 mask is broken down into 2 stages of 3x3 and then filtering is carried out according to the masks and the centre pixel under consideration.
4. We need to create an M array with the same dimension as the binary image input F (j, k).
5. According to the mask defined below decisions need to be taken:

Aakash Shanbhag

adshanbh@usc.edu

3205699915

X_0	X_1	X_2
X_3	X(j,k)	X_5
X_6	X_7	X_8

3x3 masks generally considered

6. If the centre value X(j,k)==0 on scanning the entire input image with the 3x3 conditional mask, set M (j,k)=0.
7. Generate a string pattern ($X_0X_1X_2X_3X(j,k)X_5X_6X_7X_8$) and compare it with the first pattern table generated for conditional masks of Shrinking. If any of the patterns match set M (j, k) =1 else set it to 0.
8. Scan the entire M matrix after the completion of the first phase and generate the string table ($M_0M_1M_2M_3M(j,k)M_5M_6M_7M_8$).
9. If M (j, k) ==0, Set the output to the input, i.e. G (j, k) = F (j, k).
10. If M (j, k) ==1, then compare this with the second pattern table of unconditional masks, if it matches to any of the patterns then set G (j, k) =0 else keep the original value as before.
11. Copy the output image to input for the succeeding iteration and loop back till one pixel size foreground objects are not obtained.
12. **Counting process** is carried out at each iteration to check 1 pixel values as foreground.
13. Counting process is the process of checking if any of the pixels in the image is equal to 255 with all its neighbors equal to zero. Effectively by comparing with the mask:

0	0	0
0	1	0
0	0	0

14. Multiple such iterations are carried out and hence count is updated at each iteration. The process of counting terminates when the count does not change over a large number of iterations.
15. Calculation of **Size and frequency**: The size and frequency of the occurrences of the squares in the image can also be directly calculated using the shrinking process. The process includes the following steps :

Aakash Shanbhag

adshanbh@usc.edu

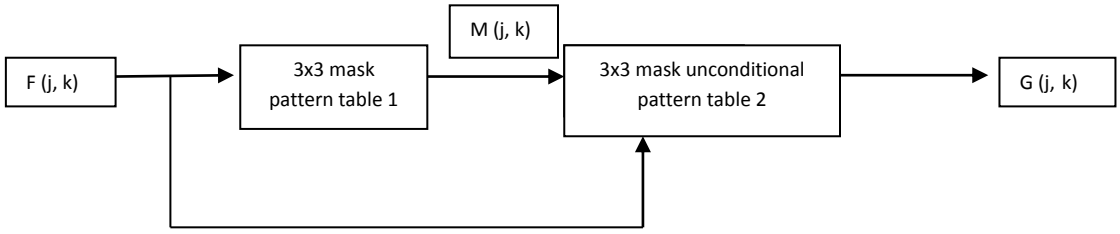
3205699915

(A).At each iteration, the number of one pixel white blocks are calculated and given to a count variable which gets updated every iteration.
(B). The frequency is hence obtained by subtracting the count obtained at each iteration- from that of the previous count with initial count set to 0.
(C).The size obtained is equal to (2*iterations)* (2*iterations) provided that there is a difference in the count from the previous count else that iteration does not count towards the size or frequency as total shrinking has not occurred at that stage.
(D).This process continues till the count does not become stagnant and ceases to change beyond a point.

3.2(b) Thinning

The process that I followed to obtain to shrinking is as follows:

1. Generate the first and second stage pattern tables that are necessary to check whether a hit or miss occurs in the entire process.
2. The flow diagram of the process of thinning is as follows:



3. A 5x5 mask is broken down into 2 stages of 3x3 and then filtering is carried out according to the masks and the centre pixel under consideration.
4. We need to create an M array with the same dimension as the binary image input F (j, k).
5. According to the mask defined below decisions need to be taken:

X_0	X_1	X_2
X_3	X(j,k)	X_5
X_6	X_7	X_8

3x3 masks generally considered

6. If the centre value X(j,k)==0 on scanning the entire input image with the 3x3 conditional mask, set M (j,k)=0.

Aakash Shanbhag

adshanbh@usc.edu

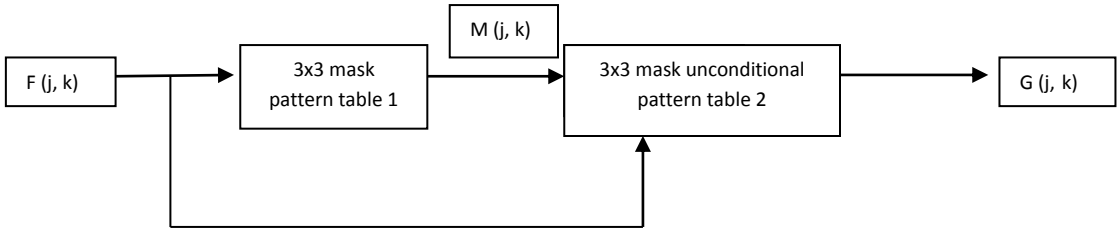
3205699915

7. Generate a string pattern ($X_0X_1X_2X_3X(j,k)X_5X_6X_7X_8$) and compare it with the first pattern table generated for conditional masks of Thinning. If any of the patterns match set M (j,k)=1 else set it to 0.
8. Scan the entire M matrix after the completion of the first phase and generate the string table ($M_0M_1M_2M_3M(j,k)M_5M_6M_7M_8$).
9. If M (j, k) ==0, Set the output to the input, i.e. G (j, k) = F (j, k).
10. If M (j, k) ==1, then compare this with the second pattern table of unconditional masks of thinning, if it matches to any of the patterns then set G (j, k) =0 else keep the original value as before.
11. Copy the output image to input for the succeeding iteration and loop back till thin objects are not obtained.

3.2(c) Skeletonizing

The process that I followed to obtain to shrinking is as follows:

1. Generate the first and second stage pattern tables that are necessary to check whether a hit or miss occurs in the entire process.
2. The flow diagram of the process of Skeletonizing is as follows:



3. A 5x5 mask is broken down into 2 stages of 3x3 and then filtering is carried out according to the masks and the centre pixel under consideration.
4. We need to create an M array with the same dimension as the binary image input F (j, k).
5. According to the mask defined below decisions need to be taken:

X_0	X_1	X_2
X_3	X(j,k)	X_5
X_6	X_7	X_8

3x3 masks generally considered

6. If the centre value X(j,k)==0 on scanning the entire input image with the 3x3 conditional mask, set M (j,k)=0.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

7. Generate a string pattern ($X_0X_1X_2X_3X(j,k)X_5X_6X_7X_8$) and compare it with the first pattern table generated for conditional masks of Skeletonizing. If any of the patterns match set M (j,k)=1 else set it to 0.
8. Scan the entire M matrix after the completion of the first phase and generate the string table ($M_0M_1M_2M_3M(j,k)M_5M_6M_7M_8$).
9. If M (j, k) ==0, Set the output to the input, i.e. G (j, k) = F (j, k).
10. If M (j, k) ==1, then compare this with the second pattern table of unconditional masks of Skeletonizing, if it matches to any of the patterns then set G (j, k) =0 else keep the original value as before.
11. Copy the output image to input for the succeeding iteration and loop back till skeletonized objects are not obtained.

3.2(d) Counting Game.

The Steps needed to obtain the desired outcome are as follows:

1. The given board.raw image is first converted to its complementary image by taking the logical not operation which replaces the white in the image to black and vice versa as shown in figure 3.11.
2. After conversion by complementing, we carry out the shrinking operation which reduces the holes inside the image to a single pixel which also signify the center of the holes (Centroid).
3. These one pixel size images are counted by checking if all the neighbors are black which results in the count of the holes which is required.
4. Now since we know the Centroid of the holes we can use these pixel locations for flood filling.
5. Flood filling process is carried out in the following recursive manner:

   Flood-fill (centre pixel location)

   {If centre pixels==0, return,

   Set the centre pixel to 255 (white)

   Flood-fill (top pixel above centre pixel);

   Flood-fill (left pixel to the centre pixel);

   Flood-fill (right pixel to the centre pixel);

   Flood-fill (bottom pixel below the centre pixel);

   Return; }

Aakash Shanbhag

adshanbh@usc.edu

3205699915

6. After the flood filling is carried out correctly, it has to be followed by Shrinking which causes the white objects in the image to shrink to small one pixel sizes.
7. Hence such one pixel values can be counted and the total number of white pixels in the image can be obtained.
8. Hence by doing the above processes we can easily obtain the total number of holes and the total black holes.
9. Once the shrinking is done we know the centroids of all the white object pixels and these locations can now be used to check with the original flood-filled image.
10. While checking the flood-filled image we take into consideration the fact that the distance of the centroid in the in the square is equal to any of its adjacent sides and hence by checking if the distances of the centroid are equal in all the white objects in the flood-filled image, we can ascertain as to whether a square is encountered or not.
11. Hence if the distances are not found equal we consider them to circle objects.
12. Hence in this way as shown through figures 3.11, 3.12 we can calculate the number of squares and the circles in the image.

Aakash Shanbhag

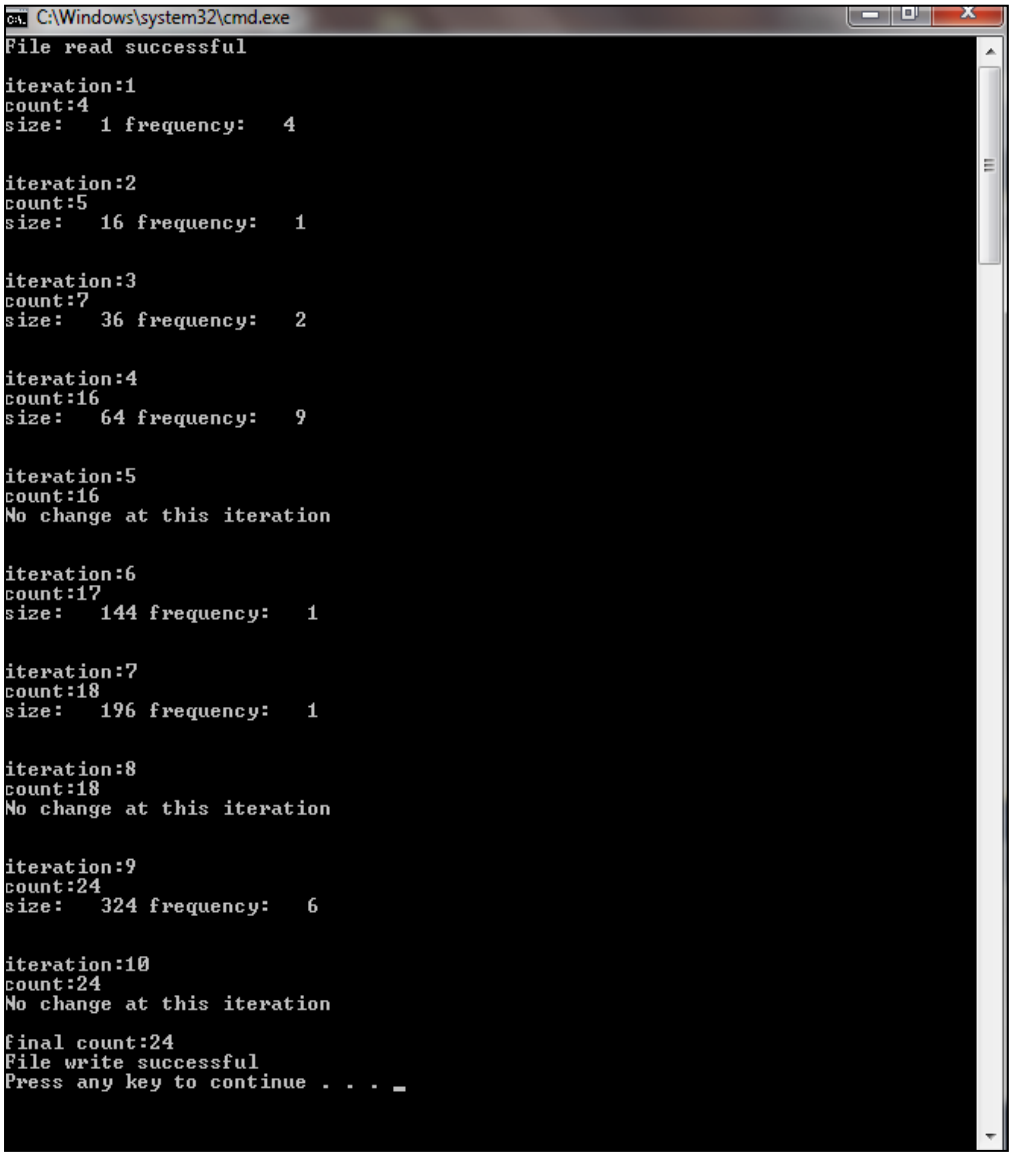
adshanbh@usc.edu

3205699915

3.3 Experimental Results.

3.3 (a) Shrinking.

The outputs generated by carrying out the process of shrinking according to procedure defined in 3.2(a) are as follows:



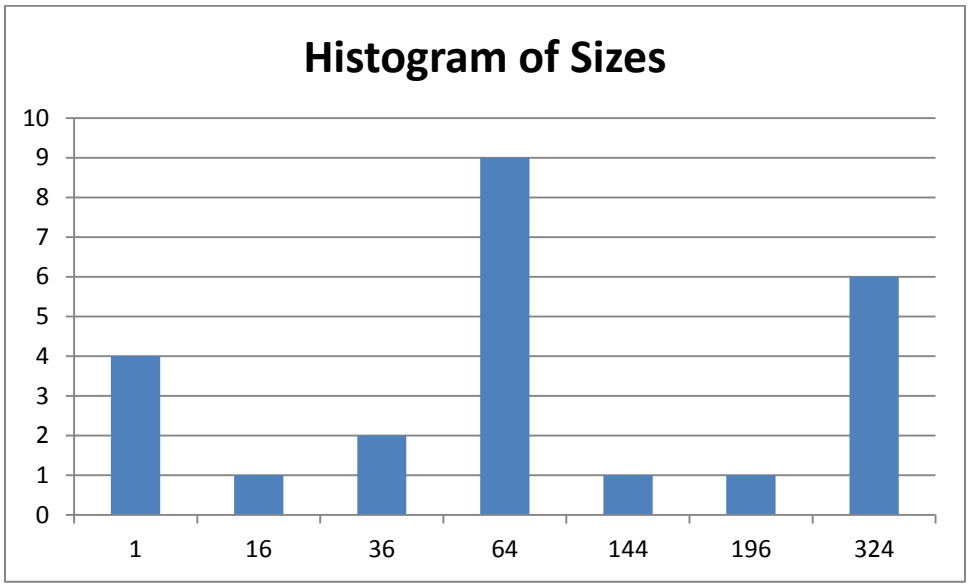
The above results indicate the process of shrinking with 10 iterations and the calculation of size and the frequency of occurrence at each stage. It is necessary to note that since we have square pixels as

Aakash Shanbhag

adshanbh@usc.edu

3205699915

foreground the above results are obtained and the histogram that indicates the size and the corresponding frequency of occurrence is as follows:



The final count obtained in this process is equal to 24 and the output after iterations is as follows:

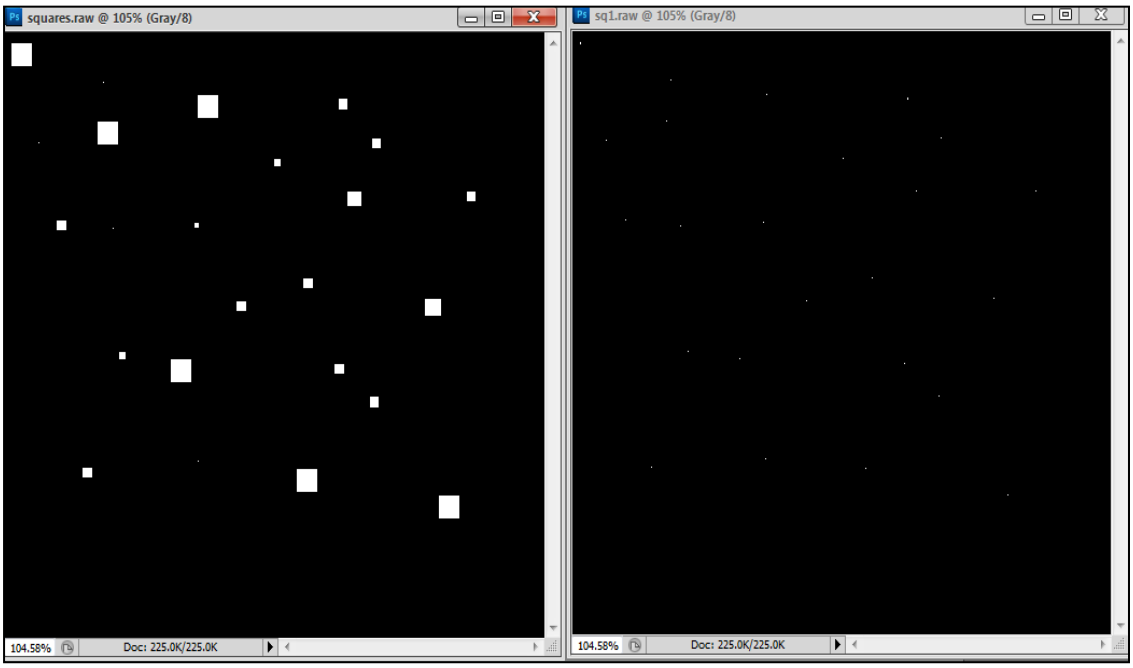


Fig 3.1: Final output of shrinking after 10 iterations.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

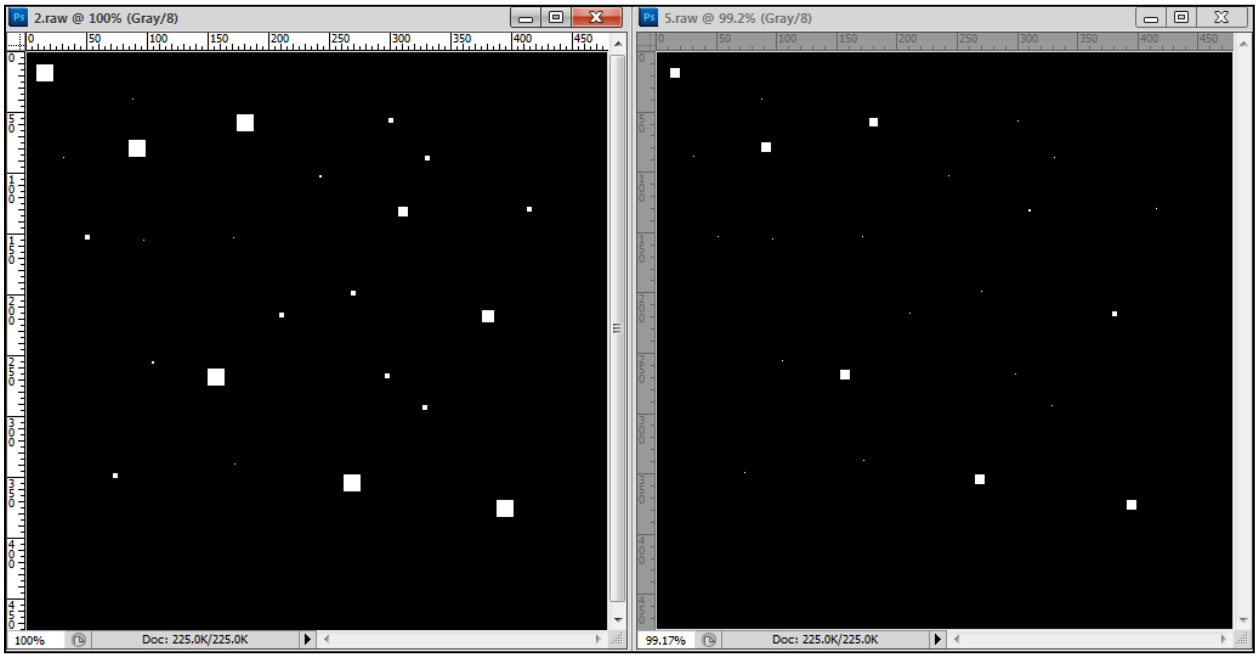


Fig 3.2: Output of shrinking after 2 and 5 iterations respectively.

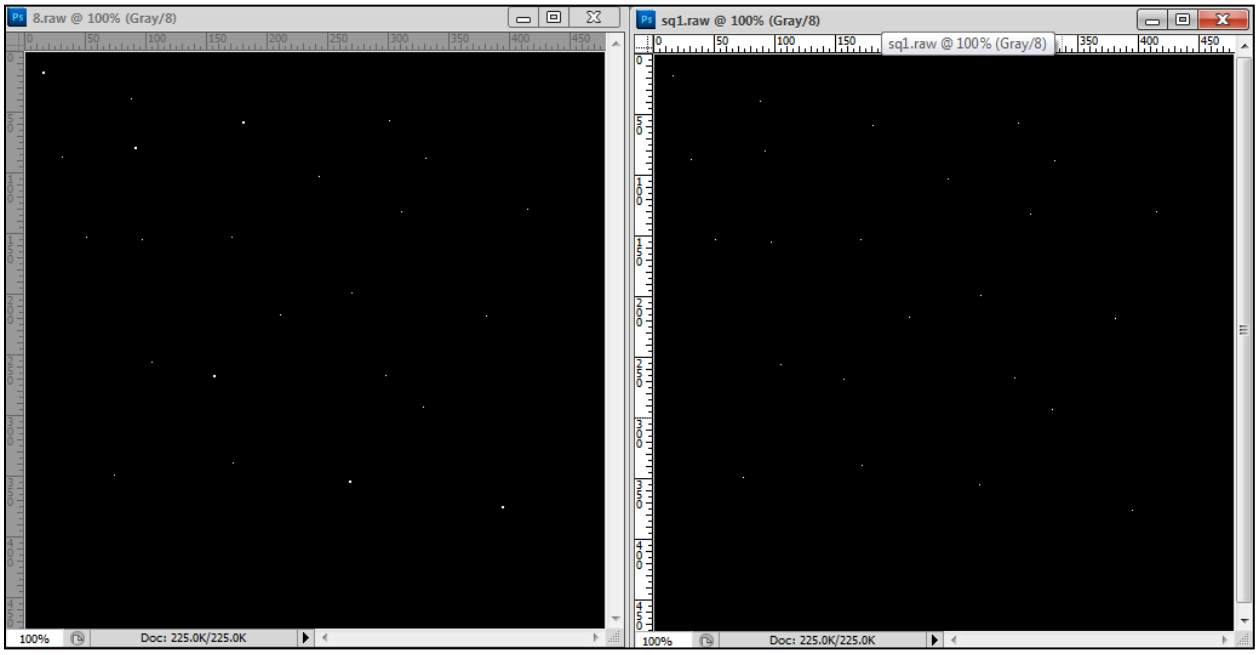


Fig 3.3: Output of shrinking after 8 and 10 iterations respectively.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

3.3 (b) Thinning.

The outputs generated by carrying out the process of shrinking according to procedure defined in 3.2(b) are as follows:

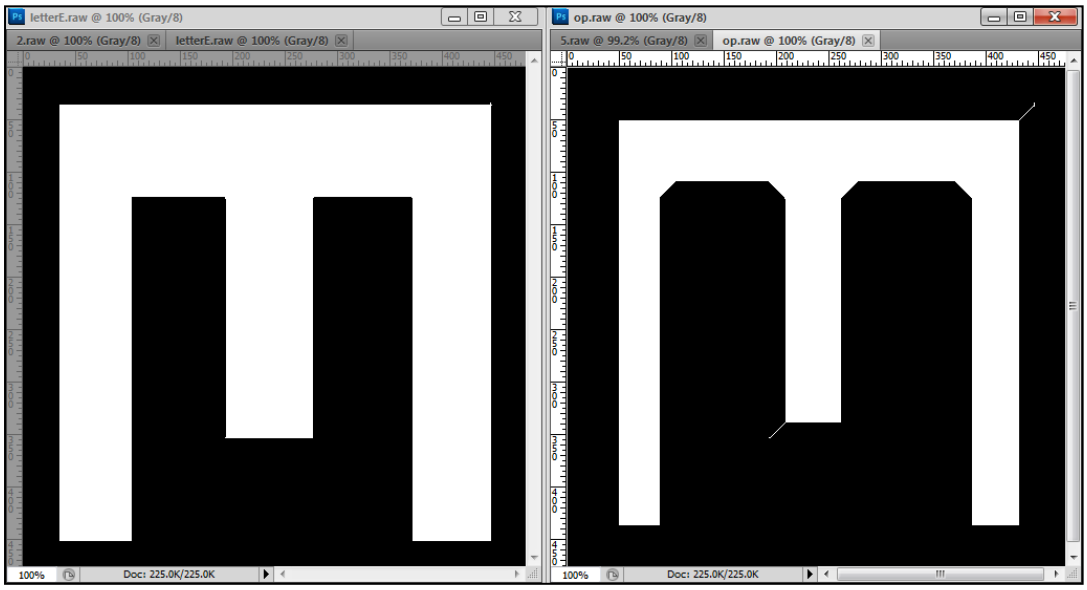


Fig 3.4: Output of Thinning after 20 iterations.

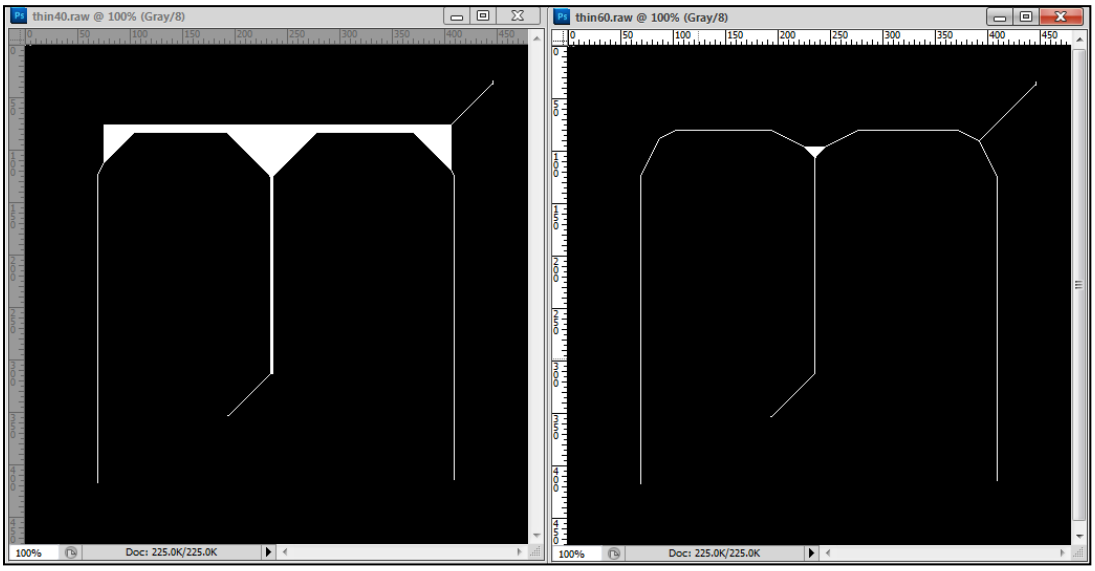


Fig 3.5: Output of thinning after 40 and 60 iterations respectively.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

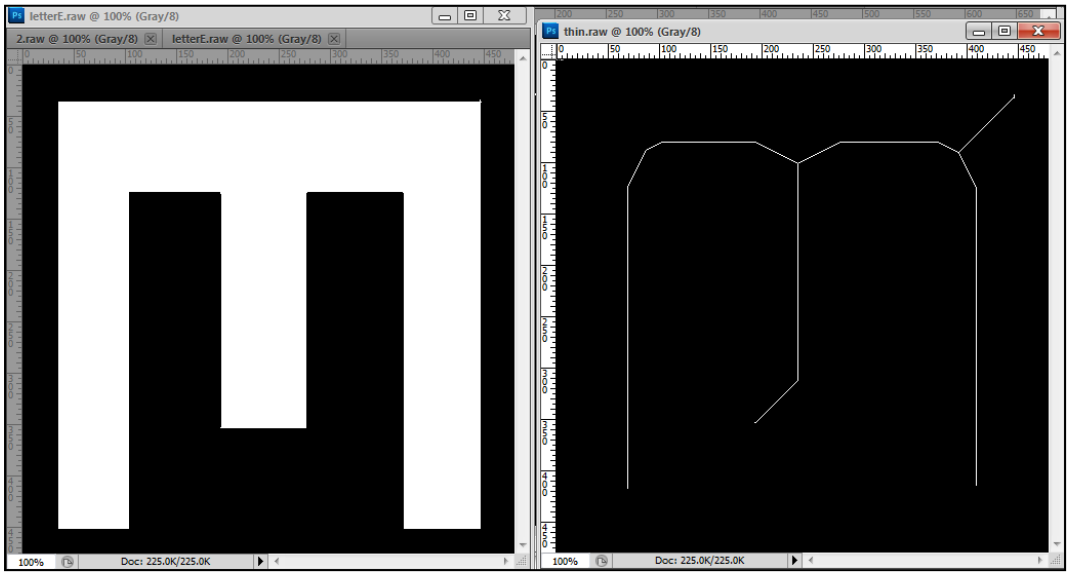


Fig 3.6: Output of thinning after 80 iterations.

3.3 (c) Skeletonizing.

The outputs generated by carrying out the process of shrinking according to procedure defined in 3.2(c) are as follows:

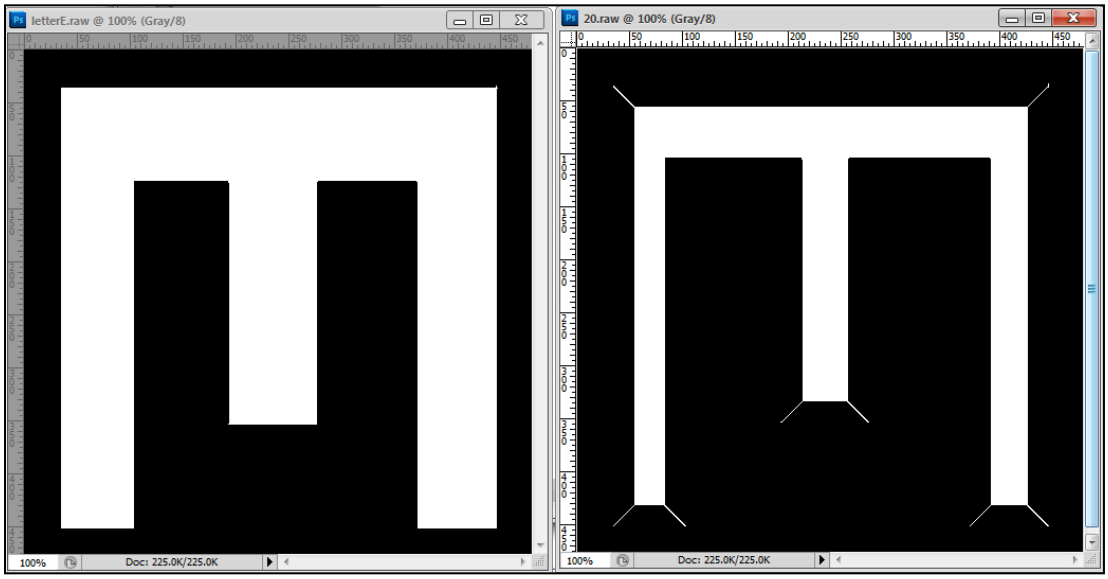


Fig 3.7: Output of skeletonizing after 20 iterations.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

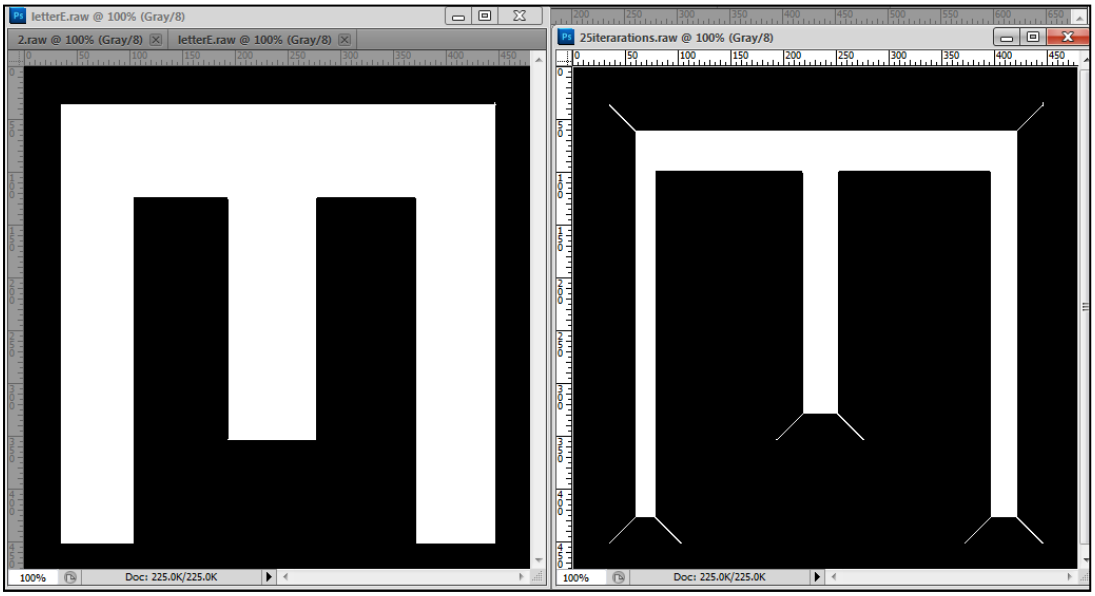


Fig 3.8: Output of skeletonizing after 25 iterations.

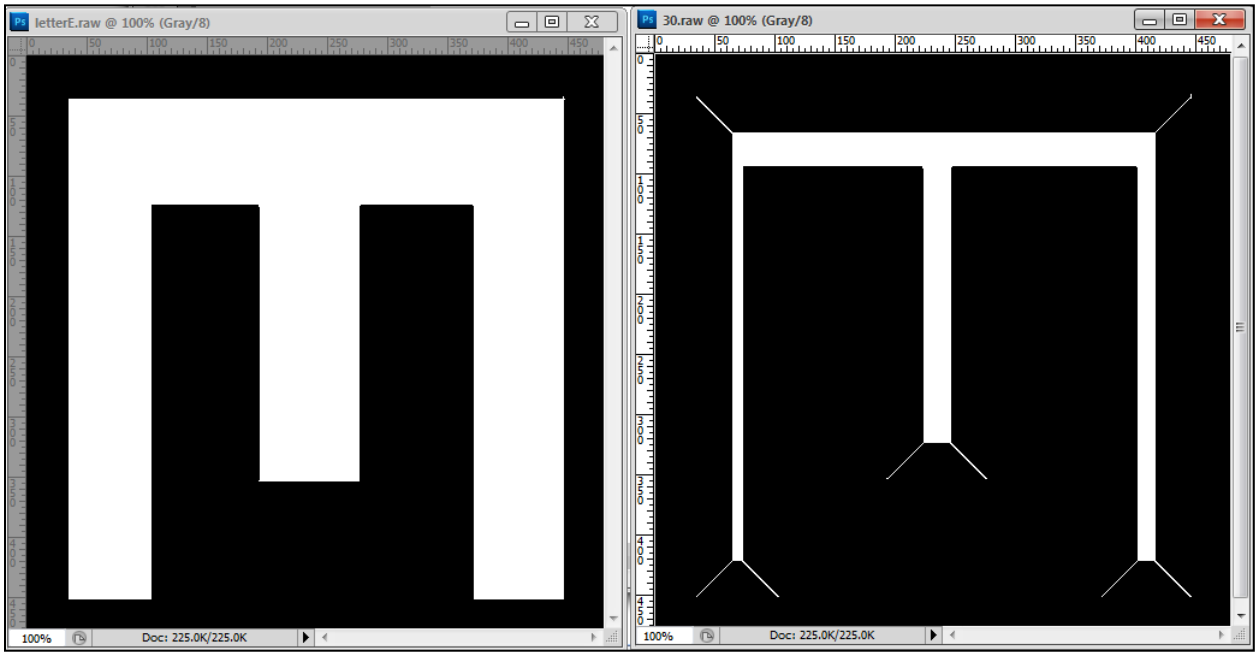


Fig 3.9: Output of skeletonizing after 30 iterations.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

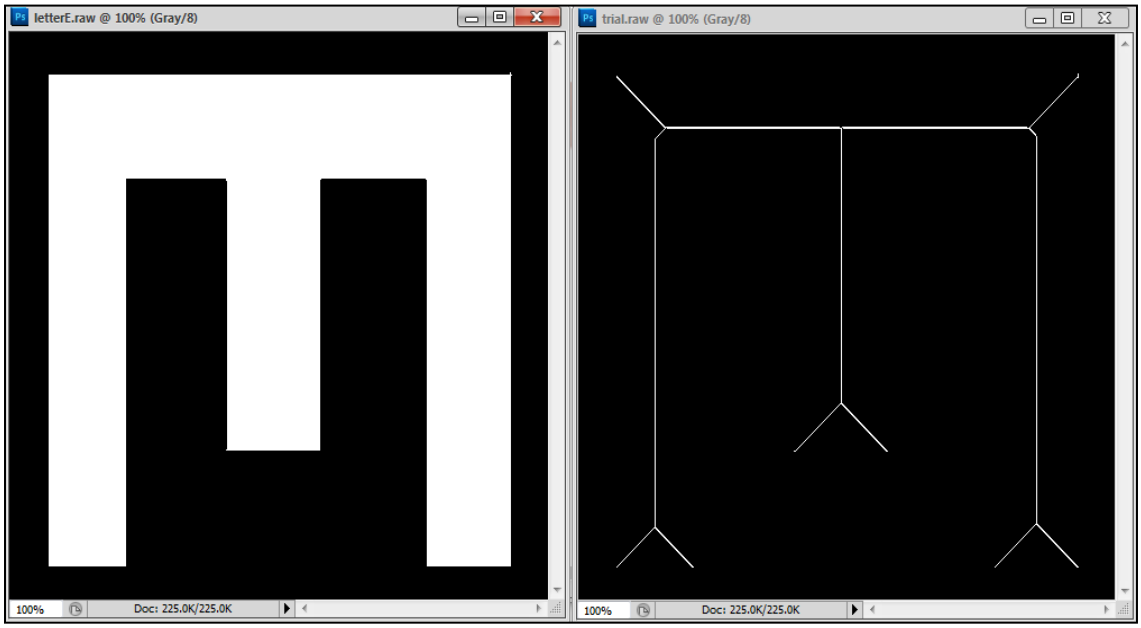


Fig 3.10: Output of thinning after 42 iterations.

3.3 (d) Counting Game.

The outputs generated by carrying out the process of counting game according to procedure defined in 3.2(d) are as follows:

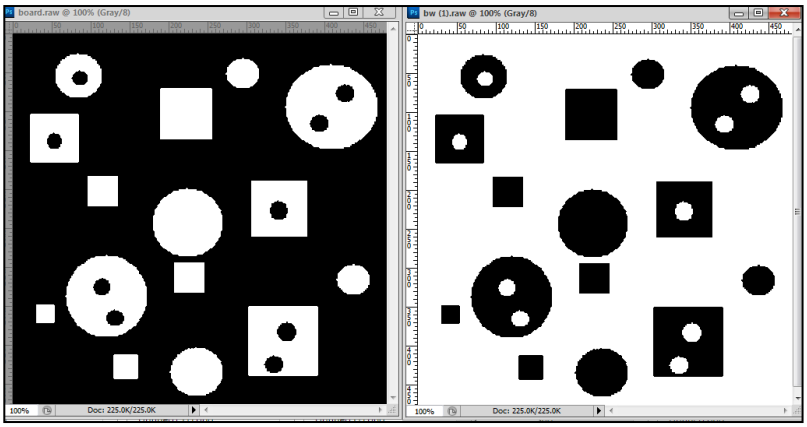


Fig 3.11: Complement operation carried out on the board image followed by shrinking to estimate the number of holes in the image.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

```
Iteration success for shrinking of holes:1
Iteration success for shrinking of holes:2
Iteration success for shrinking of holes:3
Iteration success for shrinking of holes:4
Iteration success for shrinking of holes:5
Iteration success for shrinking of holes:6
Iteration success for shrinking of holes:7
Iteration success for shrinking of holes:8
Iteration success for shrinking of holes:9
Iteration success for shrinking of holes:10
Iteration success for shrinking of holes:11
Iteration success for shrinking of holes:12
Iteration success for shrinking of holes:13
Iteration success for shrinking of holes:14
Iteration success for shrinking of holes:15
Iteration success for shrinking of holes:16
The total number of holes: 9
```

Fig 3.12:Output of total 9 holes after 1st stage as mentioned above.

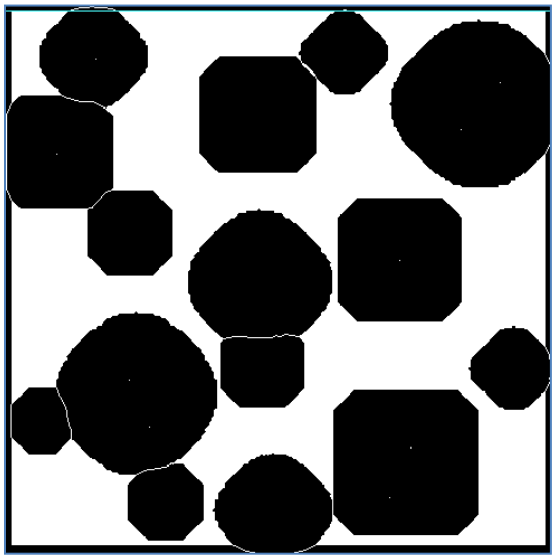


Fig 3.13: The output of the shrinking in the 1st stage after complementing which gives the number of holes.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

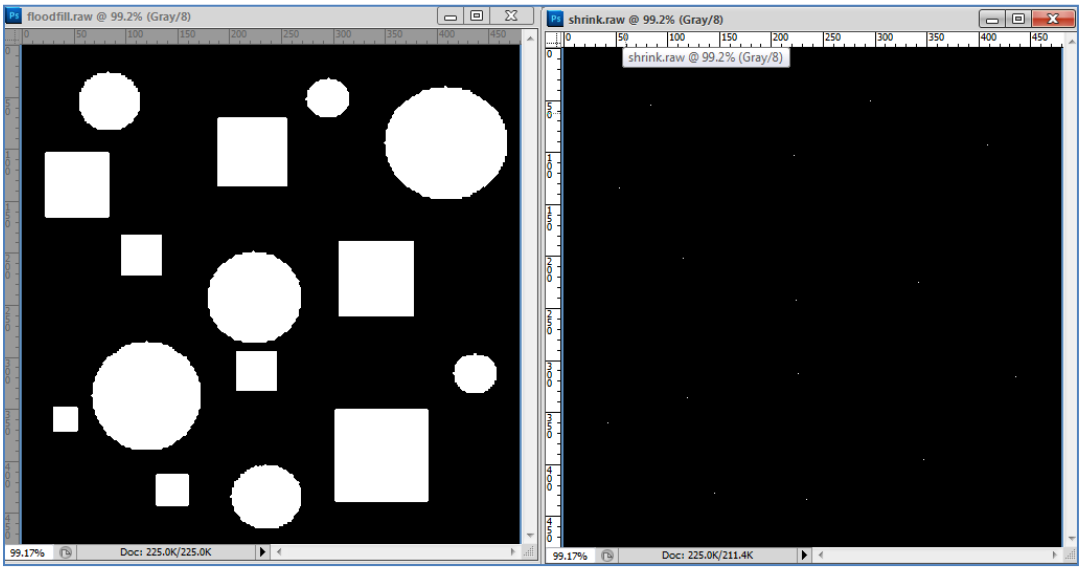


Fig 3.14: The output of flood-fill followed by the shrinking process to obtain the total white objects.

```
Iteration success for number of white objects :66

Iteration success for number of white objects :67

Iteration success for number of white objects :68

Iteration success for number of white objects :69

Iteration success for number of white objects :70

Iteration success for number of white objects :71

Iteration success for number of white objects :72

Iteration success for number of white objects :73

Iteration success for number of white objects :74

Iteration success for number of white objects :75

The number of white objects: 15


The number of squares: 8

The number of circles: 7
File write successful
```

3.14: The final output of the total objects and the final number of squares and circles in the image.

Aakash Shanbhag

adshanbh@usc.edu

3205699915

3.4 Discussion and Conclusion.

According to the requirements all the geometric modifications were made and successful results were obtained in each case. The detailed procedure and method is discussed above and some important conclusions for each of the portions are as follows:

Shrinking

Since shrinking involves 2 phase morphology in order to ease the computation of running a 5x5 mask which is computationally complex and can be replaced by a two masks of 3x3 which reduces the computation considerable and also helps in elimination of the cases when there are even number of white pixels which need to be eliminated. All the filters used in this case are subtractive. Repetitive processing is carried out until the steady state is not reached. These advanced filters have to be used iteratively. The application in counting the foreground and estimation of size is discussed in the figures above. The count obtained in the above case is 24 and the histogram of the sizes is also displayed which signifies that it each iteration the width and the height are reduced considerably to change the foreground to 1 pixel size foreground. The transition with iterations is shown in the following figures for 2, 5,8,10 iterations respectively. Shrinking can be used for shapes of any size and type without any constraints.

Thinning

The process of thinning is carried out on the letterE.raw image and the outputs corresponding to 20, 40, 60 and 80 iterations are shown above in the figures 3.4, 3.5, 3.6 above which provide evidence that the letter E can be made thin to a maximum threshold steady state beyond which there is no further changes. This can be ascertained by comparing the output of the previous stage with that of the current state and finding out the difference between them. The thinning process fails for shapes such as circular spheres as foreground cannot be ascertained by the shape using the thinning process and hence has lesser applications than shrinking and skeletonizing. The exact expected output is not obtained in this case since there must be some missing patterns which are not taken into consideration in the elimination of the structures generated. Hence instead of obtaining the exact the Letter E, we obtain some extrusions which are due to lack of the pattern matching that occurs.

Skeletonizing

The process of skeletonizing is carried out on the letterE.raw image and the outputs corresponding to 20, 25, 30 and 42 are shown above in the figures above which provide evidence that

Aakash Shanbhag

adshanbh@usc.edu

3205699915

the letter E can be skeletonized to a maximum threshold steady state beyond which there are no further changes. This can be ascertained by comparing the output of the previous stage with that of the current state and finding out the difference between them. The edges can be calculated by using this process of skeletonizing. Medial axis skeleton is generated in most cases which is pretty evident from the above figures.

Counting Game

The counting game example is an amalgamation of all the logical and morphological functions which includes the stages of complementing, shrinking, counting and correspondingly mapping the output with flood-fill or interior fill along with the process of shrinking and counting corresponding to the location of the centroid that lies inside a particular shape.

The answers for each of the questions are as follows:

1. There are total 15 objects which are white.
2. The total number of holes is equal to 9.
3. The total number of white squares is equal to 8.
4. The total number of circles is equal to 7.