2️⃣

Homework #2

Luca Airaghi, Federico Romano

Problem 1

Suppose we scale a given image by a scale factor of $c = 0.25$, using backward mapping and bilinear interpolation. We derive the formula for the intensity of the $(i, j)_{th}$ *pixel of the target image, expressing* $a^2_{i,j}$ in terms of the source image intensities $a^1_{k,l}$. We also compare this approach to the simple zoom-out algorithm that scales the image by a factor of $z = 1/c = 4$

Scaling Image Using Backward Mapping and Bilinear Interpolation

- Given the dimensions M_1, N_1 of the original image A_1, the dimensions of the target image A_2 are computed as:

  $$M_2 = \text{round}(c \cdot M_1), \quad N_2 = \text{round}(c \cdot N_1)$$

- For each pixel (i, j) in A_2, we determine the corresponding position (x, y) in A_1 using:

  $$x = \left((i + 0.5) \cdot \frac{M_1}{M_2} - 0.5\right), \quad y = \left((j + 0.5) \cdot \frac{N_1}{N_2} - 0.5\right)$$

- Identifying the four nearest pixels in A_1:
  - $A_1[k, l]$ → top-left pixel
  - $A_1[k + 1, l]$ → bottom-left pixel
  - $A_1[k, l + 1]$ → top-right pixel
  - $A_1[k + 1, l + 1]$ → bottom-right pixel
- Defining interpolation weights:

  $$u = x - k, \quad v = y - l$$

- For $c = 0.25$:

$$x = 4(i + 0.5), \quad y = 4(j + 0.5)$$

$$k = \text{round}(x) - 1, \quad l = \text{round}(y) - 1$$

$$u = x - (k + 0.5), \quad v = y - (l + 0.5)$$

- Applying bilinear interpolation:

$$a^2_{i,j} = (1 - v)\left[(1 - u)a^1_{k,l} + ua^1_{k+1,l}\right] + v\left[(1 - u)a^1_{k,l+1} + ua^1_{k+1,l+1}\right]$$
$$a^2_{i,j} = \text{round}\left(\frac{1}{4}\left[a^1_{4i+1,4j+1} + a^1_{4i+2,4j+1} + a^1_{4i+1,4j+2} + a^1_{4i+2,4j+2}\right]\right)$$

Simple Zoom-Out Algorithm

- The technique used in the other algorithm instead divides the source image into “blocks” of size $z \times z$, where $z = \frac{1}{c}$ and assigns to each pixel in the final image the average value of all the pixels contained in the corresponding “block.” This technique consists of a kind of lowpass filter, that is, it partially attenuates or eliminates the rapid variations (understood as high frequencies) of an image, while leaving intact the more gradual variations (defined instead as low frequencies). This is achieved by averaging sometimes over a large number of pixels, such as 16 in the case of $z = 4$.

Comparison of the Two Methods

- **Bilinear interpolation** uses only the four closest pixels, with weights based on their distances. This produces smoother transitions and avoids abrupt changes in pixel values.

- **Simple zoom-out** considers all pixels within a block, effectively acting as a low-pass filter that reduces high-frequency details.

For significant downscaling:

- **Simple zoom-out** is preferable, as it integrates more information and prevents aliasing artifacts.

- **Bilinear interpolation** may lose details due to limited sampling.

Effect of Using c = 0.2

For $c = 0.2$, the zoom factor becomes $z = 5$. The block averaging method now computes each pixel as the average of a 5×5 block, improving aliasing suppression. Bilinear interpolation still relies on only four pixels, leading to greater detail loss.

In conclusion, while bilinear interpolation provides smoother transitions, block averaging preserves more global information when downscaling significantly.