Assignment 2

Paolo Deidda, Raffaele Perri

March 4, 2025

Problem 1

The key idea behind backward mapping is that for each pixel (i, j) in the target image A_2, we map it back to the original image A_1 to compute its intensity.
Concerning the backward mapping formula, by using the scale factor $c = 0.25$, the corresponding coordinates in A_1 for pixel (i, j) in A_2 are:

$$x = \frac{i + 0.5}{c} = 4i + 2, \quad y = \frac{j + 0.5}{c} = 4j + 2$$

Since the mapped point (x, y) may not be an integer, we use bilinear interpolation to approximate the intensity value. The four neighboring integer pixel coordinates in A_1 are:

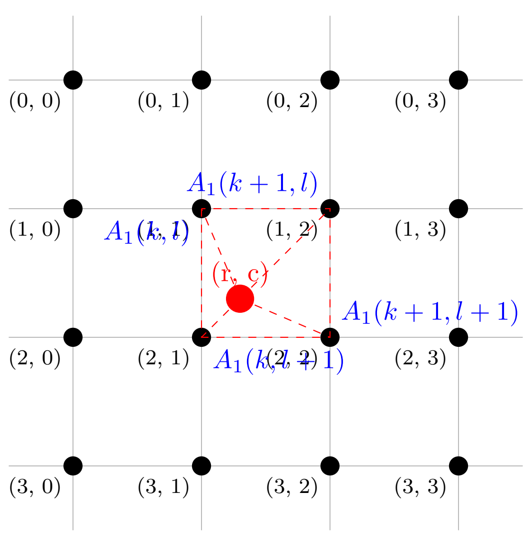
$$(k, l) = (\lfloor x \rfloor, \lfloor y \rfloor)$$

The four neighboring pixels used for bilinear interpolation are:

$$(k + 1, l), \quad (k, l + 1), \quad (k + 1, l + 1)$$

The intensity at pixel $A_2(i, j)$ in the target image is computed using bilinear interpolation:

$$A_2(i, j) = (1-v)\left[(1 - u)A_1(k, l) + uA_1(k + 1, l)\right] + v\left[(1 - u)A_1(k, l + 1) + uA_1(k + 1, l + 1)\right]$$



This method provides smooth and high-quality results because it estimates pixel intensities with bilinear interpolation rather than directly picking a nearby pixel.

This can be noticed by looking at the simple zooming out algorithm (slide 14) we divide the original image into non-overlapping blocks of size $z \times z$.
The intensity of each pixel in the target image is computed as the average of the corresponding block in the source image.

$$A_2(i,j) = \sum_{k=0}^{z-1} \sum_{l=0}^{z-1} A_1(z_{i+k}, z_{j+l})$$

This approach preserves details well but can cause aliasing artifacts, as it does not account for subpixel contributions.

Comparison

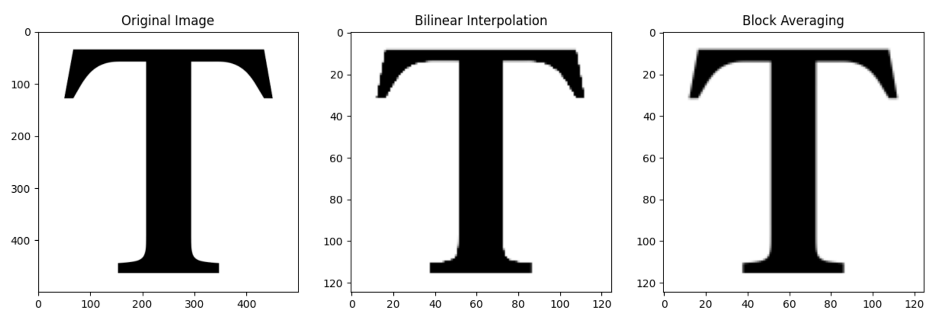


Figure 1: Comparison between the two methods with a scale factor of 0.25

By looking at the figures we can clearly see that:

- Backward mapping with bilinear interpolation provides smoother results because it considers contributions from multiple pixels, avoiding harsh artifacts.
- Averaging over blocks is computationally cheaper but can cause aliasing and loss of details because it doesn't properly weight pixels near the edges of the block.
- The backward mapping method also preserves more fine details and prevents jagged or blocky effects.

Thus, bilinear interpolation is usually the better approach for high-quality image downscaling.

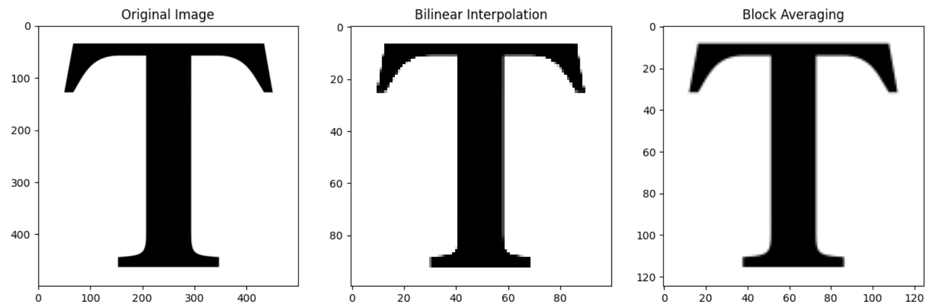


Figure 2: Comparison between the two methods with a scale factor of 0.2

With a scale factor of 0.2, the formulas remain the same, but now each pixel in A_2 corresponds to a larger area in A_1.
The backward mapping approach will still work well, as it will continue interpolating between appropriate pixels.
The block averaging method will discard even more details, increasing the chance of aliasing artifacts.

Problem 2

, To rotate the image, we use the following steps:

1. First, the image is moved to the origin using a translation matrix T_1, which shifts the image center to the origin.
2. Next, two shear matrices are defined: S_h for horizontal shearing and S_v for vertical shearing. These matrices incorporate the shear factors derived from the rotation angle θ.
3. The shearing operations are applied in a specific order to approximate the rotation. The transformation matrix is computed by combining translation and shearing matrices:
$$A = T_2 \cdot S_h \cdot S_v \cdot S_h \cdot T_1$$
where T_2 translates the image to the center of the new canva.
4. The inverse of the matrix A is computed and applied to perform the transformation and map each pixel in the target image back to the source image.
5. Finally, one-dimensional interpolation in the direction of the current sheer is used to estimate the pixel values in the rotated image.

The image was successfully rotated by 25 degrees using the described method. The following figures 3 show the result of the image rotation, first the one done in class with the rotation matrix, in the middle the one with the sheering process and lastly the absolute difference between the first two.

Figure 3: rotation matrix - sheering rotation - images difference

The average difference between the two images is approximately 1.32, computed as the mean absolute difference between the pixel values of the two images. This difference arises due to the three-step shearing process used for rotation, which involves three rounds of pixel rounding, leading to greater approximations compared to direct rotation.