Image Magnification Using Interval Information

Aranzazu Jurio, Miguel Pagola, Radko Mesiar, Gleb Beliakov, *Senior Member, IEEE*, and Humberto Bustince, *Member, IEEE*

***Abstract*—In this paper, a simple and effective image-magnification algorithm based on intervals is proposed. A low-resolution image is magnified to form a high-resolution image using a block-expanding method. Our proposed method associates each pixel with an interval obtained by a weighted aggregation of the pixels in its neighborhood. From the interval and with a linear K_α operator, we obtain the magnified image. Experimental results show that our algorithm provides a magnified image with better quality (peak signal-to-noise ratio) than several existing methods.**

***Index Terms*—Image magnification, implication operator, interval, K_α operator.**

I. Introduction

IMAGE-MAGNIFICATION methods attempt to increase the spatial resolution of an image without introducing a blur. This process is also known as superresolution [1], [2], image scaling [3], upsampling [4], zooming [5], and image enlargement [6] in related literature.

Image magnification is currently a very active area of research as it is used in many applications. For instance, it can overcome resolution limitations of low-cost imaging systems of personal digital assistants, mobile phones, or surveillance cameras. It is also considered as key medical- and satellite-imaging techniques where the diagnosis or analysis in enhanced resolution images is desired. Additionally, there exists a great demand of video sequence enhancement [7] due to the fact that most web videos are limited by network bandwidth and server storage; thus, it should be stored in low quality.

Previous works on image magnification can be roughly divided into three categories: 1) reconstruction methods; 2) approaches based on machine learning techniques; and 3) interpolation methods.

Manuscript received August 24, 2010; revised January 26, 2011 and April 06, 2011; accepted May 16, 2011. Date of publication May 31, 2011; date of current version October 19, 2011. This paper was supported in part by the National Science Foundation of Spain under Grant TIN2010-15055, by the Research Services of the Universidad Publica de Navarra, by Grant VEGA 1/0080/10, and by Grant P402/11/0378. The Associate Editor coordinating the review of this manuscript and approving it for publication was Prof. A. A. Ross.

A. Jurio, M. Pagola, and H. Bustince are with the Departamento de Automatica y Computacion, Universidad Publica de Navarra, 31006 Pamplona, Spain (e-mail aranzazu.jurio@unavarra.es; bustince@unavarra.es; miguel.pagola@unavarra.es).

R. Mesiar is with the Department of Mathematics and Descriptive Geometry, Slovak University of Technology, 813 68 Bratislava, Slovakia, with the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, 18208 Prague, Czech Republic, and also with the Institute for Research and Application of Fuzzy Modeling, University of Ostrava, 70103 Ostrava, Czech Republic (e-mail: mesiar@math.sk).

G. Beliakov is with the School Information Technology, Deakin University, Burwood 3125, Australia (e-mail: gleb@deakin.edu.au).

Digital Object Identifier 10.1109/TIP.2011.2158227

The reconstruction task is based on prior knowledge about the model that maps the high-resolution image to the low-resolution one. The magnification is then considered as the inverse problem: recovering the original high-resolution image by fusing one or more low-resolution images [8].

In the approaches based on machine learning methods [9]–[11], a set of examples of low resolution and their ideal high-resolution patches are organized in a database. The main idea consists on seeking in the database for similar low-resolution examples, given a patch from the low-resolution input. Their corresponding high-resolution counterparts can then be used for the reconstruction.

The most frequently used methods working with a single image are based on interpolation [12]. Common algorithms, such as nearest neighbor or bilinear interpolation, are computationally simple but suffer from smudge problems, particularly in the edge areas. Directional image interpolations take advantage of the geometric regularity of image structures by performing the interpolation in a chosen direction along which the image is locally regular. Therefore, many adaptive interpolations have been developed with edge detectors [4]. Nevertheless, linear approximations are the most used ones since their computational cost is lower.

Our goal in this paper is to develop a cost-efficient method to construct, starting from a given image, a new image of larger size such that each area or block in the new image is obtained by a weighted aggregation of the intensities of the pixels in the neighborhood of each pixel in the original image. To reach this goal, the method that we propose makes use of the notions of interval and K_α operators [13]. We use intervals because it has been proven in image processing that they allow keeping information about the neighborhood of each pixel [14]. In this way, we employ that information to get the intensities of the new pixels in the magnified image. We also use K_α operators because they allow us to choose, depending on the α parameter, the internal point of the interval that we must select.

To associate an interval to each pixel within the image, we present a new construction method of intervals from a pixel and its neighborhood. We are going to understand the length of this interval as a measure of the variation of intensities in the neighborhood of each pixel.

To see that our method is computationally simple and gets good results, we compare the images obtained by our algorithm and the ones obtained by other methods, both classical and more recent ones.

This paper is organized as follows: First, in Section II, we recall some preliminary definitions. In Section III, we show the method to construct intervals. In Section IV, we present the image-magnification algorithm. We finish with some experimental results in Section V and conclusions in Section VI.

II. Preliminaries

In this section, we recall the main concepts that are necessary for subsequent developments. A strictly decreasing and continuous function $n : [0,1] \to [0,1]$, such that $n(0) = 1$ and $n(1) = 0$, is called strict negation. If, in addition, n is involutive $(n(n(x)) = x, \forall x \in [0,1])$, then we say that it is a strong negation. We call automorphism of the unit interval every function $\varphi : [0,1] \to [0,1]$ that is continuous, strictly increasing, and such that $\varphi(0) = 0$ and $\varphi(1) = 1$.

Let us denote by $L([0,1])$ the set of all closed subintervals in [0, 1], that is

$$L([0,1]) = \{\mathbf{x} = [\underline{x}, \overline{x}] | (\underline{x}, \overline{x}) \in [0,1]^2 \text{and } \underline{x} \leq \overline{x}\}.$$

$L([0,1])$ is a lattice with respect to the relation $\leq_L$, which is defined in the following way, given $\mathbf{x}, \mathbf{y} \in L([0,1])$, we have

$$\mathbf{x} \leq_L \mathbf{y} \text{if and only if} \underline{x} \leq \underline{y} \text{and} \overline{x} \leq \overline{y}.$$

From now on, we denote by $W(\mathbf{x})$ the length (width) of the considered interval; that is, $W(\mathbf{x}) = \overline{x} - \underline{x}$.

Definition 1: Let $\alpha \in [0,1]$. The operator $K_\alpha : L([0,1]) \to [0,1]$ is given by

$$K_\alpha(\mathbf{x}) = \underline{x} + \alpha(\overline{x} - \underline{x})$$

for all $\mathbf{x} \in L([0,1])$.

Clearly, the following properties hold.

a) $K_0(\mathbf{x}) = \underline{x}$ for all $\mathbf{x} \in L([0,1])$.
b) $K_1(\mathbf{x}) = \overline{x}$ for all $\mathbf{x} \in L([0,1])$.
c) $K_\alpha(\mathbf{x}) = K_\alpha([K_0(\mathbf{x}), K_1(\mathbf{x})]) = K_0(\mathbf{x}) + \alpha(K_1(\mathbf{x}) - K_0(\mathbf{x})) = \underline{x} + \alpha \cdot W(\mathbf{x})$ for all $\mathbf{x} \in L([0,1])$.

Notice that operator K_α is a particular case of the Hurwicz aggregation function [15].

A. Implication Operators

Definition 2: An *implication operator* is function $I : [0,1]^2 \to [0,1]$ that satisfies the following properties [16].

$I1$: $x \leq z$ implies $I(x,y) \geq I(z,y)$ for all $y \in [0,1]$.
$I2$: $y \leq t$ implies $I(x,y) \leq I(x,t)$ for all $x \in [0,1]$.
$I3$: $I(0,x) = 1$ for all $x \in [0,1]$ (dominance of falsity).
$I4$: $I(x,1) = 1$ for all $x \in [0,1]$.
$I5$: $I(1,0) = 0$.

Remark: Properties $I3$, $I4$, and $I5$ imply that I is an extension of the standard Boolean implication operators. Indeed, it holds $I(0,0) = I(0,1) = I(1,1) = 1$, and $I(1,0) = 0$. We follow the notation presented in [16], where other properties can be found. In particular, we will use the following ones throughout this paper.

$I6$: $I(1,x) = x$ (neutrality of truth).
$I7$: $I(x, I(y,z)) = I(y, I(x,z))$ (exchange property).
$I9$: $I(x,0) = n(x)$ where n is a strong negation.
$I13$: I is a continuous function (continuity).

III. Construction of Intervals of Fixed Length

In our algorithm, we represent the information of the neighborhood of each pixel by an interval. We use this interval to construct the intensities of the new pixels in the magnified image. In this section, we propose a construction method of the elements of $L([0,1])$ from two points in [0, 1], such that the first number is an internal point of the interval and the second number represents the length of the interval.

The construction method is described in the following theorem.

Theorem 1: Let $\underline{F} : [0,1]^2 \to [0,1]$ be such that

(F1) $\underline{F}(x,0) = x$ for all $x \in [0,1]$;
(F2) $\underline{F}(x,y)$ is increasing in the first variable;
(F3) $\underline{F}(x,y)$ is decreasing in the second variable;
(F4) $\underline{F}(1,x) = 1 - x$ for all $x \in [0,1]$.

Then, $\underline{F}(x,1) = 0$, $\underline{F}(0,x) = 0$, $x \geq \underline{F}(x,y)$ for all $x \in [0,1]$, and mapping

$$F : [0,1]^2 \to L([0,1]) \text{ given by}$$
$$F(x,y) = [\underline{F}(x,y), \underline{F}(x,y) + y]$$

is such that $W(F(x,y)) = y$ for all $x, y \in [0,1]$.

Proof: By construction, $0 \leq \underline{F}(x,y) \leq \underline{F}(x,y) + y$. From (F4), $\underline{F}(1,y) + y = 1$ for all $y \in [0,1]$. By (F2), $\underline{F}(x,y) + y \leq 1$. Thus, F is well defined.

Due to (F1) and (F3), $x = \underline{F}(x,0) \geq \underline{F}(x,y)$. $W(F(x,y)) = y$ by construction. From (F2), $\underline{F}(x,1) \leq \underline{F}(1,1) = 0$ by (F4). As $\underline{F}(0,0) = 0$ and $\underline{F}(0,1) = 0$, $\underline{F}(0,x) = 0$ by (F3). ■

In the following lemma, we relate our operator F that generates intervals with implication operators. This result is really useful because implication operators have been widely studied; thus, we can use several expressions of implication operators to create F functions.

Lemma 1: Let $\underline{F} : [0,1]^2 \to [0,1]$ be such that (F1)–(F4) hold. Then, mapping

$$I : [0,1]^2 \to [0,1] \text{given by}$$
$$I(x,y) = n(\underline{F}(x,y))$$

is an implication operator for any strict negation n.

Proof: $I1$ and $I2$ follow from (F2) and (F3). $I3$ $I(0,x) = n(\underline{F}(0,x)) = n(0) = 1$. $I4$ $I(x,1) = n(\underline{F}(x,1)) = n(0) = 1$. $I5$ $I(1,0) = n(\underline{F}(1,0)) = n(1) = 0$ from (F4). ■

In the following theorem, we present a characterization of the construction method.

Theorem 2: Mapping $\underline{F} : [0,1]^2 \to [0,1]$ satisfies properties (F1)–(F4) if and only if there exists an implication operator satisfying $I6$ with respect to the standard negation and $I9$ such that

$$\overline{F}(x,y) = 1 - I(x,y).$$

Proof: (Sufficiency) I is decreasing in the first variable and increasing in the second, from (F2) and (F3). $I(0,0) = I(0,1) = I(1,1) = 1 - 0$. $I(1,0) = 1 - (1 - 0) = 0$. $I(x,0) = 1 - \underline{F}(x,0) = 1 - x$; thus, $n(x) = 1 - x$. $I(1,x) = 1 - \underline{F}(1,x) = x$. (Necessity) $\underline{F}(x,0) = 1 - I(x,0) = 1 - (1 - x) = x$ from $I9$. (F2) and (F3) from the monotonicity properties of I. $\underline{F}(1,x) = 1 - I(1,x) = 1 - x$ from $I6$. ■

Example 1: In this example, we present some expressions of F functions constructed from different implication operators.

a) Consider Kleene–Dienes implication $I(x,y) = \max(1 - x, y)$ and standard negation. Then, $\underline{F}(x,y) =$

$1 - \max(1-x, y) = \min(x, 1-y)$. In this way, the resulting interval is

$$F(x,y) = [\min(x, 1-y), \min(x, 1-y) + y]. \quad (1)$$

b) Take Reichenbach implication $I(x,y) = 1 - x + xy$ and standard negation. Then, $\underline{F}(x,y) = x - xy = x(1-y)$. Thus, the resulting interval is

$$F(x,y) = [x(1-y), x(1-y) + y]. \quad (2)$$

c) Take Lukasiewicz implication $\min(1, 1-x+y)$ and standard negation. Then, $\underline{F}(x,y) = 1 - \min(1, 1-x+y) = \max(0, x-y)$. Thus, the resulting interval is

$$F(x,y) = [\max(0, x-y), \max(0, x-y) + y]. \quad (3)$$

Remark: From now on, we will use the standard negation as n.

The following theorem is the basis for the image-magnification algorithmic construction that we present in Section IV. It is used to keep the intensity of each pixel from the original image in the new magnified image.

Theorem 3: In the setting of Theorem 2, the following item holds:

$$K_x(F(x,y)) = x \text{ if and only if}$$
$$I(x,y) = 1 - x + xy \text{(Reichenbach's implication)}.$$

Proof: (Necessity) $K_x(F(x,y)) = \underline{F}(x,y) + xy = x$. Thus, $\underline{F}(x,y) = x(1-y)$. (Sufficiency) $K_x([x(1-y), x(1-y) + y]) = x(1-y) + xy = x$. ■

Theorem 4: If $\underline{F}(x,y) = 1 - I(x,y)$ with $I(x,y) = 1 - x + xy$ for all $x, y \in [0,1]$, then $x \in F(x,y)$ for all $x \in [0,1]$.

Proof: If $I(x,y) = 1 - x + xy$, then $F(x,y) = [x(1-y), x(1-y) + y]$. $x \geq x(1-y)$, and $x \leq x - yx + y$. ■

In the next corollary, we build mappings $\underline{F}$ from automorphisms.

Corollary 1: Let $\underline{F} : [0,1]^2 \to [0,1]$ be such that (F1)–(F4) hold. Suppose that the implication operator given by Theorem 2 satisfies $I7$, $I9$, and $I13$. In these conditions, there exists an automorphism in the unit interval such that

$$\underline{F}(x,y) = \varphi^{-1}(\varphi(x) - \varphi(x)\varphi(y)). \quad (4)$$

Proof: Direct from the result, $I(x,y) = n(\varphi^{-1}(\varphi(x) \cdot \varphi(n(y))))$ proved in [16]. ■

Remark: Notice that for (4) to satisfy $(F4)$, it is necessary to take $n(x) = 1 - x$. Therefore, $\underline{F}(x,y) = \varphi^{-1}(\varphi(x)\varphi(1-y))$.

Example 2: If we take $\varphi(x) = x$, then $F(x,y) = [x - xy, x - xy + y]$ (Reichenbach's implication).

IV. Magnification Algorithm

We consider image Q as an $N \times M$ matrix. Each coordinate of the pixels in image Q is denoted by (i,j). The normalized intensity or normalized gray level of the pixel located at (i,j) is represented as q_{ij}, with $0 \leq q_{ij} \leq 1$ for each $(i,j) \in Q$.

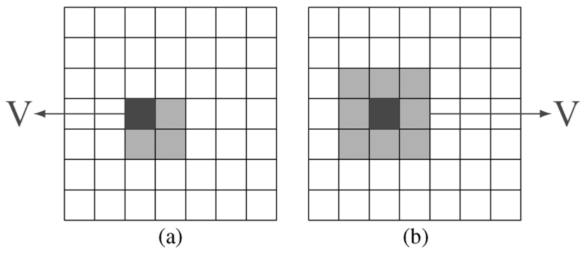


Fig. 1. How to select grid V. For the magnification of the pixel in dark gray, all the pixels in (dark and light) gray belong to grid V. (a) The case where the magnification factor is (2 × 2). (b) The case where the magnification factor is greater than or equal to (3 × 3).

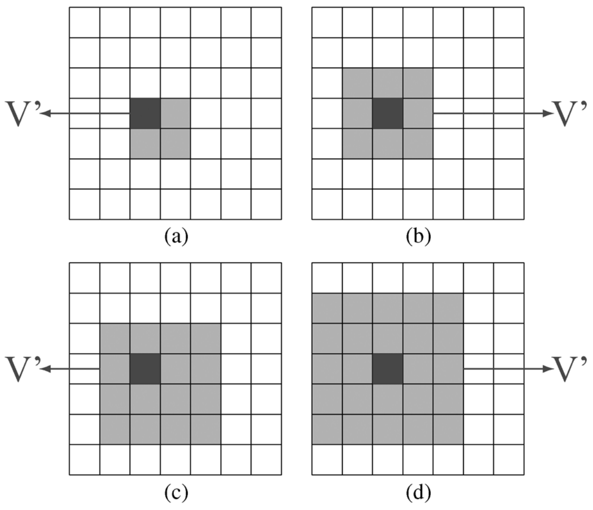


Fig. 2. How to select grid V' for different magnification factors: (a) factor (2 × 2), (b) factor (3 × 3), (c) factor (4 × 4), and (d) factor (5×5).

The purpose of our algorithm is, given image Q of dimension $N \times M$, to magnify it $n \times m$ times; that is, to build a new image of dimension $N' \times M'$ with $N' = n \times N$, $M' = m \times M$, and $n, m \in \mathbb{N}$ with $n \leq N$ and $m \leq M$. We denote $(n \times m)$ as a *magnification factor*.

To do so, we use two different grids over each pixel in the original image, in the following way:

1) Grid V from which we will calculate an interval of intensities. This block V captures the variability of the intensities in the pixel's neighborhood. The size of this window is $p \times q$ with $p, q \in \mathbb{N}$, such that $1 \leq p \leq 3$ and $1 \leq q \leq 3$. The values of p and q depend on the magnification factor (see Fig. 1).
2) Grid V' whose size is $n \times m$ (see Fig. 2). We use this grid and the interval calculated before to build a new block of size $n \times m$ in the magnified image.

The scheme of our algorithm is illustrated in Fig. 3. The image on the left is the original one. We assign to the pixel (i,j) the expanded block V'' of the same size as V' (in this example, $n = m = 3$; hence, V'' is a 3 × 3 block). The image on the right corresponds to the result of the algorithm.

Algorithm 1 presents the method we propose using grids V, V', and V''. In it, we can see that this method is characterized by its simplicity from the implementation point of view.

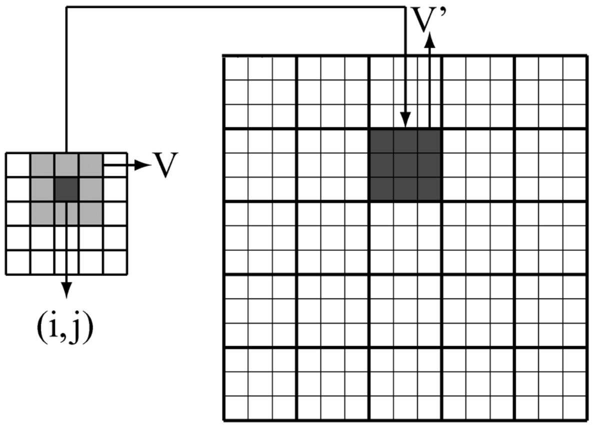


Fig. 3. Magnified image with $n = m = 3$.

Algorithm 1

```
INPUT: $Q$ original image, $(n \times m)$ magnification factor.
  1. Select $p, q \in \{1, 2, 3\}$.
  2. FOR each pixel $(i, j)$ DO
    2.1 Fix a grid $V$ of dimension $p \times q$ centered at $(i, j)$.
    2.2 Calculate $W$ as the difference between the largest and the smallest intensities of the pixels in $V$.
    2.3 Calculate $\delta \cdot W \in [0, 1]$, where $\delta = 1 - 2 \cdot \sigma$, being $\sigma$ the standard deviation of the intensities of $V$.
    2.4 Build the interval $F(q_{ij}, \delta \cdot W)$ using (2).
    2.5 Fix a grid $V'$ of dimension $n \times m$ centered at $(i, j)$.
    2.6 Build a new empty block $V''$ of size $n \times m$.
    2.7 FOR each element $(k, l)$ of $V''$ DO
          $q''_{kl} := K_{q'_{kl}} \left( F(q_{ij}, \delta \cdot W) \right).$
        ENDFOR
  2.8 Put the block $V''$ in the magnified image.
    ENDFOR
```

Next, we explain the steps of Algorithm 1 by means of an example. Given an image in Fig. 4 of dimension 5 × 5, we want to build a magnified image of dimension 15 × 15 (magnification factor $= (3 \times 3)$).

A. Step 1: Select $p, q \in \{1, 2, 3\}$

These parameters represent the size of grid V. Their values (p for rows and q for columns) are selected depending on the following magnification factor:

$$\begin{cases} \text{if } n = 1 \text{ then } p = 1 \\ \text{else if } n = 2 \text{ then } p = 2 \\ \text{else } p = 3 \end{cases}$$

0.70	0.60	0.65	0.40	0.41
0.66	0.59	0.60	0.50	0.46
0.62	0.70	0.60	0.52	0.48
0.60	0.68	0.55	0.53	0.50
0.63	0.62	0.53	0.50	0.50

Fig. 4. Example: Original image.

0.70	0.60	0.65	0.40	0.41
0.66	0.59	0.60	0.50	0.46
0.62	0.70	0.60	0.52	0.48
0.60	0.68	0.55	0.53	0.50
0.63	0.62	0.53	0.50	0.50

Fig. 5. Example: Original image.

which is analogous for m and q. We just use the set of values $\{1, 2, 3\}$ because the greater the size of the grid V, the greater the neighborhood of each pixel that we consider, and we do not want that remote pixels interfere in the magnification.

In the example, we want to magnify the image by factor (3 × 3); therefore, $p = q = 3$.

B. Step 2.1: Fix Grid V of Dimension $p \times q$ Centered at Each Pixel

This grid V represents the neighborhood that is used to build the interval. In the example, for pixel (2, 3) (marked in dark gray in Fig. 5), we fix a grid of dimension 3×3 $(p \times q)$ around it (in light gray).

Remark: For pixels in the first or the last row/column, we choose a grid centered at them, as shown in Fig. 6.

C. Step 2.2: Calculate W as the Difference Between the Largest and the Smallest of the Intensities of the Pixels in v

W is the maximum length that the interval we are going to build can have, i.e.,

$$W = \max_{q_{kl} \in V} \{q_{kl}\} - \min_{q_{kl} \in V} \{q_{kl}\}. \tag{5}$$

In the example, for pixel (2, 3), we calculate W as

$$\begin{aligned} W =& \max(0.6, 0.65, 0.4, 0.59, 0.6, 0.5, 0.7, 0.6, 0.52) \\ &- \min(0.6, 0.65, 0.4, 0.59, 0.6, 0.5, 0.7, 0.6, 0.52) \\ =& 0.7 - 0.4 = 0.3. \end{aligned}$$

D. Step 2.3: Calculate $\delta \cdot W$

$\delta \cdot W$ represents the length of the interval that we build. Our first idea was to use W as the length of the interval (instead of $\delta \cdot W$) in our construction. However, that proposal has some

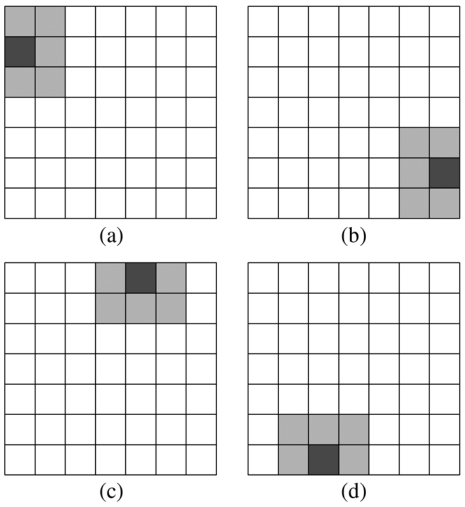
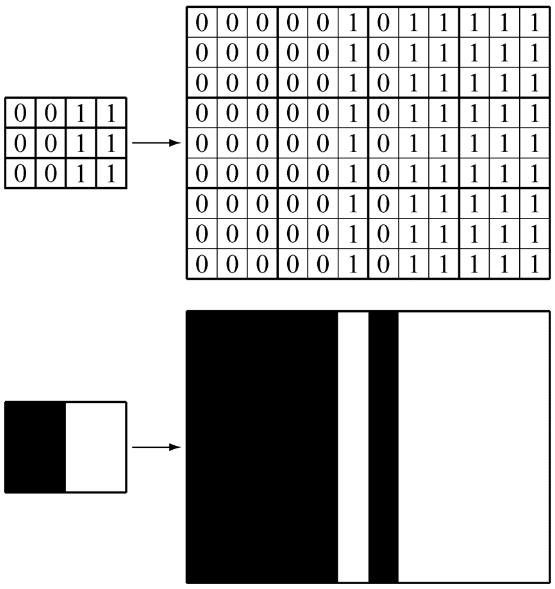


Fig. 6. Example: Grid for pixels in the first or last row/column. (a) Pixel in the first column. (b) Pixel in the last column. (c) Pixel in the first row. (d) Pixel in the last row.



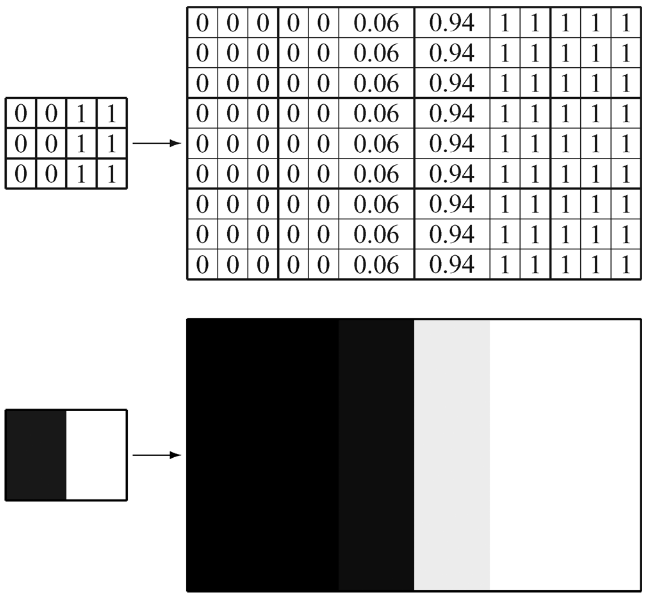
0	0	1	1
0	0	1	1
0	0	1	1

0	0	0	0	0	1	0	1	1	1	1	1
0	0	0	0	0	1	0	1	1	1	1	1
0	0	0	0	0	1	0	1	1	1	1	1
0	0	0	0	0	1	0	1	1	1	1	1
0	0	0	0	0	1	0	1	1	1	1	1
0	0	0	0	0	1	0	1	1	1	1	1
0	0	0	0	0	1	0	1	1	1	1	1
0	0	0	0	0	1	0	1	1	1	1	1
0	0	0	0	0	1	0	1	1	1	1	1

Fig. 7. Problem with the edges in binary images if the amplitude is W.

problems, as it is shown in the example in Fig. 7. In the upper part, it is the numerical calculation and below is its representation in the image, where 0 means black and 1 means white.

As we can see, the edge in the magnified image is not correct. To solve this problem, the length of the interval is proportionally reduced to parameter δ, and in this way, the jump from white to black is gradual. To calculate the δ value, we use the standard deviation of the intensities in grid V. We use the expression $\delta = 1 - 2 \cdot standard\ deviation$ because the maximum standard deviation in grid V is 0.5, and we want the δ value to vary in [0, 1]. In this sense, if all the pixels in V have the same intensity, the amplitude of the interval is W ($\delta = 1$); nevertheless, when there is a big difference between the intensities (presence of an edge), the value of δ decreases, and therefore, the amplitude also decreases. Fig. 8 shows the effect of parameter δ.



0	0	1	1
0	0	1	1
0	0	1	1

0	0	0	0	0	0.06	0.94	1	1	1	1	1
0	0	0	0	0	0.06	0.94	1	1	1	1	1
0	0	0	0	0	0.06	0.94	1	1	1	1	1
0	0	0	0	0	0.06	0.94	1	1	1	1	1
0	0	0	0	0	0.06	0.94	1	1	1	1	1
0	0	0	0	0	0.06	0.94	1	1	1	1	1
0	0	0	0	0	0.06	0.94	1	1	1	1	1
0	0	0	0	0	0.06	0.94	1	1	1	1	1
0	0	0	0	0	0.06	0.94	1	1	1	1	1

Fig. 8. Solution to the problem in Fig. 7 using δ parameter.

	$q'_{1,1}$=0.70	$q'_{1,2}$=0.65	$q'_{1,3}$=0.40
V'=	$q'_{2,1}$=0.59	$q'_{2,2}$=0.60	$q'_{2,3}$=0.50
	$q'_{3,1}$=0.70	$q'_{3,2}$=0.60	$q'_{3,3}$=0.52

Fig. 9. Original V' block for pixel (2, 3).

Following the example in Fig. 4, the standard deviation of grid V is 0.0834; thus, $\delta = 1 - 2 * deviation = 0.8332$. Then, the length for the interval associated to pixel (2, 3) is $\delta \cdot W = 0.8332 \cdot 0.3 = 0.25$.

E. Step 2.4: Build Interval $F(q_{ij}, \delta \cdot W)$

We associate to each pixel an interval of length $\delta \cdot W$. To do so, we use the method explained in Section III. In this case, we take the expression in (2) based on Reichenbach's implication. The resulting interval is

$$F(q_{ij}, \delta \cdot W) = [q_{ij}(1 - \delta \cdot W), q_{ij}(1 - \delta \cdot W) + \delta \cdot W].$$

In the example, the interval associated to pixel (2, 3) is given by

$$F(0.6, 0.25) = [0.6(1–0.25), 0.6(1–0.25) + 0.25] = [0.45, 0.70].$$

F. Step 2.5: Fix Grid V' of Dimension $n \times m$ Centered at (i, j)

In the example, this new block is shown in Fig. 9.

G. Step 2.6: Build a New Empty Block V'' of Size $n \times m$

Following the example, the size of this block is 3 × 3 (see Fig. 10).

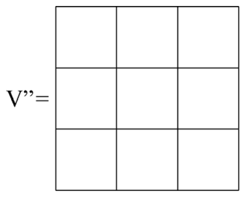


Fig. 10. Empty block V''.

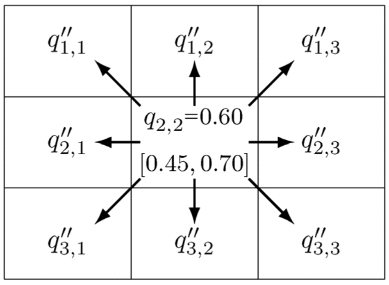


Fig. 11. Expanded block for pixel q_{23}.

0.60	0.61	0.65
0.60	0.60	0.58
0.63	0.60	0.58

Fig. 12. Numerical expanded block for pixel q_{23}.

H. Step 2.7: Calculate $K_{q_{kl}}(F(q_{ij}, \delta \cdot W))$ for Each Pixel

Next, we expand pixel (i, j) in image Q over the new block V''. In the example, pixel (2, 3) is expanded, as shown in Fig. 11.

To keep the value of the original pixel at the center of the new block, Theorem 3 states that α should be equal to the intensity of that pixel. In the case of pixel (2, 3), we have

$$0.6 = q''_{22} = K_{q'_{22}}([0.45, 0.7]) = 0.45 + q'_{22}0.25 = 0.6.$$

We apply this method to fill in all the other pixels in the block. In this way, from Proposition 3, we take as α for each pixel the value of that pixel in the grid V'.

- $\alpha = q'_{11}$. Then, $q''_{11} = 0.45 + q'_{11}0.25 = 0.45 + 0.6 \cdot 0.25 = 0.6$.
- $\alpha = q'_{12}$. Then, $q''_{12} = 0.45 + q'_{12}0.25 = 0.45 + 0.65 \cdot 0.25 = 0.6125$.
- $\ldots$
- $\alpha = q'_{33}$. Then, $q'_{33} = 0.42 + q'_{33}0.3 = 0.45 + 0.52 \cdot 0.25 = 0.58$.

In Fig. 12, we show the expanded block for pixel (2, 3) in the example.

Once each of the pixels has been expanded, we join all the blocks (Step 2.8) to create the magnified image. This process is shown in Fig. 3.

V. Experimental Results

To quantitatively estimate the quality of magnification, we take a set of gray-scale images of size 512 × 512 and reduce them to several sizes (see Section V-A). Then, we enlarge the previously reduced images back to 512 × 512 and compare them to the images from which we started. Our expectation is that the enlarged images will be similar to the original ones, and we can estimate the similarity quantitatively by the peak signal-to-noise ratio (PSNR).

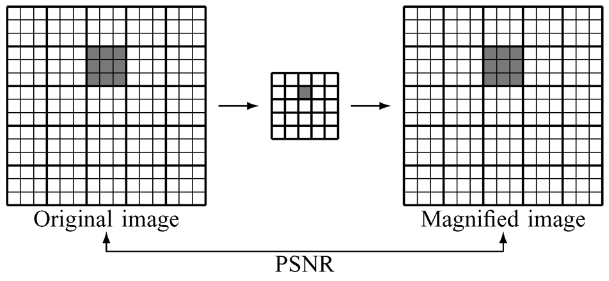


Fig. 13. Schema to check the algorithm.

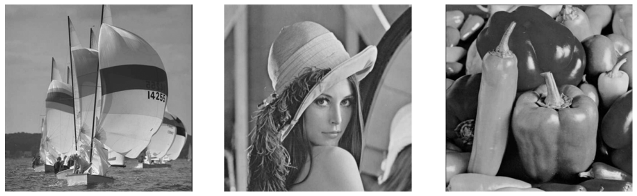


Fig. 14. Images used in the different factors experiment. (From left to right) Ship, Lena, and Peppers.

TABLE I
PSNRs (in Decibels) Over Images in Fig. 14. (From Left to Right) Images Magnified by Factors (1 × 2), (2 × 1), (2 × 2), (3 × 3), and (4 × 4)

	(1 × 2)	(2 × 1)	(2 × 2)	(3 × 3)	(4 × 4)
Ship	31.0370	33.3127	28.7784	26.6363	24.0692
Lena	33.8153	36.8167	31.6368	29.0135	26.2505
Peppers	33.2101	34.1512	30.3555	28.0643	25.4590

In Fig. 13, we see a schema of the procedure when we reduce the image to obtain a 9 × smaller one and then enlarge the result by the magnification factor (3 × 3).

A. Reduction Algorithm

To reduce the images, we use the algorithm proposed in [17]. Starting from the images of dimension $N \times M$, the scheme of the algorithm is the following.

1) Divide image Q in blocks of size $l \times s$. If N is not a multiple of l or M of s, we delete the minimum number of rows/columns in the boundary of the image until the new size of the image satisfies the property.
2) Associate each block with an interval in the following way: The lower bound of the interval is given by the minimum of the intensities in the block and the upper bound, by the maximum.
3) Associate each interval with the middle point of the interval.

We select this reduction algorithm because it takes into account all the pixels of the block, instead of other reduction methods such as subsampling (take one of every n pixels).

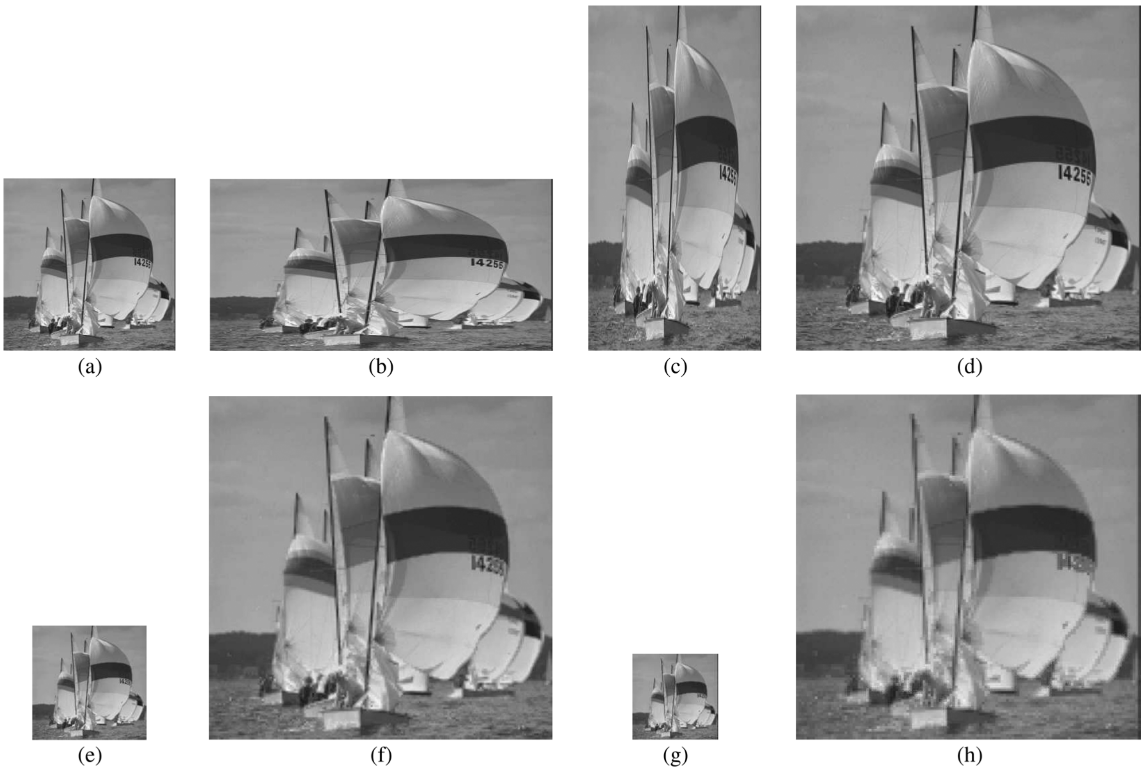


Fig. 15. (a) Reduced image Ship of size 256 × 256 and its magnification by (b) factor (1 × 2) of size 256 × 512, (c) factor (2 × 1) of size 512 × 256, and (d) factor (2 × 2) of size 512 × 512. (e) Reduced image Ship of size 170 × 170 and (f) its magnification by factor (3 × 3) of size 510 × 510. (g) Reduced image Ship of size 128 × 128 and (h) its magnification by factor (4 × 4) of size 512 × 512.

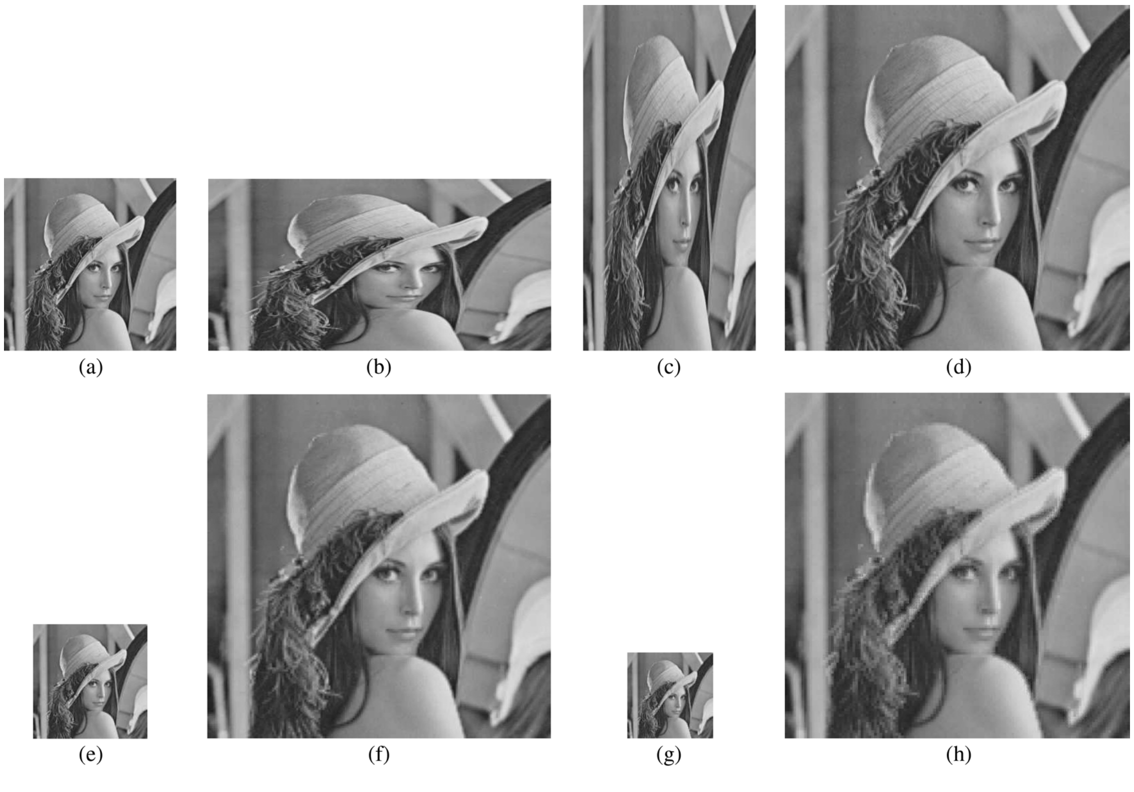


Fig. 16. (a) Reduced image Lena of size 256 × 256 and its magnification by (b) factor (1 × 2) of size 256 × 512, (c) factor (2 × 1) of size 512 × 256, and (d) factor (2 × 2) of size 512 × 512. (e) Reduced image Lena of size 170 × 170 and (f) its magnification by factor (3 × 3) of size 510 × 510. (g) Reduced image Lena of size 128 × 128 and (h) its magnification by factor (4 × 4) of size 512 × 512.

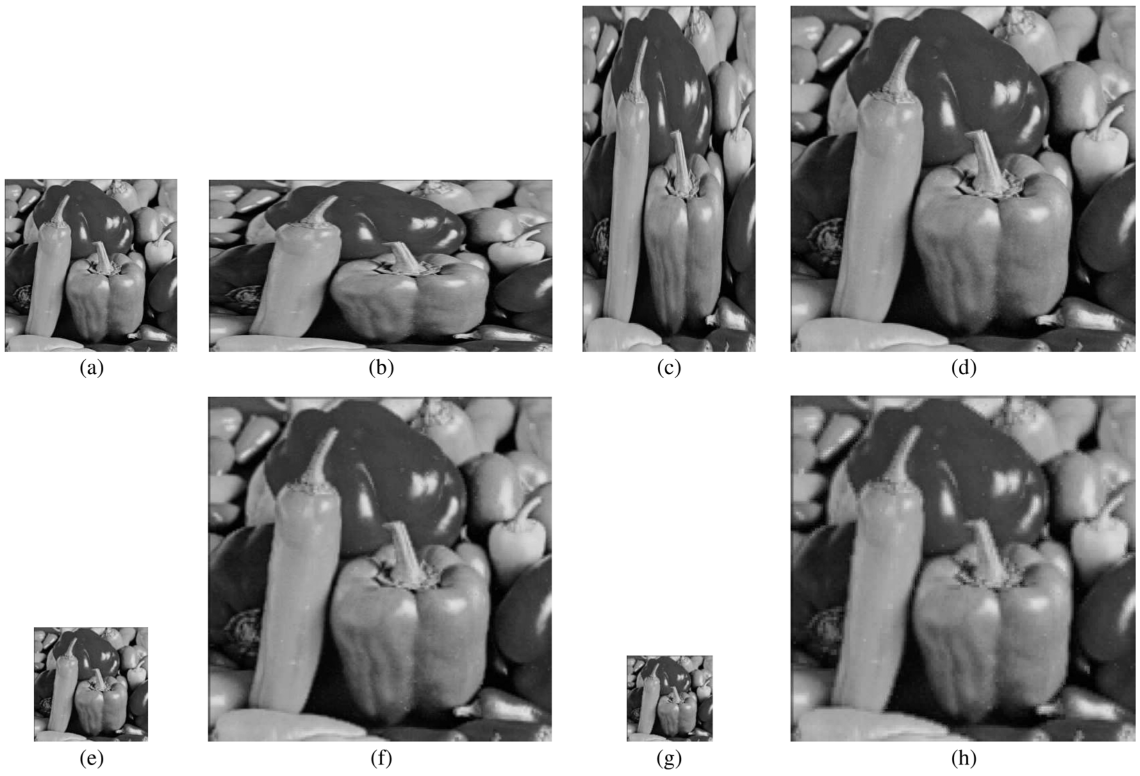


Fig. 17. (a) Reduced image Peppers of size 256 × 256 and its magnification by (b) factor (1 × 2) of size 256 × 512, (c) factor (2 × 1) of size 512 × 256 and (d) factor (2 × 2) of size 512 × 512. (e) Reduced image Peppers of size 170 × 170 and (f) its magnification by factor (3 × 3) of size 510 × 510. (g) Reduced image Peppers of size 128 × 128 and (h) its magnification by factor (4 × 4) of size 512 × 512.

TABLE II
PSNRs (IN DECIBELS) OVER IMAGES SHIP, LENA, AND PEPPERS AND THE AVERAGE PSNR OVER 96 IMAGES, WHEN THE MAGNIFICATION FACTOR IS (2 × 2)

	NN	$Bilin.$	$Bicub.$	$Spline$	SME	KR	SP	$Alg.1$
Ship	23.9771	27.7511	28.0477	28.1508	28.2358	27.4305	31.1254	28.7754
Lena	25.9197	30.2295	30.5104	30.5858	30.5807	30.3803	34.9058	31.6368
Peppers	27.7726	29.5300	29.7810	29.8471	29.8962	29.5755	32.4561	30.3555
Avg. 96 images	22.6663	25.9656	26.1462	26.1751	26.1757	25.9218	28.3549	27.0354

TABLE III
PSNRs (IN DECIBELS) OVER IMAGES SHIP, LENA, AND PEPPERS AND THE AVERAGE PSNR OVER 96 IMAGES, WHEN THE MAGNIFICATION FACTOR IS (3 × 3)

	NN	$Bilin.$	$Bicub.$	$Spline$	KR	$Alg.1$
Ship	24.1994	24.9149	25.0071	25.0300	25.0135	26.6363
Lena	25.7412	26.6057	26.7249	26.7390	26.7191	29.0135
Peppers	25.1190	26.1295	26.2840	26.3278	26.1756	28.0643
Avg. 96 images	22.5275	23.2897	23.3299	23.3219	23.3017	24.7205

TABLE IV
AVERAGE TIME (IN SECONDS) FOR THE MAGNIFICATION OF ONE IMAGE FOR MAGNIFICATION FACTORS (2 × 2) AND (3 × 3)

	NN	$Bilin.$	$Bicub.$	$Spline$	SME	KR	SP	$Alg.1$
Factor (2 × 2)	0.0333	0.07877	0.2254	0.1296	337.0709	19.1854	692.9004	1.6204
Factor (3 × 3)	0.0332	0.0756	0.2211	0.0917		17.2334		0.7207

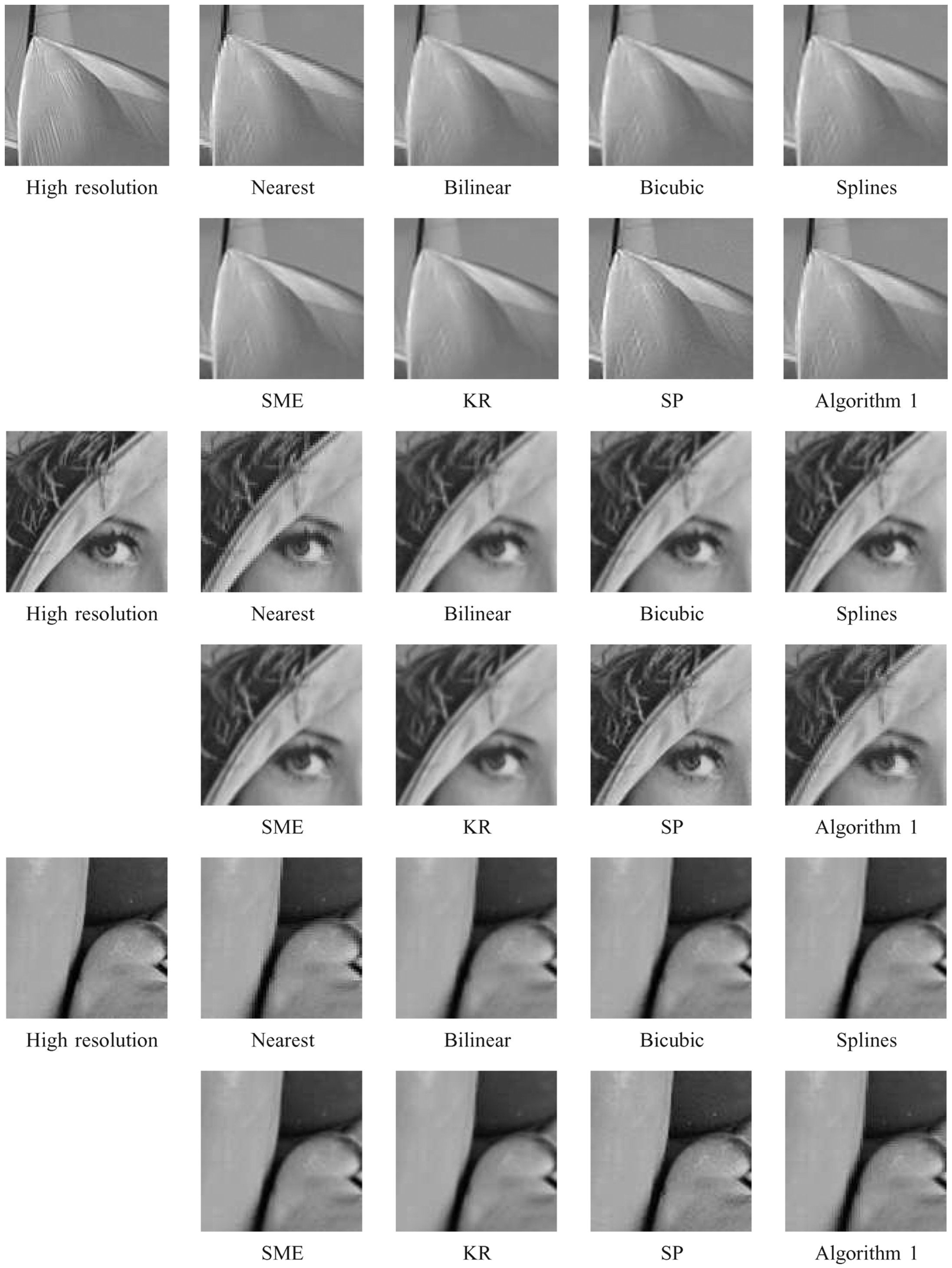


Fig. 18. Cropped images from Ship, Lena, and Peppers. (From top to bottom and from left to right) High-resolution image, magnification by a factor of 2 with nearest neighbor, bilinear, bicubic, and splines interpolations; SME; KR; and Algorithm 1.

B. Results Obtained for Different Magnification Factors in Algorithm 1

In this experiment, we compare different solutions obtained when the magnification factor in Algorithm 1 varies. We take three images from the database [18] (see Fig. 14). In Figs. 15 –17, we show the magnified images using the magnification factors (1×2), (2×1), (2×2), (3×3), and (4×4), as well as the reduced images from which we start in each case.

In Table I, we present the PSNR obtained by each one of the images. From this table, we see that the obtained results are very good. Evidently, as it happens with all the magnification methods, when the magnification factor is increased, the result loses quality.

C. Comparison With Other Methods

In this section, we compare our image-magnification algorithm with other methods that can be found in the literature.

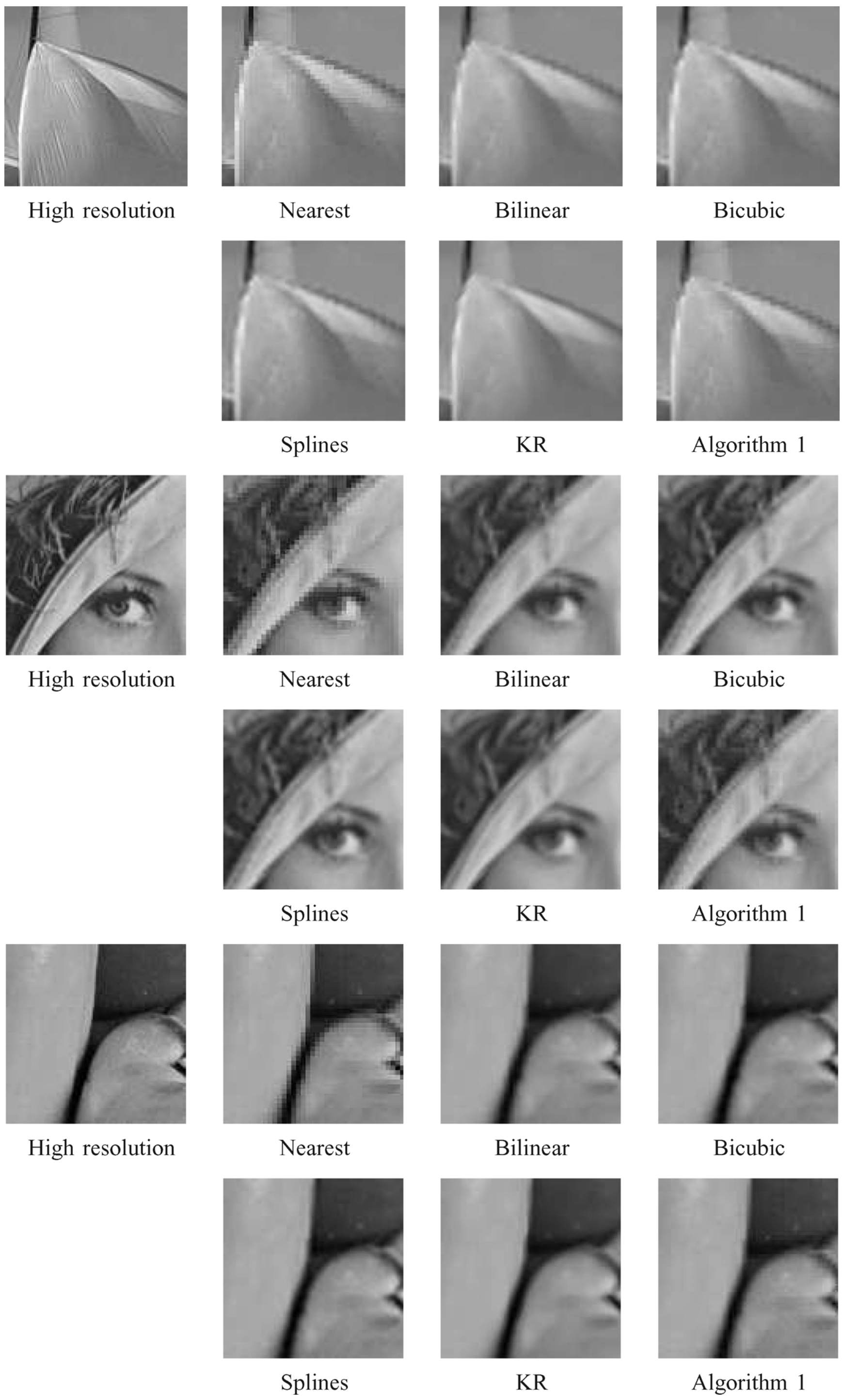


Fig. 19. Cropped images from Ship, Lena, and Peppers. (From top to bottom and from left to right) High-resolution image, magnification by a factor of 3 with nearest neighbor, bilinear, bicubic, and splines interpolations; KR; and Algorithm 1.

Specifically, we use four classic interpolations [12]: 1) nearest neighbor; 2) bilinear; 3) bicubic; and 4) splines interpolations, as well as three recent superresolution algorithms: 1) sparse mixing estimation (SME) [8]; 2) kernel regression (KR) [19]; and 3) sparse representation (SP) [11]. All experiments are performed with softwares provided by the authors (SME, KR, and SP) and MATLAB implementation of different interpolations.[1]

[1]This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. This includes the MATLAB implementation of Algorithm 1, three example images, and a read-me file. This material is 104 kB in size.

In the experiment, we work with 96 images from the database given in [18]. The original dimension of all the images is 512 $\times$ 512. We reduce them with the reduction algorithm, as explained in Section V-A, to dimensions 256 $\times$ 256 (using block size 2 $\times$ 2) and 170 $\times$ 170 (using block size 3 $\times$ 3). We magnify them with the seven mentioned methods, as well as our method. We calculate the similarity between each of the resulting images and the original images by means of PSNR.

Next, we show the different magnifications of the cropped images Ship, Lena, and Peppers. In Fig. 18, the starting point is a reduced image of dimension 256×256, and the magnification process has a factor of (2×2). In Fig. 19, the starting point is a reduced image of dimension 170×170, and the magnification factor is 3×3. In this case, we do not show the SME and SP algorithms because these methods can magnify images by a factor of (2×2).

We visually check that the SME, KR, and SP methods get the results with less jaggy artifacts. However, if we study the PSNR obtained by each method, we check that the quality of images magnified with Algorithm 1 is better than the quality obtained by the rest of the implemented methods, except with the SP method (see Tables II and III). This happens with the three shown images and with the average of the 96 images of the experiment. In this way, we argue that our proposal works very well, and therefore achieves very good results, in all the nonedge areas, i.e., areas where the changes of intensities are not very big.

We have already said that simplicity is one of the strongest points of Algorithm 1. This fact has an effect on the execution time of the algorithm. All the experimentation in this paper have been executed with a processor Intel Core 2 Duo 6750 at 2.66 GHz with 3 GB of random access memory. The software for the SME and KR methods has been provided by their authors, and the interpolation methods have been implemented by the function *interp2* of MATLAB, over the version MATLAB 7.8.0 (R2009a). In Table IV, the mean time for the magnification of one image with each one of the implemented methods (for magnification factors 2×2 and 3×3) is shown. In this sense, the fastest algorithms are the four classic interpolations, whose time is always smaller than 0.3 s. However, comparing with the more recent methods, Algorithm 1 is between 11 and 23 times faster than KR, 208 times faster than SME, and 427 times faster than SP.

VI. Conclusion and Future Research

In this paper, we have presented an image-magnification algorithm that uses intervals and K_α operators. In this method, a new block is constructed for every pixel of the image, and the central pixel of that block maintains the intensity of the original pixel. To fill in the rest of the pixels, we use the relation between the pixel in the original image and its neighbors.

In this sense, the algorithm shows that intervals are a good representation of the effect of the neighborhood over an element. Due to this, our algorithm provides better results than some of the most commonly used methods as we have confirmed with the PSNR values of the experiments. We have also proven the computational simplicity of our algorithm.

From the results we have obtained, we plan as future research the study of new methods to calculate the δ parameter and the use of filters to improve our results in edge zones.

Acknowledgment

The authors would like to thank the editor and the anonymous reviewers for their valuable suggestions that have greatly improved this paper.

References

[1] S. Park, M. Park, and M. Kang, "Super-resolution image reconstruction: A technical overview," *IEEE Signal Process. Mag.*, vol. 20, no. 3, pp. 21–36, May 2003.
[2] K. I. Kim and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 6, pp. 1127–1133, Jun. 2010.
[3] C. Kim, S. Seong, J. Lee, and L. Kim, "Winscale: An image-scaling algorithm using an area pixel model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 6, pp. 549–553, Jun. 2003.
[4] Y. J. Lee and J. Yoon, "Nonlinear image upsampling method based on radial basis function interpolation," *IEEE Trans. Image Process.*, vol. 19, no. 10, pp. 2682–2692, Oct. 2010.
[5] C. Arcelli, N. Brancati, M. Frucci, G. Ramella, and G. S. di Baja, "A fully automatic one-scan adaptive zooming algorithm for color images," *Signal Process.*, vol. 91, no. 1, pp. 61–71, Jan. 2011.
[6] M. Unser, A. Aldroubi, and M. Eden, "Enlargement or reduction of digital images with minimum loss of information," *IEEE Trans. Image Process.*, vol. 4, no. 3, pp. 247–258, Mar. 1995.
[7] Z. Xiong, X. Sun, and F. Wu, "Robust web image/video super-resolution," *IEEE Trans. Image Process.*, vol. 19, no. 8, pp. 2017–2028, Aug. 2010.
[8] S. Mallat and G. Yu, "Super-resolution with sparse mixing estimators," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2889–2900, Nov. 2010.
[9] P. P. Gajjar and M. V. Joshi, "New learning based super-resolution: Use of DWT and IGMRF prior," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1201–1213, May 2010.
[10] K. S. Ni and T. Q. Nguyen, "Image superresolution using support vector regression," *IEEE Trans. Image Process.*, vol. 16, no. 6, pp. 1596–1610, Jun. 2007.
[11] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution Via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.
[12] A. Amanatiadis and I. Andreadis, "A survey on evaluation methods for image interpolation," *Meas. Sci. Technol.*, vol. 20, no. 10, pp. 104 015–104 023, Oct. 2009.
[13] H. Bustince, T. Calvo, B. De Baets, J. Fodor, R. Mesiar, J. Montero, D. Paternain, and A. Pradera, "A class of aggregation functions encompassing two-dimensional OWA operators," *Inf. Sci.*, vol. 180, no. 10, Sp. Iss. SI, pp. 1977–1989, May 15, 2010.
[14] H. Bustince, E. Barrenechea, J. Fernandez, M. Pagola, J. Montero, and C. Guerra, "Contrast of a fuzzy relation," *Inf. Sci.*, vol. 180, no. 8, pp. 1326–1344, Apr. 15, 2010.
[15] L. Hurwicz, Optimality Criteria for Decision Making Under Ignorance Cowles Communication Discussion Paper, Statistics No. 370, 1951.
[16] H. Bustince, P. Burillo, and F. Soria, "Automorphisms, negations and implication operators," *Fuzzy Sets Syst.*, vol. 134, no. 2, pp. 209–229, Mar. 1, 2003.
[17] H. Bustince, D. Paternain, B. De Baets, T. Calvo, J. Fodor, R. Mesiar, J. Montero, and A. Pradera, "Two methods for image compression/reconstruction using OWA operators," in *Recent Developments in the Ordered Weighted Averaging Operators: Theory and Practice*. Berlin, Germany: Springer-Verlag, 2010.
[18] [Online]. Available: http://decsai.ugr.es/cvg/dbimagenes/g512.php
[19] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 349–366, Feb. 2007.

Aranzazu Jurio received the M.Sc. degree in computer sciences in 2008 from the Public University of Navarra, Pamplona, Spain, where she is currently working toward the Ph.D. degree and holds a research position in the Department of Automatics and Computation.

Her research interests are image processing, focusing on magnification and segmentation, and applications of fuzzy sets and their extensions.

Miguel Pagola received the M.Sc. degree in industrial engineering from the Public University of Navarra, Pamplona, Spain, in 2000.

He is an Associate Lecturer with the Department of Automatics and Computation, Public University of Navarra. He is the author of over 14 published original articles and involved in teaching artificial intelligence to students of computer science. His research interests include fuzzy techniques for image processing, fuzzy set theory, medical image segmentation, and medical data mining.

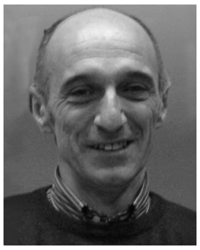
Radko Mesiar received the Ph.D. degree from Comenius University, Bratislava, Slovakia, in 1979 and the D.Sc. degree from Czech Academy of Sciences, Prague, Czech Republic, in 1996.

He is currently the Head of the Department of Mathematics, Faculty of Civil Engineering, Slovak University of Technology, Bratislava, Slovakia. He has been a Fellow Member of the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Prague, Czech Republic, since 1995 and the Institute for Research and Application of Fuzzy Modeling, University of Ostrava, Ostrava, Czech Republic, since 2006. He is the author of more than 200 papers in Web of Science and the coauthor of two scientific monographs and five edited volumes. His research interests include the area of uncertainty modeling, fuzzy logic and several types of aggregation techniques, nonadditive measures, and integral theory.

Dr. Mesiar is the founder and organizer of the Conferences of Fuzzy Set Theory and Applications and Aggregation Operators.

Gleb Beliakov (SM'08) received the Ph.D. degree in physics and mathematics from the Russian Peoples Friendship University, Moscow, Russia, in 1992.

He was a Lecturer and a Research Fellow with Los Andes University, Bogota, Colombia; the University of Melbourne, Victoria, Australia; and the University of South Australia, Adelaide, Australia. He is currently an Associate Professor with the School of Information Technology, Deakin University, Burwood, Australia. He is the author of a hundred research papers in the areas below and a number of software packages. His research interests are fuzzy systems, aggregation operators, multivariate approximation, global optimization, decision support systems, and applications of fuzzy systems in health care.



Humberto Bustince (M'06) received the Ph.D. degree in mathematics from the Public University of Navarra, Pamplona, Spain, from 1994.

He is a Full Professor with the Department of Automatics and Computation, Public University of Navarra. He is the author of more than 80 papers in Web of Science. His research interests are fuzzy logic theory, extensions of fuzzy sets (type-2 fuzzy sets, interval-valued fuzzy sets, Atanassov's intuitionistic fuzzy sets), fuzzy measures, aggregation operators, and fuzzy techniques for image processing.