TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA
ROMANIA

Facultatea de Electronică,
Telecomunicații și
Tehnologia Informației

TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA
ROMANIA

# Zooming and shrinking images by Pixel Replication and Bicubic Interpolation

-Technical Report-

Telecommunications Technologies and Systems, Cluj-Napoca, 12 December 2022

Coordinator teacher:
Sl.dr.ing. Cîşlariu Mihaela

Students:
Raluca Nelega
Elena-Sabina Onuţ
Diana-Elena Mircea

# Table of Contents

# 1. State of The Art

Image scaling plays a definite role when trying to compress and transfer images: "*if you upload a very large image to a website and scale it down to a smaller size, the website still must load the full size version of that image and could cause the web page to load more slowly*" [1].

There are two operations that can be performed upon modifying the dimensions of an image: resampling- change the total number of pixels contained in the image- or resizing-change the physical size of an image while maintaining the pixel dimensions constant. In highly used image processing applications, such as Photoshop, resizing can be done by keeping the number of pixels contained in an image, but with different resolution (pixels per inch). Therefore, by scaling down, resolution is improved, while by scaling up, the same number of pixels would have to fill a greater space, and the image quality decreases [2]. This paper summarizes the applications of image processing algorithms "Pixel Replication" and "Bicubic Interpolation" which are encountered in many fields like medical images, industrial technology, archeology, military, mobile phone cameras. Bicubic Interpolation is nowadays used many image processing software such as Photoshop, After Effects, Avid and Final Cut Pro [6]. In comparison to other recent papers [3], where these 2 algorithms were somehow combined to give a fast computation with good quality, we concentrated on testing a straight forward implementation of these basic algorithms and see what potential do they have in a real-time application.

# 2. Introduction

In this modern era of high-performance computing machines, imaging applications are becoming widely used in our everyday life. The most common and practical method of storing or transmitting information is through digital images. They share information regarding the positions, sizes, and relationships between various objects.

Two of the fundamental image operations, image zooming and image shrinking are frequently employed in numerous applications and are related to image sampling and quantization. Zooming may be viewed as oversampling, and shrinking may be viewed as under sampling [4]. Digital image zooming is a significant process in image processing. Basically, there are two phases needed in zooming: creating new pixels and assigning a gray level to those new pixels. Digital image shrinking is done in same manner as zooming in.

The fact that zooming and shrinking are performed to a digital image is the key difference between them and sampling and quantizing an original continuous image.

Image zooming and shrinking are achieved by Pixel Replication and Bicubic Interpolation (BI) methods, which will be presented in the next chapter.

# 3. Theoretical Fundamentals

## 1. Image Zooming

Magnification of an image, meaning an increase in the number of pixels constituting an image, is a process performed to achieve a higher resolution image. If we take for example a simple image, made of 2x2 matrix, with 4 pixels, each in one corner (X, Y, Z, Q), when it will be magnified, (X, Y, Z, Q) pixels will constitute the corners, while the rest of the free locations will need to be filled with new values, as in Figure 1. These pixel's values, like P, will be estimated using interpolation algorithms. In this paper we describe and compare only two of them: Pixel Replication, and BI. "*Interpolation is the process of calculating the intermediate values of a continuous event from available discrete samples*" [5].

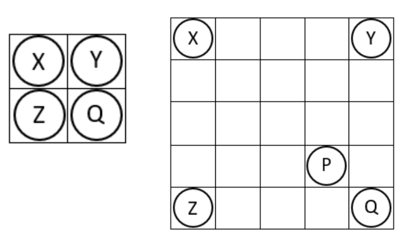



*Figure 1. Schematic diagram of image magnification*

## 1.1 Pixel Replication

Pixel Replication is a particular case of nearest neighbor interpolation, which computes the distances from the free pixel, P(i,j) (a pixel is described by the column and lines occupied in an MxN matrix), to its neighboring points taken as reference (X,Y,Z,Q). The gray levels of pixel P will be assigned with the values of the nearest pixel in the original image. The nearest pixel will be decided as the one that has the smallest distance from P(i,j) [6].

This algorithm can be applied only if it is desired to zoom an image by an integer number of times. Therefore, the gray values of the new pixels will be assigned such that the created locations will be placed at exactly multiples of the old locations in the original image [4].

## 1.2 Bicubic Interpolation (BI)

If more neighbors are taken into consideration for interpolation, the computations will be more complex, but will produce smoother results [4]. For BI, the pixel's P value will be computed using its 16 adjacent pixels. Its gray levels are assigned based on the distances towards other 16 pixels.

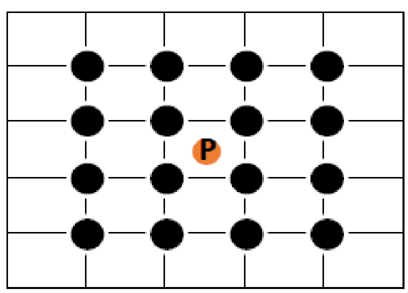



*Figure 2*

The interpolated surface is fitted among 4 corner pixels, and it is described by third-order polynomial function in Eq. (1). The interpolation is computed using the locations: derivate at the 4 corners on vertical, horizontal and diagonal axis, and also the intensity values of those pixels. The function in (1) is computed based on the distances between each one of the 16 coefficients towards the reference pixel. [5]

$$p(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij}\, x^i y^j \qquad (1)$$

The questioned pixel is calculated for every row in Eq. (2). It describes the interpolated pixels on the horizontal axis [P1', P2', P3', P4′]. Eq. (3) gives the final values of the pixel, where $wr_i$ are the coefficients from the $i^{th}$ row, and $wc_i$ are the coefficients of the $i^{th}$ column [3].

$$Pi' = \sum_{k=1}^{4} p_{ik} * w_{r_k}(dx) \quad , \quad \text{i} = [1,2,3,4,] \quad (2)$$

$$P = \sum_{k=1}^{4} P'_k * w_{C_k}(dy) \qquad (3)$$

For simplification, a kernel like Eq. (4) is used to approximate the coefficients [5]:

$$W(d) = \begin{cases} \frac{3}{2}|d| - 5/2|d|^2 + 1, & 0 < |d| < 1 \\ -\frac{1}{2}|d|^3 + \frac{5}{2}|d|^2 - 4|d| + 2, & 1 < |d| < 2 \\ 0 & 0.W \end{cases} \qquad (4)$$

### 2. Image Shrinking

The process of shrinking an image is similar to zooming. Instead of filling newly created pixels, each row and column will be deleted accordingly [4].

# 4. Implementation

### 1. Method of Bicubic Interpolation:

1.1 Write the Interpolation Kernel Function:

The interpolation kernel uses a coefficient *a,* to help decide the performance of the estimation. For best performance, it usually lies between -0.5 and -0.75 [8].

For all the computations, we used *a* as -0.5.

1.2. Pad the initial image:

It may be required to extend the border of the image such that the pixel values of all the destination locations lying along the boundaries to be computed. This can be done, by adding a padding function to the input image, filling all the spaces with zeros.

1.3. Write the BI algoritm & generate the resized image:

We defined new dimensions by creating height (*dH*) and width (*dW*) dimensions separately by rounding down the product between height and width values with their specific factors: *width_factor, height_factor*. Then, we created a null matrix with the dimensions formed above. After we got the coordinates of the nearby values (as *x* and *y*) in the range of *dH* and *dW*, we obtained 4 coordinates on X-horizontal axis *(x1, x2, x3, x4)* and 4 values on Y-vertical axis *(y1, y2, y3, y4)*; By interpolating them inside a matrix we obtained 16 values of the new positions in *mat_m* matrix. We used the interpolation kernel to approximate the values of the coefficients on both axes (in matrices *mat_l* and *mat_r*, respectively). The final image is constructed from dot product between all three matrices from above (returned in *dst*).

## 2. Method of Pixel Replication

Similar to the Bicubic function, we chose height and width factor to scale the image, created a new grid based on these new dimensions, and then filled it up - with the new pixel's values. This is done by indexing the new matrix for each row and column, finding the ratio between the new size and the old size separately, for row and column, and then round down the values obtained for each ratio. With these values of the new pixel's locations, we interpolate row-wise and column-wise simultaneously the pixel's values from the old locations to the new ones, until each output pixel will be replaced by its nearest pixel in the input.

## 3. The graphical user interface

It allows the user to select a gray scale image (that has to be *.png* format). The reason for using only gray scale images is to minimize the quantity of information that has to be processed. User can size it by choosing width and height factors, scale the image up or down by the preferred algorithm (*Pixel Replication* or *BI)* and resize the image to the original specifications. It also plots underneath each picture the histogram (intensity given by the number of pixels represented), and it computes the MSE- Mean Square Error- between the initial image and the rescaled image. An example can be seen in Fig.3.

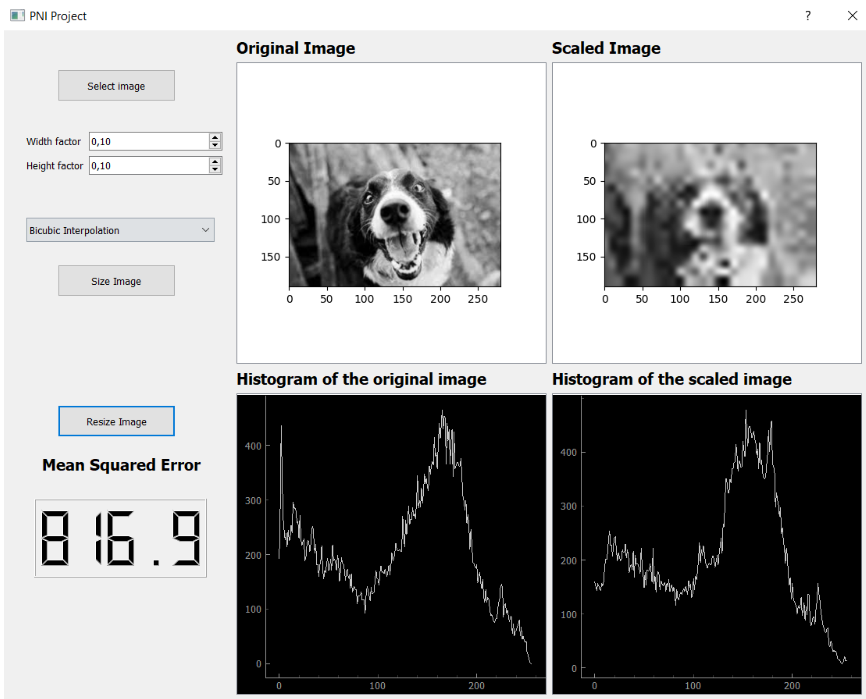



*Figure 3 - Graphical User Interface*

# 5. Experimental Results

During the experiments we ran, we focused on testing the performance of both algorithms, by judging the visual clarity of the sized image and the differences between the input image and the resized image, in the form of Mean Square Error.

A core observation noted that, for both algorithms, the smaller was the sizing factor, the greater was the MSE, which showed the degree by which the image was distorted. Table 1 contains the results obtained by sizing different factors on the same image. Also, if we chose different sizing factors for weight and height, the MSE is smaller compared to when both of them have the same value.

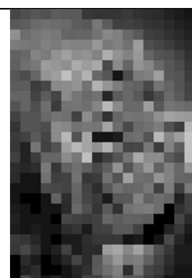
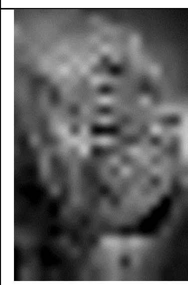
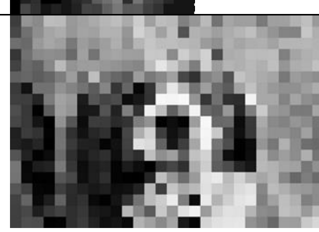
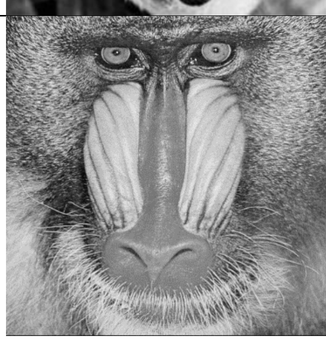
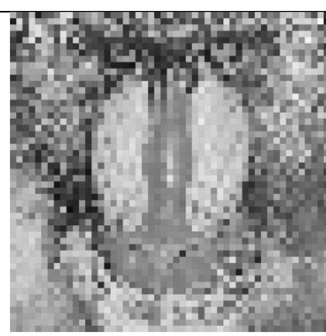
| Width | Height | MSE | MSE |
|---|---|---|---|

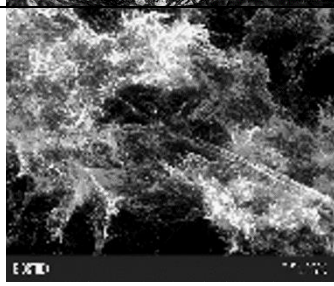
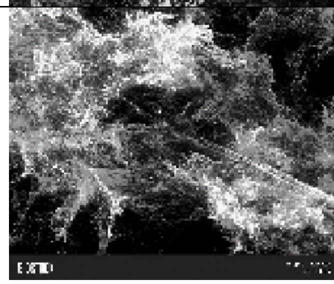
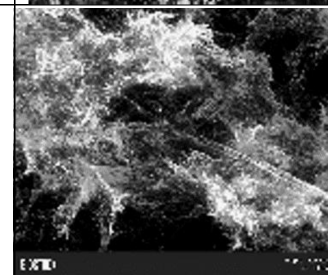
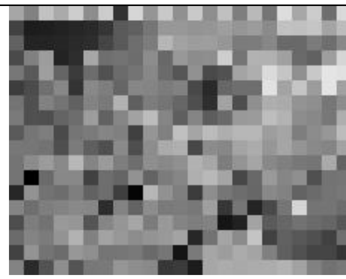
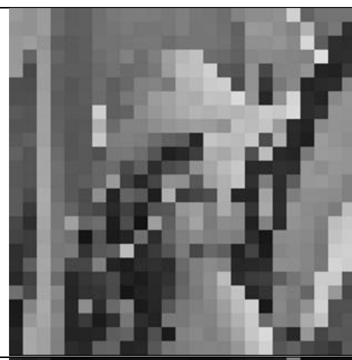
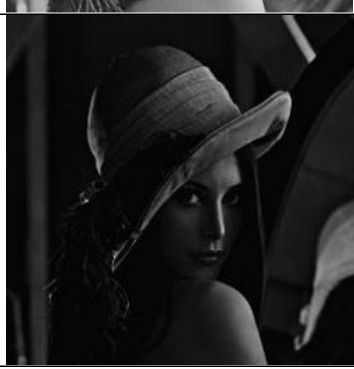
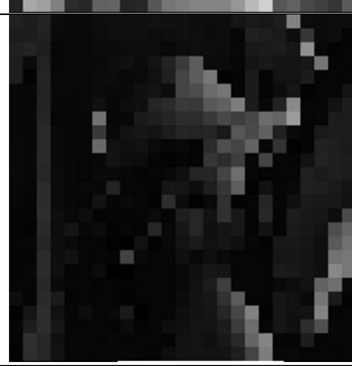
| | | Pixel Replication | Bicubic interpolation |
|---|---|---|---|
| 0.1 | 0.2 | 927.1 | 1.587 |
| 0.1 | 0.1 | 1536 | 0.513 |
| 0.05 | 0.05 | 2773 | 1278 |

*Table 1- Different scaling upon a picture of **1500x1000px***

Our goal was to track the MSE of images of different sizes that were first scaled down 10 times and then brought back to the original dimensions. The results are illustrated in Table 2.

*Table 2-Results after scaling down 10x and then back to original*

| Nr | Dimen sions [px] | Original Image | Resized-Pixel Replication | MSE | Resized-Bicubic Interpolation | MSE |
|---|---|---|---|---|---|---|
| 1 | 180x270 |  |  | 1203 |  | 624.6 |
| 2 | 280x190 |  |  | 2085 |  | 816.9 |
| 3 | 510X510 |  |  | 1915 |  | 1301 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 4 | 1500x1000 |  |  | 1536 |  | 1300 |
| 5 | 1500x1240 |  |  | 1661 |  | 979.7 |
| 6 | 230x180 |  |  | 2210 |  | 1364 |
| 7 | 250x250 |  |  | 1847 |  | 765.8 |
| 8 | 250x250 |  |  | 577.4 |  | 241.8 |

Also, in Figure 4, the histogram of the image provided by the BI is much more similar to the original image histogram, than that of the Pixel Replication, showing a less distorted result.

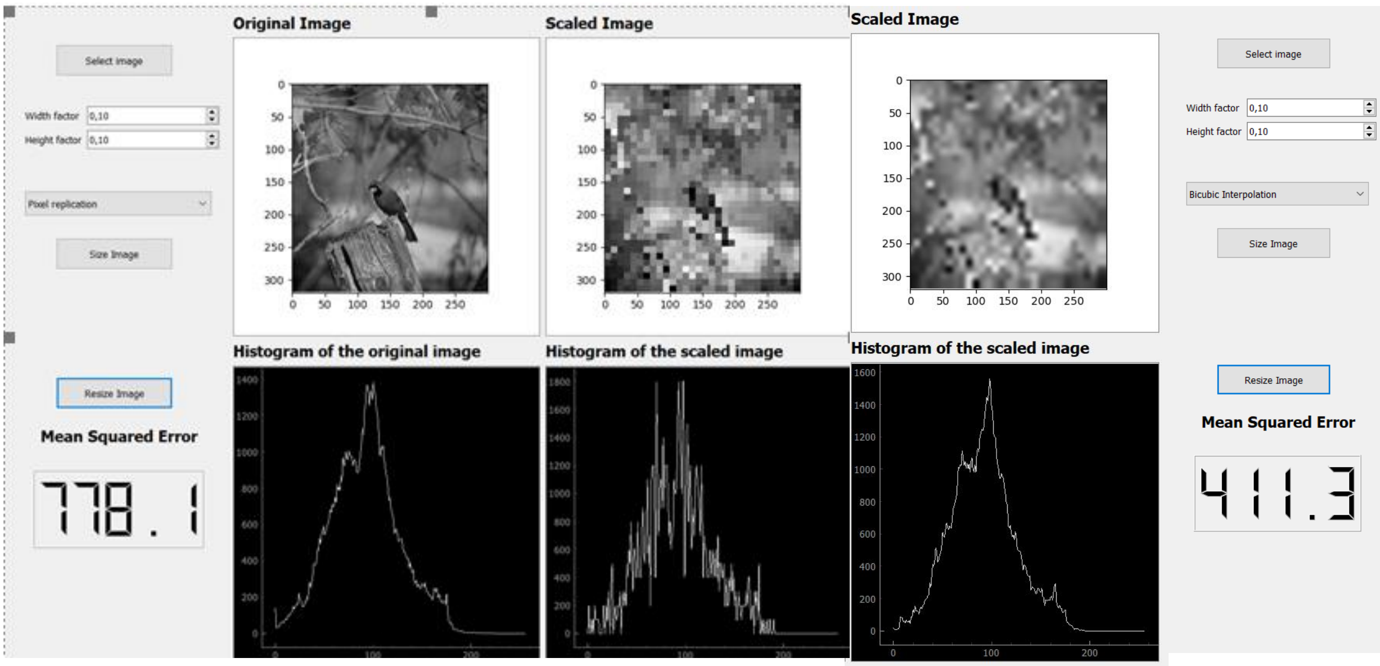



*Figure 4-Pixel Replication vs Bicubic Interpolation*

# 6. Conclusions

Regarding the visual quality of the recreated images, BI algorithm provided better results, with smoother curves, getting closer to the details in the original image, as opposed to more pixelated images given by Pixel Replication. In all cases, BI provided a smaller MSE and better quality images. On the other hand, Pixel Replication requires a significant shorter time to be executed than BI, that has more complex computations to perform.

As expected, the results in Table 2. show that for bigger images, a higher MSE is obtained, because the grid has to be compressed and extended to gather more information, for the same scaling factor of 1/10. Also if we take images with similar dimensions, but with different white and black grading, MSE is bigger for those containing more white than black.

In conclusion, even though BI is more complex than Pixel Replication, it produces better quality images, and less blurring effect. A difference between the original and reconstructed image appears because when an image is minimized, its resolution is poorer and its grid has to encapsulate more information in a smaller place, therefore it loses its quality when someone tries to bring it to its original size.

# 7. Bibliography

[1] **University of Michigan,** 7 July 2022, https://guides.lib.umich.edu/c.php?g=282942&p=1885347, 04.12.2022

[2] **Adobe** 16 Nov 2022, <https://helpx.adobe.com/photoshop/kb/advanced-cropping-resizing-resampling-photoshop.html>, 04.12.2022

[3] Sabrin, Kaeser & Ali, Md. (2014). An Intelligent Pixel Replication Technique by Binary Decomposition for Digital Image Zooming, 3.01.2023

[4] **R. Gonzalez and P. Wintz**, *Digital Image Processing, 2nd edition*, Addison-Wesley Publishing Company, 1987, pp 64– 69.

[5] **Khaledyan, Donya & Amirany, Abdolah & Jafari, Kian & Moaiyeri, Mohammad & Zargari, Abolfazl & Mashhadi, Najmeh** (2020), Low-Cost Implementation of Bilinear and Bicubic Image Interpolation for Real-Time Image Super-Resolution.

[6] **Han, Dianyuan** (2013), Comparison of Commonly Used Image Interpolation Methods. Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering.

[7] **Bilal Himite**, Aug 16, 2021, *https://towardsdatascience.com/image-processing-image-scaling-algorithms-ae29aaa6b36c*, 27.11.2022.

[8] 08 May, 2021, https://www.geeksforgeeks.org/python-opencv-bicubic-interpolation-for-resizing-image/, 03.12.2022

[9] **A. Jain** *Fundamentals of Digital Image Processing*, Prentice-Hall, 1986, Chap. 4

[10] **E. Davies** *Machine Vision: Theory, Algorithms and Practicalities*, Academic Press, 1990

[11] **A. Marion** *An Introduction to Image Processing*, Chapman and Hall, 1991