Evaluation of Rounding Functions in Nearest Neighbor Interpolation

Olivier Rukundo
Norwegian Colour and Visual Computing Laboratory, Department of Computer Science,
Norwegian University of Science and Technology, Gjøvik, Norway

ABSTRACT

A novel evaluation study of the most appropriate round function for nearest neighbor image interpolation is presented. Evaluated rounding functions are selected among the five rounding rules defined by the Institute of Electrical and Electronics Engineers IEEE 754-2008 standard. Both full- and no-reference image quality assessment metrics are used to evaluate the influence of rounding functions on nearest neighbor interpolation image quality. The concept of the number of occurrences is used to determine the percentage of achieved occurrences – which is also equivalent to the sample proportion percentage used to calculate the margin of error. Inferential statistical analysis is used to deduce from a small number of images and draw a conclusion of the performance of each rounding function on a big number of images. Experimental results are provided to demonstrate with 95% confidence the minimum performance percentage of the most appropriate rounding function.

Keywords: nearest neighbor interpolation, number of occurrences, inferential statistics, rounding functions.

1. INTRODUCTION

Interpolation is a widely used method, in many fields and applications, to construct a new data value within the range of a set of known data [1],[2],[3],[4],[5],[6],[23]. In the video and/or dynamic imaging, if the interpolation method becomes too computationally inefficient or time-consuming, it may lead to the jerky appearance of images. In image upscaling or resolution enhancement, if the interpolation method is not accurate enough, it may result in heavy error propagation or visual artefacts, particularly at the edges of upscaled or enlarged image objects. Visual artefacts - such as aliasing/jaggy, blurring, and edge-halo artefacts - are important contributors to the loss of image interpolation quality. Such artefacts become more easily visible when the scaling ratio is significantly increased, thus significantly reducing the image interpolation quality, which ideally should not have to be the case. In digital zoom, one of the advantages of image upscaling is the possibility to get a closer view of objects in small-sized images and/or videos [7],[8],[9],[10], without the need for a mechanical device of lens elements such as the one used in optical zoom. Many works on image interpolation reported new strategies for the minimization of visual artefacts, at specific scaling ratios. Those strategies can be classified into adaptive [11],[12],[13] and non-adaptive [14],[15],[16] as well as, very recently, into non-extra pixel and extra pixel categories [21]. The nearest-neighbor (NN) image algorithm belongs to the non-extra pixel category and its performance depends entirely on the accuracy or precision of the selected or used rounding function. The computational simplicity of such a rounding function gives the NN algorithm its main advantage of being the fastest and crispest image-edge productive among other/existing image interpolation algorithms [3],[5],[15]. The disadvantage of using the nearest-neighbor algorithm for image interpolation is the production of upscaled images with the most jagged-edges among other well-known algorithms [5]. Such a disadvantage is mainly linked to the loss of precision of a given rounding function used to round-off any scaled coordinates of non-integer type [21]. There exist many rounding functions and rules which can round-off a non-integer output to an integer output. The IEEE 754-2008 standard defined five rounding rules [17]. Here, rounding-off or rounding a non-integer means transforming some non-integer quantity from a greater precision to lesser precision [22]. In NN image interpolation, such a lesser precision has a direct effect on which pixel to pick from the source image and copy in the destination or upscaled image. Therefore, in the attempt to answer the question of the most appropriate rounding function that, from its imprecision, yields the least error propagation damages on the NN image interpolation quality, the author evaluates different rounding functions using image quality assessment metrics, the number of occurrences and inferential statistical analysis. The rest of the paper is organized as follows: Part 2 recaps the scaling, the rounding rules, and the nearest-neighbor interpolation algorithm. Part 3 gives the example of coordinates translation and pixel replication in NN interpolation. Part 4 presents the experimental results and discussions. The conclusion is given in Part 5.

2. SCALING, ROUNDING FUNCTIONS AND NEAREST NEIGHBOR INTERPOLATION

a) Scaling

Referring to the widely known straight-line equation Eq.1, when x gets multiplied by m, it is called a **scaling** by a factor m [32]. When b is added to the product $m\,x$, it is called a **shift** by an amount b. Here, m and b are constants, and the *linear function* takes a number x as input and returns the number $m\,x + b$ as output [32].

$$y = m\,x + b \tag{1}$$

In digital image processing, scaling image from original or input size to a new or output size requires translating from original image coordinate system to a new coordinate system according to the chosen factor or scaling ratio. A solution to this simple mathematical problem of scaling is to proceed as follows: "the point that lies *n* percent along the original coordinate axis lies at m percent point along the new coordinate axis". For example, the image pixel or point that lies halfway along the original or source image axis corresponds to the pixel that lies halfway along the new or destination image axis [21]. Given a particular pixel coordinate on the destination axis it is possible to find the corresponding pixel coordinate or point on the source axis. This can be done by finding the proportion of the point along the destination image axis. The proportion of that destination point can be found by dividing it by the length of that destination axis and multiplying the result by the length of the source image axis (see Figure 1). The linear scaling Eq. 2 shows how the correspondence between pixel coordinates can be achieved for NN image interpolation - where the $srcLength$ is a variable representing the length of the source image, the $dstLength$ is a variable representing the length of the destination image, the scaling ratio $dstLength/srcLength$ is the constant of proportionality (that plays the same role as the factor $\boldsymbol{m}$ in Eq. 1.), the $srcCoord$ is a variable representing the source pixel coordinates and the $dstCoord$ is a variable representing the destination pixel coordinates.

$$\frac{srcCoord}{srcLength} = \frac{dstCoord}{dstLength} \tag{2}$$

Note that the $srcCoord$ and $dstCoord$ are conventionally expected to be of integer type quantities. To ensure this is respected when Eq. 2 outputs coordinates of non-integer type, rounding operation is required and must be performed to meet the digital format requirement and enable translation from one coordinate system to another.

b) Rounding functions

Round-off or rounding functions transform a non-integer quantity from a greater precision to lesser precision [22]. In NN interpolation algorithm, rounding functions allow creating integers and/or converting scaled coordinates of non-integer type to integer type coordinates. Table 1 shows five rounding rules defined by the Institute of Electrical and Electronics Engineers (IEEE) 754-2008 standard as well as Maxfield's diagram [17],[22]. The scope of this work encompasses three of five namely *floor*, *ceil*, and *round* with the main objective to evaluate their effects on NN image interpolation quality. As can be seen in Table 1, *floor* means rounding towards minus infinity, *ceil* means rounding towards plus infinity, and *round* means rounding to the nearest integer if a non-integer has a tie or if it is a half-integer round to the nearest integer away from zero.

Table 1: Five rounding rules defined by the IEEE 754-2008 standard

	EXAMPLES OF HALF-INTEGERS			
RULES	+11.5	+12.5	-11.5	-12.5
round to nearest, ties/half-integers to even	+12.0	+12.0	-12.0	-12.0
round to nearest, ties/half-integers away from zero (*round)*	+12.0	+13.0	-12.0	-13.0
round toward 0 (*fix)*	+11.0	+12.0	-11.0	-12.0
round toward +∞ (*ceil)*	-12.0	-13.0	-11.0	-12.0
round toward −∞ (*floor)*	+11.0	+12.0	-12.0	-13.0

c) Nearest neighbor interpolation

The nearest-neighbor interpolation, also known as zero-order interpolation, is the lowest level of computational complexity pixel replication method which is widely used in many applications [3], [5], [21], [36]. The NN interpolation algorithm is based on the linear scaling equation (Eq. 2) and rounding functions to achieve image interpolation operations, in digital zooming. Here, Eq. 2 allows scaling a given input image coordinates to a desired new image coordinates according to the selected scaling ratio. A chosen rounding function is subsequently used to round off any non-integer scaled coordinates. As an example, assume that *the* Dm represents the destination image and Sm represents the source image. Referring to Eq. 2, as well as properties of interpolation presented in [36], the output equation of the NN algorithm is written as shown by Eq.3.

$$Dm(dstCoord(c), destCoord(r)) = Sm(\lceil srcCoord(c) \rceil, \lceil srcCoord(r) \rceil) \qquad (3)$$

where $srcCoord(.) = dstCoord(.) * \left(\frac{srcLength}{dstLength}\right)$, (c) indicates the column direction, (r) indicates the row direction, and $\lceil . \rceil$ denotes the rounding to the nearest integer.

3. EXAMPLE OF COORDINATES TRANSLATION AND PIXEL REPLICATION IN NN

Figure 1 shows an example of the source and destination images with the coordinate and length information. As can be seen, the source image has four coordinates or indices, namely 1, 2, 3, 4, and its length equals four. The destination image has seven coordinates, namely 1, 2, 3, 4, 5, 6, 7, and its length equals seven. Since the source and destination images are not equal in lengths, the source image pixels are insufficient to fill in or complete the remaining (or empty) locations in the destination image.

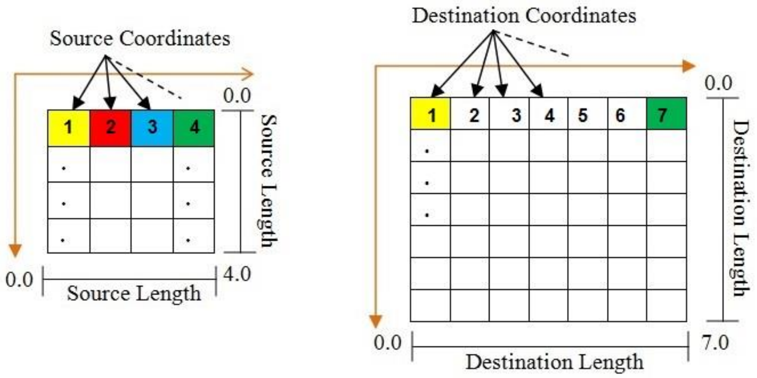

Figure 1: Example showing the source and destination images

It is, therefore, necessary to use the linear scaling Eq. 2 to find all coordinates correspondences and rounding them off to be able to approximate the missing pixels to fill in all empty locations in the destination image. Unlike other interpolation algorithms, the nearest neighbor algorithm does not create extra-pixels to find those additional pixels to use in the destination image[21]. Note that extra-pixels are pixels that do not belong to the source image[21]. Table 2 gives a numerical example showing how linear scaling and rounding operations generate the missing pixels in NN interpolation. As can be seen, the first column represents the destination coordinates. The second column represents the output of Eq.2 with its variables. The third column contains linearly calculated source coordinates (of both integer and non-integer types). The fourth, fifth, and sixth columns show the rounded values or integers obtained using the *floor*, *ceil*, and *round* functions, respectively.

Table 2: Linear scaling and rounding

$dstCoord$	***equation 2***	***calculated*** $srcCoord$	***floor***	***ceil***	***round***
1	1 x (4/7)	0.57	0	1	1
2	2 x (4/7)	1.14	1	2	1
3	3 x (4/7)	1.71	1	2	2
4	4 x (4/7)	2.28	2	3	2
5	5 x (4/7)	2.85	2	3	3
6	6 x (4/7)	3.42	3	4	3
7	7 x (4/7)	4	4	4	4

Here, depending on the rounding function of choice, it is possible to copy the source image pixel color or gray level from its specific source coordinate to the corresponding pixel coordinate in the destination image (see Figure 2) - by simply matching the source integers with the destination integers and filling in the destination image with the relevant source pixel colors. For example, if $dstCoord = 4$, and $ceil = 3$ (i.e., if *ceil* function is selected), the pixel color to copy to the ‘empty’ destination coordinate 4 will be the pixel color found at the source image strip coordinate 3. Therefore, the color to copy to $dstCoord = 4$ is the blue color (i.e., $4(3) = blue$).

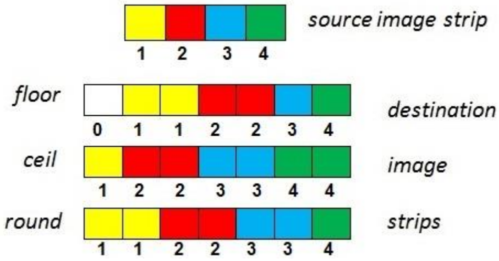

Figure 2: Image strip of length = 4 upscaled to length =7

It is important to note that, here, while using the *floor* function, one destination strip coordinate became invalid since there was no coordinate equal to zero number in the source image (this was only due to MATLAB indexing). However, using the *ceil* function, all destination strip coordinates were valid and matched with their corresponding coordinates in the source image. The same when the round function was used. Again, it is important to note that, in all three cases presented, the gray levels or colors were copied differently due to different rounding methods exhibiting different precisions in this approximation direction. In this context, the author in [33] proposed the analytic method, residual power series, that accomplished the task of approximations directly with less computational complexity without being affected by computation round off errors. Due to all these advantages, the proposed method in [33] will be thoroughly investigated for the extension to other applications in the author’s future works.

4. EXPERIMENTAL EVALUATIONS

a) Evaluation methods:

i. **Image quality assessment (IQA):** The selected full- and no reference IQA metrics are Mean Squared Error (MSE) [5], Structural Similarity Index (SSIM) [29], and Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) [27], Naturalness Image Quality Evaluator (NIQE) [28], respectively. Here, the reason is that, for digital image zooming, where the pursuit for image interpolation quality is the main concern, the metrics that specifically consider the image structures, smallest differences between estimations and realities, and similarity to natural scenes are appropriate. Note that, the lower MSE, BRISQUE, and NIQE scores mean generally the better image quality - and the higher SSIM score means generally the better image quality [18],[19],[20]. For graphical presentation purposes, the SSIM, NIQE, BRISQUE, MSE scores were rescaled to the intervals [0.6, 1.4], [2.5, 3.0], [4.0, 4.6], [5.4 and 6.01], respectively.

ii. **Inferential statistical analysis:** Inferential statistics refers to methods, that rely on probability theory and distributions to predict population values based on sample data [24], which are classified as either parametric or nonparametric [25]. Nonparametric statistics are most used for variables at the nominal or ordinal level of measurement. More information on nonparametric statistics is provided in [25]. Parametric statistics are the most common approach to inferential statistical analysis. Inferential statistical analysis infers properties of a population, for example by testing hypotheses and deriving estimates [24], [34]. In this work, it is used to deduce from a small number of images and draw a conclusion of the behavior of each rounding function on a big number of images [24],[25],[26]. For example, in this work, where there is a great need to infer performance about a big number of images based on a small sample of images - these statistics become a guess. In this way, guessing will contain some degree of potential error. Such an error is called the margin of error (MOE) and it shows how much the difference (between the actual and estimated quantity) probably is, either side of the correct figure [35]. The MOE can be calculated using the following (Eq. 4 or) Eq. 5, where *z* is the z-score associated with a level of confidence (for a confidence level of 95%, *z* = 1.96 as shown in [26], [31]), *p* is the sample proportion percentage, *n* is the sample size, *N* is the population size.

$$MOE = \frac{z * \sqrt{p * (1-p)}}{\sqrt{n}} \quad (4)$$

$$MOE = \frac{z * \sqrt{p * (1-p)}}{\sqrt{\frac{(N-1) * n}{N-n}}} \tag{5}$$

Here, note that the smaller the MEO, the more confidence one may have that the results will be representative of the number of occurrences.

iii. **The number of occurrences**: To quantify how each rounding function outperformed or underperformed among others, in all cases analyzed, the method of tracking the number a given rounding function occurred with the highest scores is introduced. The number tracked and quantified is referred to as the “number of occurrences”. Here, the number of occurrences is directly proportional to the number of test images, the number of IQA metrics, and the number of scaling ratios. The number of occurrences is divided into two categories, namely: the number of targeted occurrences and the number of achieved occurrences. The number of targeted occurrences is the maximum number of times that each rounding function can reach the highest scores. The number of achieved occurrences is the number of times that each rounding function reached the highest scores.

b) **Dataset**:

The dataset, containing 10 test images, was downloaded from the USC-SIPI Image Database [30]. Given that it was necessary to extensively evaluate rounding functions using grayscale images of different sizes, scaling ratios varying from two to five were used. The Microsoft picture manager was used to scale the downloaded 10 test grayscale images to match the following sizes: 128 x 128 to 256 x 256, 170 x 170 to 510 x 510, 128 x 128 to 512 x 512 and 102 x 102 to 510 x 510. Figure 3 shows the caption-labeled 10 test images used in the experimental part. Note that only the IMAGE-9 (5.3.01) was cropped from its original image size of 1024 x 1024 to the size 512 x 512 using MATLAB’s *imcrop* function. Also, MATLAB-r2019 is the main simulation software used to evaluate different rounding functions and generate the results.

c) **Results and discussions**:

Figure 3 shows 10 samples of test images of the size 512 x 512, each. Other sample resized images are not included in this paper. All the results obtained during the rounding function evaluations are graphically presented in Figure 4, Figure 5, Figure 6, Figure 7, and Figure 8. As can be seen, it may seem unclear to readers to know which rounding function achieved the highest scores, mainly, due to the rescaling strategy adopted for graphical representation purposes. Therefore, the number of times the highest scores was achieved by C (i.e., *ceil* function), F (i.e., *floor* function), and R (i.e., *round* function) is presented in Table 3, Table 4, Table 5, Table 6, and Table 7. Referring to these tables and based on the number of test images, IQA metrics, and scaling ratios, each rounding function has a chance of achieving the number of targeted occurrences per image that is equal to 16. In other words, the best rounding function should ideally occur 16 times per image.

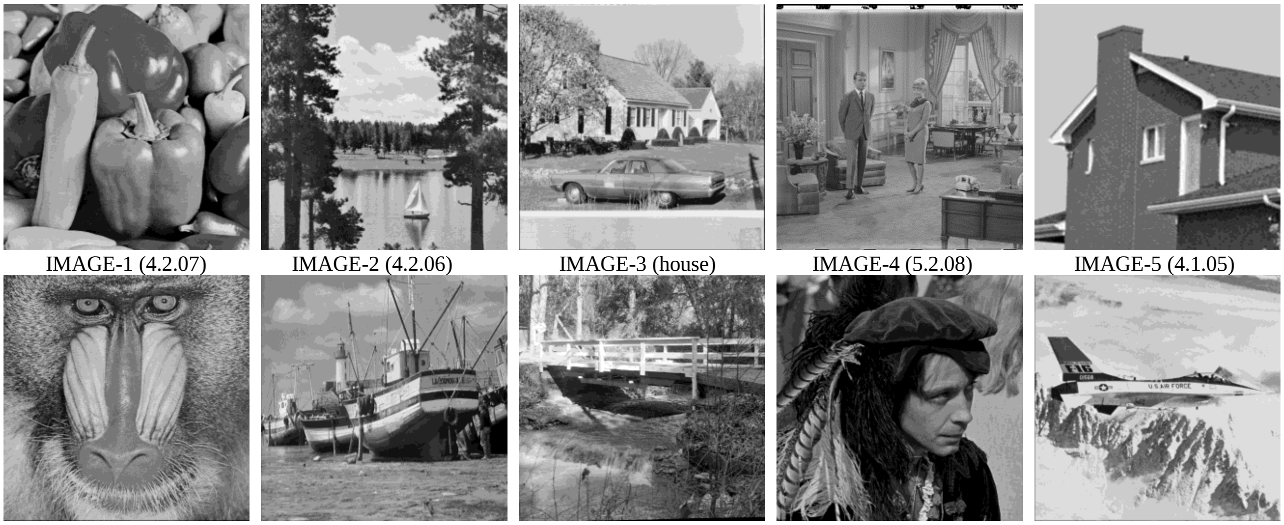

IMAGE-1 (4.2.07) IMAGE-2 (4.2.06) IMAGE-3 (house) IMAGE-4 (5.2.08) IMAGE-5 (4.1.05)

IMAGE-6 (4.2.03) IMAGE-7 (boat.512) IMAGE-8 (5.2.10) IMAGE-9 (5.3.01) IMAGE-10 (4.2.05)

Figure 3: Downloaded sample/test grayscale images

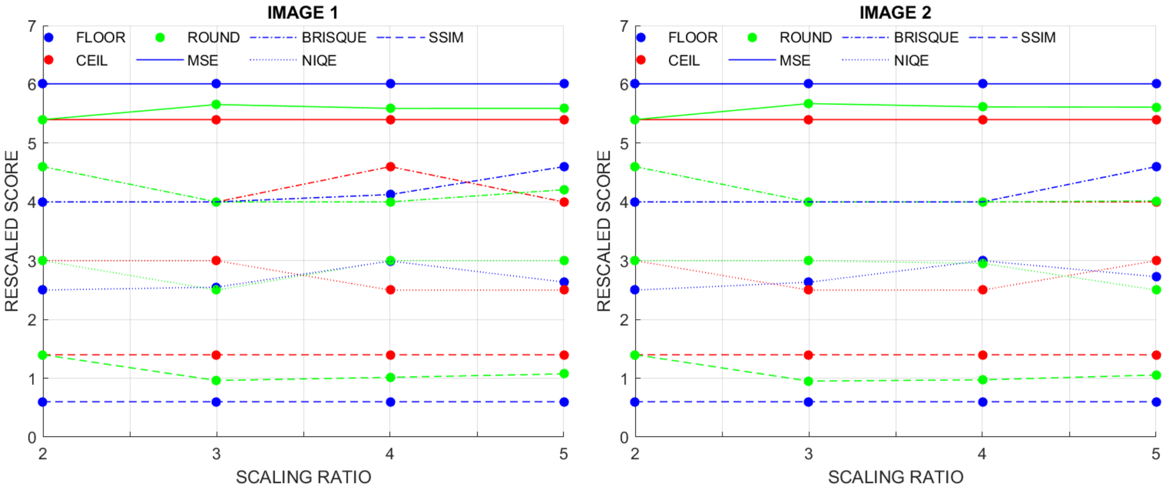

Figure 4: IMAGE-1 (4.2.07) and IMAGE-2 (4.2.06).

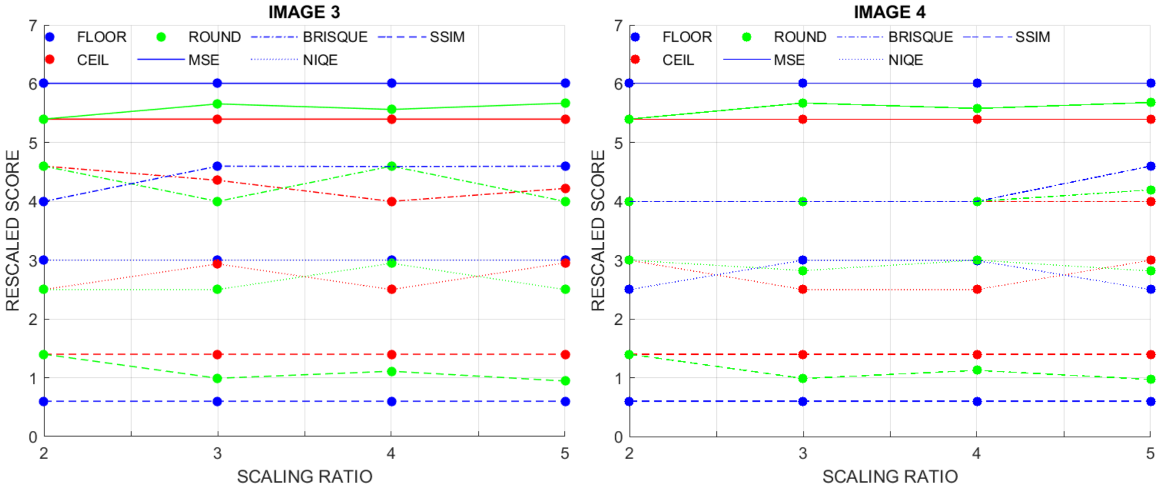

Figure 5: IMAGE-3 (house) and IMAGE-4 (5.2.08).

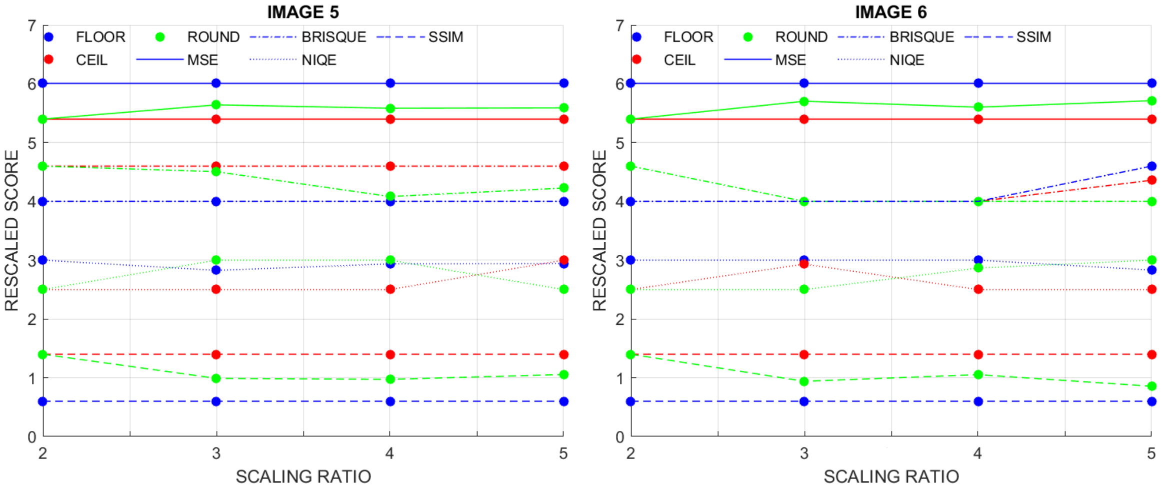

Figure 6: IMAGE-5 (4.1.05) and IMAGE-6 (4.2.03).

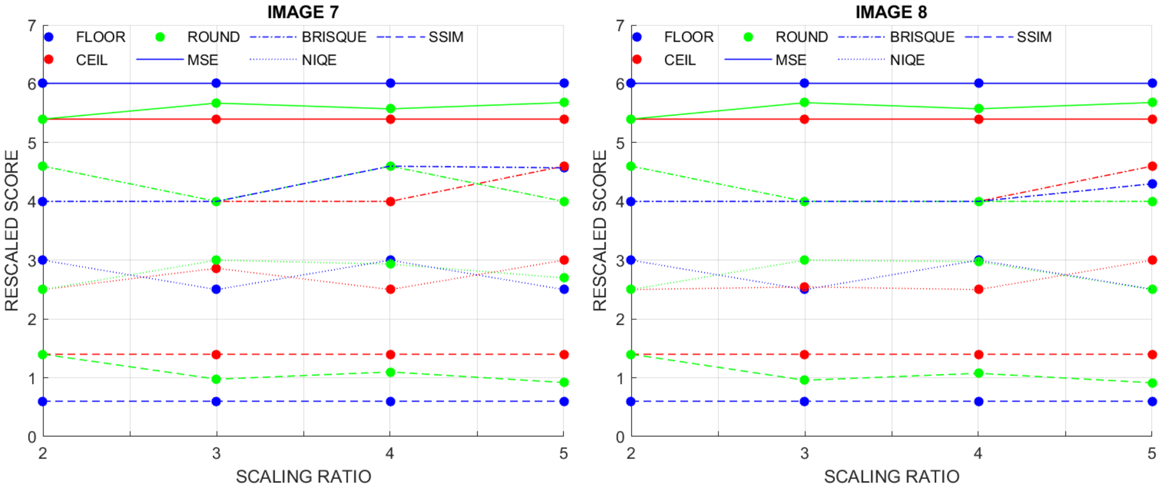

Figure 7: IMAGE-7 (boat.512) and IMAGE-8 (5.2.10).

However, the results presented in Table 3, Table 4, Table 5, Table 6, and Table 7 show that none of the rounding function achieved the maximum number of occurrences per image. Also, all rounding functions did not tie simultaneously in every examined situation. Despite that, the *ceil* function gives the impression of being repeated more times than other rounding functions. In other words, the number of occurrences of the *ceil* function looks to be much higher than that of the other two rounding functions. However, this statement is not enough to draw a conclusion based on observation, because some may raise concerns saying that this cannot be accurate enough to lead to an acceptable (or generalizable) conclusion, especially when a smaller number of sample images was only used. In the effort to alleviate such concerns, the analysis of the number of occurrences based on the number of achieved occurrences over the number of targeted occurrences is presented in Table 8.

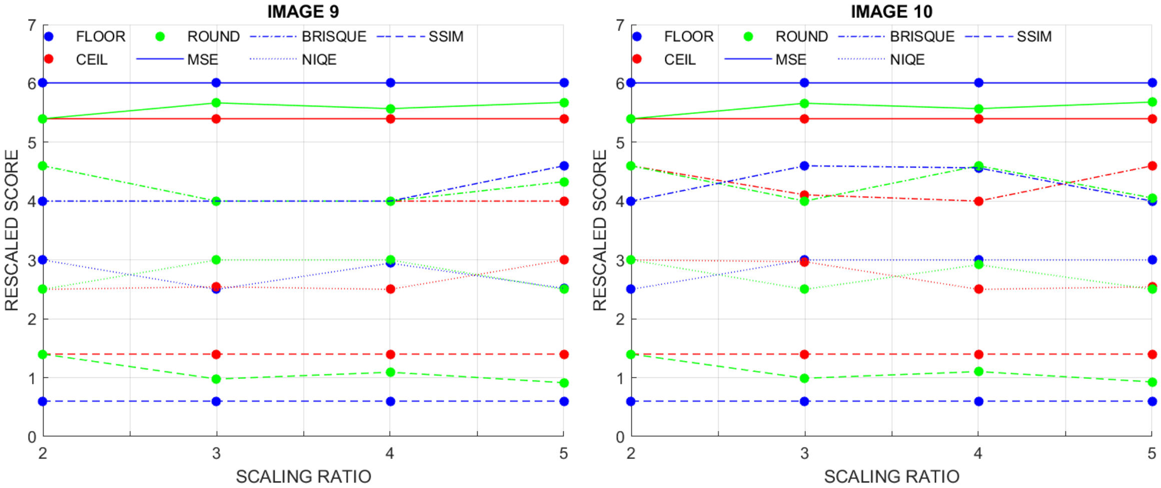

Figure 8: IMAGE-9 (5.3.01) and IMAGE-10 (4.2.05).

Table 3: F = 7 times, C = 27 times, R = 10 times

IMAGE 1 & 2	ratio = 2	ratio = 3	ratio = 4	ratio = 5	ratio = 2	ratio = 3	ratio = 4	ratio = 5
MSE	C & R	C	C	C	C & R	C	C	C
BRISQUE	F & C	F & C & R	R	C	F & C	F & C & R	F & C & R	C
NIQE	F	R	C	C	F	C	C	R
SSIM	C & R	C	C	C	C & R	C	C	C

As mentioned earlier, each rounding function has a chance of achieving 16 targeted occurrences per image. Now, doing the number of occurrence analysis, on all 10 sample test images, each of the three rounding functions had a chance of achieving 160 targeted occurrences. Table 3, Table 4, Table 5, Table 6, and Table 7 show that none of the 3 rounding functions achieved that number of targeted occurrences.

Table 4: F = 6 times, C = 25 times, R = 12 times

IMAGE 3 & 4	ratio = 2	ratio = 3	ratio = 4	ratio = 5	ratio = 2	ratio = 3	ratio = 4	ratio = 5
MSE	C & R	C	C	C	C & R	C	C	C
BRISQUE	F	R	C	R	F & C & R	F & C & R	F & C & R	C
NIQE	C & R	R	C	R	F	C	C	F
SSIM	C & R	C	C	C	C & R	C	C	C

Table 5: F = 7 times, C = 24 times, R = 11 times

IMAGE 5 & 6	ratio = 2	ratio = 3	ratio = 4	ratio = 5	ratio = 2	ratio = 3	ratio = 4	ratio = 5
MSE	C & R	C	C	C	C & R	C	C	C
BRISQUE	F	F	F	F	F	F & C & R	F & C & R	R
NIQE	C & R	C	C	R	C & R	R	C	C
SSIM	C & R	C	C	C	C & R	C	C	C

However, in these 10 images, Table 8 shows that the *ceil*, *round,* and *floor* function respectively achieved 78.75 %, 35%, and 22.5% of the number of targeted occurrences at the MOE = ± 0 %. Here, the MEO = ±0 % because, the population size (N) is equal to the sample size (n) and equal to 160 (i.e., the number of targeted occurrences in all 10 test images presented in Figure 3) - which makes the proportion of percentage (p) equal to 100%. Again, in the effort to alleviate the concern of just drawing a conclusion based on a small number of images, it is still important to statistically produce results showing how each rounding function would have performed if a bigger number of images were used. In this way, let us consider the number of images varying from 50 to 50 000 images and find the percentage of the number of targeted occurrences, the *ceil, round,* and *floor* functions can achieve with the level of confidence equals to 95%.

Table 6: F = 9 times, C = 26 times, R = 12 times

IMAGE 7 & 8	ratio = 2	ratio = 3	ratio = 4	ratio = 5	ratio = 2	ratio = 3	ratio = 4	ratio = 5
MSE	C & R	C	C	C	C & R	C	C	C
BRISQUE	F & C	F & C & R	C	R	F & C	F & C & R	F & C & R	R
NIQE	C & R	F	C	F	C & R	F	C	F & R
SSIM	C & R	C	C	C	C & R	C	C	C

Table 7: F = 7 times, C = 24 times, R = 11 times

IMAGE 9 & 10	ratio = 2	ratio = 3	ratio = 4	ratio = 5	ratio = 2	ratio = 3	ratio = 4	ratio = 5
MSE	C & R	C	C	C	C & R	C	C	C
BRISQUE	F & C	F & C & R	F & C & R	C	F	R	C	F
NIQE	C & R	F	C	R	F	R	C	R
SSIM	C & R	C	C	C	C & R	C	C	C

Table 8: The number of occurrences

Functions	**Number of test images**	**Achieved / Targeted occurrences**	**Achieved occurrences in %**	**MOE**
ceil (C)	10	126/160	78.75%	± 0 %
round (R)	10	56/160	35%	± 0 %
floor (F)	10	36/160	22.5%	± 0 %

Here, the sample size (n) equals 160 (i.e., the total number of targeted occurrences in all 10 test images) for all rounding functions, the population size (N) changes from 800 to 800 000 targeted occurrences (corresponding to 50 to 50 000 images), the confidence level equals to 95 %, z-score equals to 1.96, and the MOE is obtained using Eq.5. Referring to results presented in Table 9, it can be concluded with 95% confidence that the *ceil, round,* and *floor* functions can respectively achieve at least 72.412 %, 27.61 %, and 16.03 % of the number of targeted occurrences in 50 to 50 000 test images – thus confirming that the *ceil* function can achieve better performance than other rounding functions on a big number of images. It is important to note that the sample proportion percentage (p) is calculated using achieved occurrences and targeted occurrences of each rounding function, as shown in Table 8. And, here, the proportion percentage is equal to the percentage of achieved occurrences shown in Table 8.

Table 9: Performance of *ceil*, *round*, and *floor* functions

Number of test images	**Targeted occurrences**	***ceil* ± MOE**	***round* ± MOE**	***floor* ± MOE**
10	160	78.75 % ± 0 %	35% ± 0 %	22.5 % ± 0 %
50	800	78.75 % ± 5.673 %	35% ± 6.615 %	22.5 % ± 5.791 %
500	8,000	78.75 % ± 6.275 %	35% ± 7.317 %	22.5 % ± 6.406 %
5,000	80,000	78.75 % ± 6.332 %	35% ± 7.383 %	22.5 % ± 6.464 %
50,000	800,000	78.75 % ± 6.338 %	35% ± 7.39 %	22.5 % ± 6.47 %

5. CONCLUSION

In this study, the *ceil*, *floor*, and *round* functions were selected and evaluated in nearest neighbor image interpolation. Studying the influence of rounding functions on nearest neighbor image interpolation quality encompasses the contribution and novelty of this paper. Demonstrations and/or descriptions were provided focusing on the linear scaling equation, the rounding functions, the coordinates translation, and pixel replication. The IQA metrics, the number of test images, the number of scaling ratios were used to develop and implement the concept of the number of occurrences. The number of occurrences-based experiments concluded that the *ceil*, *round*, and *floor* functions achieved 78.75 %, 35 %, and 22.5 % of the number of targeted occurrences in 10 test images. Inferential statistics-based experiments concluded with 95% confidence that the ceil, round, and floor functions can respectively achieve at least 72.412 %, 27.61 %, and 16.03 % of the number of targeted occurrences in 50 to 50 000 test images – thus making the ceil function the most appropriate rounding function for nearest-neighbor interpolation purposes.

CONFLICT OF INTEREST

On behalf of all authors, the corresponding author states that there is no conflict of interest.

REFERENCES

[1] Rukundo, O., Cao, H.Q.: Advances on image interpolation based on ant colony algorithm. SpringerPlus 5(1), 1–11 (2016).

[2] Lehmann, T.M, Gonner, C., Spitzer, K.: Survey: Interpolation methods in medical image processing. IEEE Trans. on Medical Imaging 18(11), 1049-1075 (1999).

[3] Rukundo, O., Cao, H.Q.: Nearest neighbour value interpolation. Int. J. of Adv. Com. Sc. and App. 3(4), 25–30 (2012).

[4] Rukundo, O., Cao, H.Q., Huang, M.H.: Optimization of bilinear interpolation based on ant colony algorithm. In Zeng, D.H. (ed.) Int. Conf. of Electrical and Electronics Eng. 2011, LNEE, vol. 137, pp. 571–580. Springer, Heidelberg (2012).

[5] Rukundo, O., Wu, K.N., Cao, H.Q.: Image interpolation based on the pixel value corresponding to the smallest absolute difference. In: 4th International Workshop on Advanced Computational Intelligence, pp. 432–435. IEEE, Wuhan (2011).

[6] Pan, M., Yang, X. and Tang, J.: Research on interpolation methods in medical image processing. Journal of Medical Systems 36(2), 777–807 (2010).

[7] Rukundo, O.: Effects of empty bins on image upscaling in capsule endoscopy. In: 9th International Conference on Digital Image Processing, pp. 104202P-1-104202P-8. SPIE, Hong Kong (2017).

[8] Rukundo, O., Schmidt, S.: Effects of rescaling bilinear interpolant on image interpolation quality. In: Proc. SPIE 10817, Optoelectronic Imaging and Multimedia Technology V, 1081715. SPIE, Beijing (2018).

[9] Rukundo, O., Schmidt, S.: Extrapolation for image interpolation. In: Proc. SPIE 10817, Optoelectronic Imaging and Multimedia Technology V, 108171F. SPIE, Beijing (2018).

[10] Richard S. K., Douglas G.A., et al.: High-resolution and high-magnification endoscopes. Gastrointestinal Endoscopy 69(3), 399–407 (2009).

[11] Li, X., Orchard, M. T.: New edge-directed interpolation. IEEE Trans. on Image Processing 10(10), 1521–1527 (2001).

[12] Tian, Q. C., Wen, H., et al.: A fast edge-directed interpolation algorithm. In: Huang, T.W., Zeng, Z.G., Li, C.D., Lueng, C.S. (eds.) Int. Conf. on Neural Information Processing 2012, LNCS, vol. 7665, pp. 398–405. Springer, Heidelberg (2012).

[13] Zhang, L., Wu, X.: An edge-guided image interpolation algorithm via directional filtering and data fusion. IEEE Trans. on Image Processing 15(8), 2226–2238 (2006).

[14] Ramponi, G.: Warped distance for space-variant linear image interpolation. IEEE Trans. on Im. Proc. 8(5), 629–639 (1999).

[15] Rukundo, O., Maharaj, B.T.: Optimization of image interpolation based on nearest neighbor algorithm. In: 2014 International Conference on Computer Vision Theory and Applications, pp. 641–647. IEEE, Lisbon (2014).

[16] Hale, D.: Image-guided blended neighbor interpolation of scattered data. In: 2009 International Exposition and Annual Meeting, pp. 1127–1131. SEG Library, Houston (2009)

[17] Rukundo, O.: Effects of improved-floor function on the accuracy of bilinear interpolation algorithm. Computer and Information Science 8(4), 1–11, (2015)

[18] Rukundo, O., Schmidt, S.: Aliasing artefact index for image interpolation quality assessment. In: Proc. SPIE 10817, Optoelectronic Imaging and Multimedia Technology V, 108171E. SPIE, Beijing (2018)

[19] Rukundo, O., Pedersen, M., Hovde, Ø.: Advanced Image Enhancement Method for Distant Vessels and Structures in Capsule Endoscopy. Computational and Mathematical Methods in Medicine 2017, 1-13 (2017)

[20] Rukundo, O.: Half-unit weighted bilinear algorithm for image contrast enhancement in capsule endoscopy. In: 9th International Conference on Graphic and Image Processing, pp.106152Q-1-106152Q-9. SPIE, Qingdao (2018)

[21] Rukundo, O.: Non-extra Pixel Interpolation, International Journal of Image and Graphics, 20(4), 2050031, 1-14, (2020)

[22] Maxfield, C., Brown, A.: The definitive guide to how computers do math. John Wiley & Sons, Inc., Hoboken (2005)

[23] Ghoniem, M., Elmoataz, A., Lezoray, O. Discrete infinity harmonic functions: Towards a unified interpolation framework on graphs, In: 18th IEEE International Conference on Image Processing, pp.1361-1364. IEEE, Brussels, (2011)

[24] Vergura, S., Acciani, G., et al.: Descriptive and Inferential Statistics for Supervising and Monitoring the Operation of PV Plants. IEEE Transactions on Industrial Electronics 56(10), 4456-4463, (2009)

[25] Allua, S., Thompson, C.B.: Inferential Statistics. Air Medical Journal 28(4), 168-171, (2009)

[26] Trafimow, D., Confidence intervals, precision and confounding, New Ideas in Psychology, Volume 50, 48-53, (2018)

[27] Mittal, A., A. K. Moorthy, and A. C. Bovik. "Referenceless Image Spatial Quality Evaluation Engine." Presentation at the 45th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, November 2011

[28] Mittal, A., R. Soundararajan, A. C. Bovik.: Making a Completely Blind Image Quality Analyzer, IEEE Signal Processing Letters., 22(3), 209–212, (2013)

[29] Wang, Z., Bovik, A.C., Sheikh, H.R. and Simoncelli, E.P., Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4), pp.600-612, (2004)

[30] The USC-SIPI Image Database, <http://sipi.usc.edu/database/>, accessed 2020-04-20

[31] Margin of Error Calculator, https://goodcalculators.com/margin-of-error-calculator/, accessed 2020-04-20

[32] Linear functions and straight lines, Math On The Web, https://mathonweb.com/help_ebook/html/graphs_2.htm, accessed 2020-01-14

[33] Arqub, O. A., Application of Residual Power Series Method for the Solution of Time-fractional Schrodinger Equations in One-dimensional, Fundamenta Informaticae, 166, 87–110, (2019)

[34] Upton, G., Cook, I.: Oxford Dictionary of Statistics, OUP. ISBN 978-0-19-954145-4, (2008)

[35] Statistics Solutions, Why Do We Have Margin Of Error In Statistics? <https://www.statisticssolutions.com/why-do-we-have-margin-of-error-in-statistics/>, accessed 2021-01-15

[36] Allebach, J.P., Image Scanning, Sampling, and Interpolation, In Handbook of Image and Video Processing (2nd Ed.), (2005)