EE5175: Image Signal Processing - Lab 1 report

Ruban Vishnu Pandian V (EE19B138)

February 2, 2023

Contents

1 Geometric transforms

Aim: To perform the following tasks on mentioned images:

- Translate the given image (`lena_translate.png`) by $t_x = 3.75, t_y = 4.3$ pixels.
- Rotate the given image (`pisa_rotate.png`) about the image centre, so as to straighten the Pisa tower.
- Scale the given image (`cells_scale.png`) by 0.8 and 1.3 factors.
- Use bilinear interpolation during target-to-source mapping.

2 Bilinear interpolation:

It is an interpolation method to determine the intensity of a target point. After doing target-to-source mapping, we cannot always assure that the source co-ordinates are perfect integers. In most cases, they will have fractional parts and hence, will lie within the square enclosed by the neighbouring four integer co-ordinates. Sometimes, the source co-ordinates may even completely lie outside the image dimensions (In which case, an intensity of 0 is given to the target point). But assuming the source co-ordinates are fractional and lie within the image, we have the following interpolation formula:

$$I(x, y) = (1 - a)(1 - b)I(x', y') + (1 - a)(b)I(x', y' + 1) + (a)(1 - b)I(x' + 1, y') + (a)(b)I(x' + 1, y' + 1) \tag{1}$$

where x' is the floor of x and y' is the floor of y. Pictorially, this interpolation method is represented as shown below:

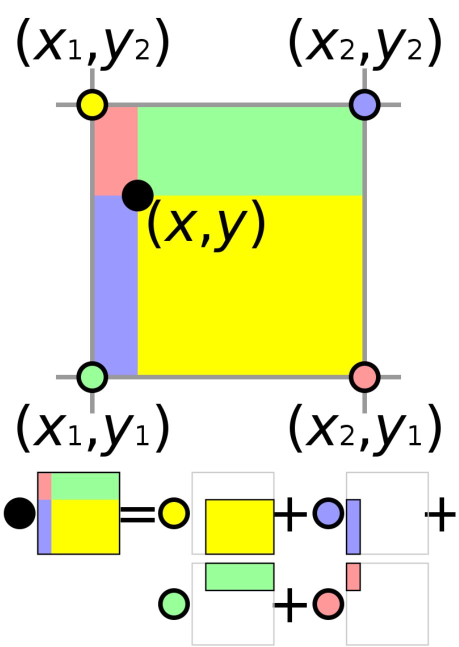

Figure 1: Bilinear interpolation

Note: In all the questions, target-to-source mapping is done. Target co-ordinates are assumed to be (i,j) and the corresponding source co-ordinates are (x,y) where i and x are row indices, j and y are column indices.

3 Image translation:

In this question, we've been asked to translate the given image `(lena_translate.png)` by $t_x = 3.75$ and $t_y = 4.3$ pixels.

Figure 2: Lena

The required target-to-source transformation is:

$$x = i - t_x$$

$$y = j - t_y$$

These source co-ordinates are now passed on to a bilinear interpolator function which returns the intensity value. This process is performed for all target image co-ordinates. The final target image is given below:

Figure 3: Lena translated

As expected, the target image has moved down by 3.75 pixels and to the right by 4.3 pixels. However, it's not as sharp as the original image since bilinear interpolation has averaged out the intensities and hence, sharpness in the target image is lower. It still has retained most of the features the source image has.

4 Image rotation:

In this question, we've been asked to rotate the given image `(pisa_rotate.png)` about the image center to make the tower straight.

Figure 4: Leaning tower of Pisa

By seeing the image, it is clear that the image has to be rotated anti-clockwise to make the tower straight. To perform target-to-source mapping which is an inverse rotation in this case, we first need to determine the center co-ordinates of the image. Let's assume that the image dimensions are $xmax$ and $ymax$. Then, we have:

$$xmid = \frac{xmax}{2} \qquad ymid = \frac{ymax}{2}$$

$$i\_cent\_ref = i - xmid \qquad j\_cent\_ref = j - ymid$$

With the target co-ordinates obtained w.r.t image center, clockwise rotation is performed by pre-multiplying the co-ordinates vector with a clockwise rotation matrix. The equation is given below:

$$\begin{bmatrix} x\_cent\_ref \\ y\_cent\_ref \end{bmatrix} = \begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} i\_cent\_ref \\ j\_cent\_ref \end{bmatrix}$$

Finally, to obtain the actual co-ordinates as row and column indices, we have:

$$x = x\_cent\_ref + xmid \qquad y = y\_cent\_ref + ymid$$

These source co-ordinates are now passed on to a bilinear interpolator function which returns the intensity value. This process is performed for all target image co-ordinates. It is observed that the angle tilt the Pisa tower has w.r.t the vertical is 4^o. Hence, the source image is rotated anti-clockwise by 4^o to get the target image. The final target image is given below:

Figure 5: Rotated tower of Pisa

The tower is now straight. However, the image once again suffers from the problem of lower sharpness due to bilinear interpolation. The edges too look ragged.

5 Image scaling:

In this question, we've been asked to scale the given image `(cells_scale.png)` about the image center by scaling factors $s_x = 0.8$ and $s_y = 1.3$.

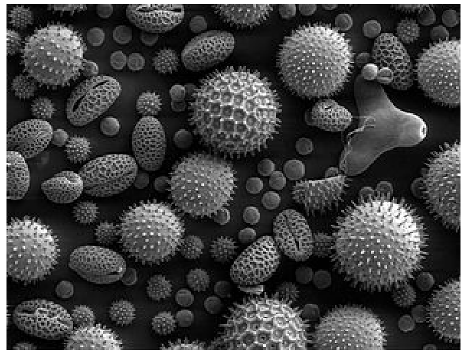

Figure 6: Cells

In this case, the required target-to-source mapping is:

$$x = xmid + \frac{(i - xmid)}{s_x}$$

$$y = ymid + \frac{(j - ymid)}{s_y}$$

These source co-ordinates are now passed on to a bilinear interpolator function which returns the intensity value. This process is performed for all target image co-ordinates. The final target image is given below:

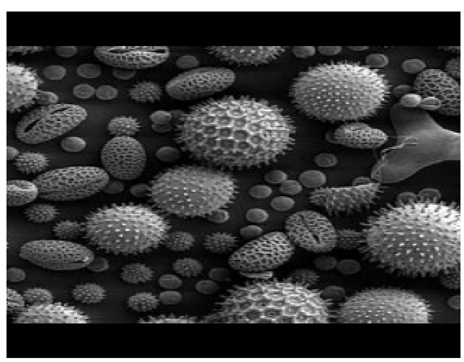

Figure 7: Scaled cells

As expected, the target image is scaled by 0.8 along the vertical axis and by 1.3 along the horizontal axis. This image too suffers from lower sharpness because of bilinear interpolation.

6 Observations and Conclusions

1. All the basic geometric transformations are quite elementary to implement and provide convincing results too.
2. Bilinear interpolation is a fairly simple interpolation algorithm and is easy to implement.
3. Lower sharpness is a big disadvantage of the bilinear interpolation method.
4. When images are rotated, the edges look ragged and better interpolation algorithms must be used to mitigate this issue.