

RESEARCH

Open Access

# A novel method in adaptive image enlargement

Mozhgan Bayat<sup>1</sup>, Ghazaleh Kafaie<sup>1\*</sup>, Ali Ayremlou<sup>2</sup> and Farrokh Marvasti<sup>1</sup>

## Abstract

This article introduces a new adaptive method for image interpolation. In order to obtain a high resolution (HR) image from its low resolution (LR) counterpart (original image), an interpolator function (array) is used, and the main focus of this manuscript is to formulate and define this function. By applying this interpolator function to each row and column of a LR image, it is possible to construct its HR counterpart. One of the main challenges of image interpolation algorithms is to maintain the edge structures while developing an HR image from the LR replica. The proposed approach overcomes this challenge and exhibits remarkable results at the image edges. The peak signal to noise ratio and structural similarity criteria by using this innovative technique are notably better than those achieved by alternative schemes. Also, in terms of implementation speed, this method displays a clear advantage and outperforms the high performance algorithms in the ability to decrease the artifact results of image enlargement such as blurring and zigzagging.

**Keywords:** Image interpolation, Linear least square, Low resolution, High resolution

## Introduction

Image interpolation in this article refers to algorithms that transform a low resolution (LR) image to high resolution (HR) one and has many applications. In medical applications, it is highly preferred for images to have high resolution while medical equipments cannot produce images with resolutions higher than a specific standard. In another application, with the development of digital monitors, we need to transform Standard Definition TeleVision (SDTV) video frames to High Definition TeleVision (HDTV) equivalents. Another application of this technology is to find missing pixels or blocks in an image.

Traditional methods such as bilinear, bicubic, and cubic convolution [1,2] are the common methods in image enlargement. Although the merits of these techniques are simplicity and fast implementation, they suffer from visual degradations such as jagged edges, blurring, and ringing around the image edges.

Human visual system is sensitive to the edges of the objects in an image. As a result, there are several mechanisms which improve the edges of an image.

Undoubtedly, as the quality of image enhances, the computational complexity increases. Hence, in all of these methods, there is a trade off between performance and affordable complexity. In the last few decades, several authors have carried out researches on this topic. Along the same line, we briefly introduce [3-10] as follows.

In NEDI [3], the aim is to maintain the edge structure in HR images. This is a nonlinear algorithm whose basic idea is to estimate the covariance of HR image from the covariance of LR counterpart. The estimation is based on the geometric duality between low and high resolution. The algorithm presented in [4] which is based on directional filtering and data fusion to find the missing pixel, considers two orthogonal sets which produce an estimation of the pixel value. In SAI [5], using a moving window in the input LR image, one can find the model parameters. The pixel structure obtained in this way can be led to a block of both available and estimated pixels using a soft-decision estimation process. In WZPCS [6], wavelet transformation is applied to LR image. Using sub-bands of the resultant image, one can estimate sub-bands of HR image. Afterwards, the HR image can be obtained by applying the inverse wavelet transform.

\*Correspondence: ghazal90kafaie@gmail.com

<sup>1</sup>Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran

Full list of author information is available at the end of the article

In Optimized spline [7], by taking advantages of calculus of variations, the optimization problem is simplified from a nonlinear infinite dimensional case to a linear finite dimensional one in order to design compact-support interpolation kernels for a class of signals. The method presented in [8] tries to create up-scaled images that are suitable for real time applications and appear natural to human observer. In this algorithm a two step grid filling is used. Then, an iterative correction is obtained by minimizing an objective function depending on the second order directional derivatives of the image intensity.

In [9], in order to enlarge the image, it is down-sampled by using the bilinear method and then, the missing pixels are estimated by using a combination of the other pixels, while the desired coefficients are optimized by using a least mean square technique. In [10], the authors have extended the algorithm presented in [9] to display products such as video restoration.

In our approach, we work out the parameters that are used to enlarge an image from its LR counterpart. These parameters depend on the original image and the factor which we want to enlarge the image by. In order to enlarge an image or find the missing pixels in it, an interpolation array which is constructed by the mentioned parameters is convolved by the matrix that contains the values of the image pixels.

The rest of the article is organized as follows: The following section describes the proposed interpolation method for one-dimensional vectors. The two-dimensional application is proposed in section "Two-dimensional algorithm". Simulation results and comparisons are given in section "Simulation results", and section "Conclusion" concludes the paper.

### Algorithm description

In order to describe the proposed algorithm for image interpolation, we first apply it to one-dimensional vectors. Let  $X_{LR}$  be a  $1 \times N$  vector. Suppose we wish to enlarge  $X_{LR}$  to a  $kN$  sized vector  $X_{HR}$  using the following steps:

We put  $k - 1$  zeros between any two successive entries of vector  $X_{LR}$ .

In order to achieve these  $kN - N$  new entries of  $X_{HR}$ , we choose  $2M$  of  $N$  entries of  $X_{LR}$ .

The new value will be a linear combination of these  $2M$  entries.

In order to find the  $2M$  indeterminate multipliers of this linear combination, we should follow the procedure stated below, bearing in mind that for each fixed vector of any size the procedure and the multipliers applied to enlarge the vector are fixed. This means that, the multipliers used

for obtaining a  $1 \times N$  vector from its  $1 \times \lfloor \frac{N}{k} \rfloor$  counterpart are the same as the ones used to calculate a  $1 \times \lfloor \frac{N}{k} \rfloor$  vector from its  $1 \times \lfloor \frac{N}{k^2} \rfloor$  counterpart.

**Step(1-1):** As presented in (1) and (2), we down-sample the original vector  $X_{LR}$  by a factor of  $k$  to find  $X'_{1 \times \lfloor \frac{N}{k} \rfloor}$ .

**Step(1-2):** Again we down-sample  $X'_{1 \times \lfloor \frac{N}{k} \rfloor}$  by a factor of  $k$  to obtain  $X''_{1 \times \lfloor \frac{N}{k^2} \rfloor}$  vector.

**Step(1-3):** Now, by constructing  $1 \times \lfloor \frac{N}{k} \rfloor$  vector  $X'$  from vector  $X''$  of size  $1 \times \lfloor \frac{N}{k^2} \rfloor$ , we can find the multipliers used to obtain the enlarged image. To approach this, we must zero-pad  $X''$  by a factor of  $k$  to build vector  $X''_{ZP}$ . This means that,  $k - 1$  zeros are inserted between every two successive entries of  $X''$ .

As it was stated earlier, by using these multipliers, we can find a new vector  $X_{HR}$  from  $X'_{1 \times \lfloor \frac{N}{k} \rfloor}$ .

In order to evaluate the enlarged image, we take advantages of the criteria peak signal to noise ratio (PSNR) and Structural SIMilarity (SSIM) [11] by down-sampling the original vector by a factor of  $k$  and interpolating it using the proposed algorithm.

$$X = (x_1 \ x_2 \ \dots \ x_N) \quad (1)$$

down-sample by a factor of  $k \downarrow$

$$X' = (x_1 \ x_{k+1} \ x_{2k+1} \ \dots \ x_{(\lfloor \frac{N}{k} \rfloor - 1)k+1}) \quad (2)$$

down-sample by a factor of  $k \downarrow$

$$X'' = (x'_1 \ x'_{k+1} \ x'_{2k+1} \ \dots \ x'_{(\lfloor \frac{N}{k^2} \rfloor - 1)k+1}) \quad (3)$$

Zero padding  $\downarrow$

$$X''_{ZP} = \left( x''_1 \ \underbrace{0 \dots 0}_{(k-1)\text{times}} \ x''_2 \ \underbrace{0 \dots 0}_{(k-1)\text{times}} \ x''_3 \ \dots \ 0 \ x''_{\lfloor \frac{N}{k^2} \rfloor} \right) \quad (4)$$

**Step(1-4):** In order to compute vector  $X'$  from  $X''_{ZP}$ , we must insert a linear combination of  $x''_i$ 's;  $i = 1 : \lfloor \frac{N}{k^2} \rfloor$  instead of each zero in  $X''_{ZP}$ . For this purpose, we can convolve  $X''_{ZP}$  with an interpolator vector,  $A$  (which is determined by (5)).

In this equation,  $A$  is a  $1 \times (1 + 2(M - 1) + 2M(k - 1))$  vector. The  $1 + (M - 1) + M(k - 1)$ <sup>st</sup> entry, which is located at the middle of the vector, is 1. Except for the mentioned entry, the entries with the indices that are multiples of  $k$  are 0, and there are  $(k - 1)$   $a'_i$ 's and  $(k - 1)$   $a_i$ 's between

any two successive zeros at the left and right side of the entry 1, respectively which can be calculated by using the proposed algorithm.

In (5), the parameter  $M$  can take an optional value which is set to 2 in our experiments. However, by setting  $M$  higher than this value, implementation speed decreases while image quality increases.

$$A = [a'_{M(k-1)} a'_{M(k-1)-1} \dots a'_{(M-1)(k-1)+1} 0 a'_{(M-1)(k-1)} a'_{(M-1)(k-1)-1} \dots a'_{(M-2)(k-1)+1} 0 \dots a'_{k-1} \dots a'_2 a'_1 1 a_1 a_2 \dots a_{k-1} 0 \dots a_{(M-2)(k-1)+1} \dots a_{(M-1)(k-1)} 0 a_{(M-1)(k-1)+1} \dots a_{M(k-1)}]$$
 (5)

$$X' = X''_{ZP} * A$$
 (6)

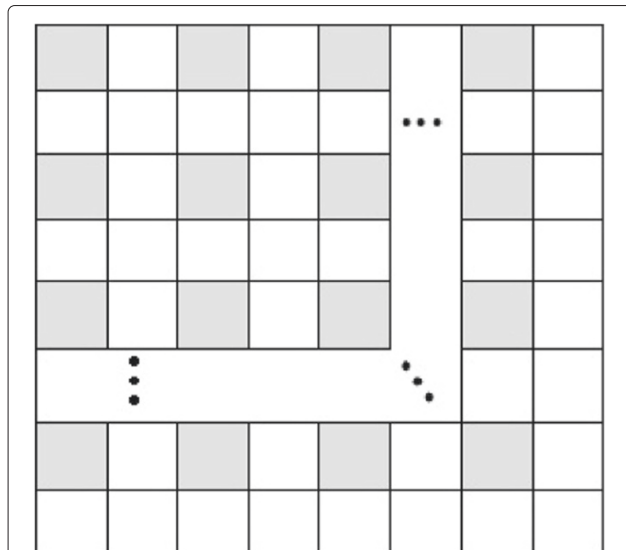
In (6),  $X'$  and  $X''_{ZP}$  are known and  $A$  is unknown, and there are  $[\frac{N}{k}] - [\frac{N}{k^2}]$  equations and  $2M(k-1)$  unknown variables. This statement means that, if (6) is to have a solution,  $[\frac{N}{k}] - [\frac{N}{k^2}]$  must be greater than or equal to  $2M(k-1)$ .

**Step(1-5):** To solve (6), we have the following matrix equation:

$$B_{[\frac{N}{k}] \times (1+2(M-1)+2M(k-1))} \times A_{(1+2(M-1)+2M(k-1)) \times 1} = C_{[\frac{N}{k}] \times 1}$$
 (7)

where  $C = X'^T$ , and  $A$  denotes the interpolator matrix.

**Step(1-6):** In order to find matrix  $B$ , We add  $(Mk-1)$



**Figure 1** A zero-padded image by factors of  $k_1 = k_2 = 2$ .

zeros to the beginning and the end of vector  $X''_{ZP}$  to obtain vector  $X''_{ZP0}$ .

$$X''_{ZP0} = (0 \dots 0 x''_{ZP,1} x''_{ZP,2} \dots x''_{ZP, [\frac{N}{k}]} 0 \dots 0)$$
 (8)

where  $x''_{ZP,i}$  is the  $i^{\text{th}}$  entry of vector  $X''_{ZP}$ .

**Step(1-7):** By using  $X''_{ZP0}$ , we can find matrix  $B$  as follows:

$$B_{i*} = X''_{ZP0}(i : i + \text{size}(A) - 1) = X''_{ZP0}(i : i + 2M(k-1) - 1)$$
 (9)

$$i = 1 : \text{size}(X') = 1 : [\frac{N}{k}]$$

where  $B_{i*}$  is the  $i^{\text{th}}$  row of matrix  $B$ .

Now, following Step(1-5) and then Step(1-4), we can obtain the multipliers used for image enlargement.

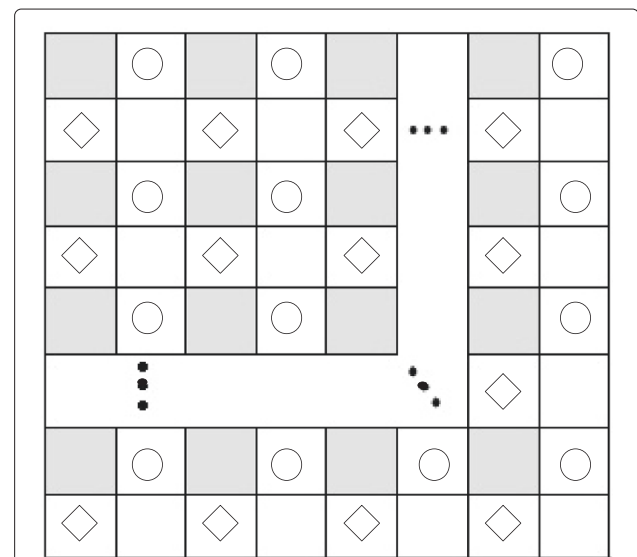
To understand the above algorithm, we can illustrate the above relation with an example in which  $N = 36, K = 3, M = 2$ .

**Step(1-1):**

$$X' = \text{downsample}(X, 3) = (x_1 \ x_4 \ x_7 \ x_{10} \ \dots \ x_{25} \ x_{28} \ x_{31} \ x_{34}) = (x'_1 \ x'_2 \ x'_3 \ x'_4 \ \dots \ x'_9 \ x'_{10} \ x'_{11} \ x'_{12})$$
 (10)

**Step(1-2):**

$$X'' = \text{downsample}(X', 3) = (x''_1 \ x''_4 \ x''_7 \ x''_{10})$$
 (11)



**Figure 2** Blocks containing circles are obtained by applying step (1), and those with diamonds are obtained by applying step (2) to Figure 1.

**Table 1 PSNR**

Images	Bilinear	Bicubic	WZP-CS	SAI	NEDI	Optimized spline	Proposed method
Lena	29.8768	29.7936	29.7871	32.5795	32.999	32.2976	<b>33.4913</b>
Couple	26.7166	26.5342	26.6294	27.4276	27.8451	27.912	<b>28.6865</b>
Barbara	23.6373	23.0442	23.5365	23.0267	21.1754	25.1	<b>25.3936</b>
Fishing boat	26.8857	26.7076	26.8139	28.1566	28.1784	28.5054	<b>28.8645</b>
Cameraman	30.0606	30.2018	29.972	32.2363	32.995	33.2886	<b>38.1691</b>
Crowd	19.8674	19.6677	19.7829	20.7484	21.2964	21.8277	<b>21.906</b>
Lake	27.1699	27.0709	27.0786	28.6805	28.5925	28.4984	<b>29.8961</b>
Baboon	22.0974	21.7332	22.0273	22.6396	23.1457	22.509	23.0153
Girl	30.2527	29.8225	30.1298	29.2056	31.7538	30.9017	31.605
Average	26.2849	26.0639	26.1952	27.1889	27.5534	27.8711	<b>29.0030</b>

**Step(1-3):**

$$X''_{ZP} = (x'_1 \ 0 \ 0 \ x'_4 \ 0 \ 0 \ x'_7 \ 0 \ 0 \ x'_{10}) \quad (12)$$

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & x'_1 & 0 & 0 & 0 & x'_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & x'_1 & 0 & 0 & 0 & x'_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & x'_1 & 0 & 0 & 0 & x'_4 & 0 & 0 \\ 0 & 0 & x'_1 & 0 & 0 & 0 & x'_4 & 0 & 0 & 0 & x'_7 & 0 \\ 0 & x'_1 & 0 & 0 & 0 & x'_4 & 0 & 0 & 0 & x'_7 & 0 & 0 \\ x'_1 & 0 & 0 & x'_4 & 0 & 0 & 0 & x'_7 & 0 & 0 & 0 & x'_{10} \\ 0 & 0 & x'_4 & 0 & 0 & 0 & x'_7 & 0 & 0 & 0 & x'_{10} & 0 \\ 0 & x'_4 & 0 & 0 & 0 & x'_7 & 0 & 0 & 0 & 0 & x'_{10} & 0 \\ x'_4 & 0 & 0 & x'_7 & 0 & 0 & 0 & 0 & x'_{10} & 0 & 0 & 0 \\ 0 & 0 & x'_7 & 0 & 0 & 0 & x'_{10} & 0 & 0 & 0 & 0 & 0 \\ 0 & x'_7 & 0 & 0 & 0 & x'_{10} & 0 & 0 & 0 & 0 & 0 & 0 \\ x'_7 & 0 & 0 & x'_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (13)$$

**Two-dimensional algorithm**

A typical image is a matrix of  $N$  rows and  $N$  columns which can be enlarged by a factor of  $k$  to  $kN \times kN$  by applying the one-dimensional algorithm discussed in the previous section to each row and column separately. Enlargement requires that we first down-sample the given matrix by factors of  $k_1$  and  $k_2$ , then up-sample the modified  $[\frac{N}{k_1}] \times [\frac{N}{k_2}]$  matrix by factors of  $k_1$  and  $k_2$  to obtain a

new  $N \times N$  matrix. In order to illustrate the mentioned statement, we assume a  $16 \times 16$  matrix  $F$ . We also define a matrix  $F'$  with entries that are the odd entries of  $F$ . Down-sampling  $F'$  by a factor of  $k = 2$ , we can obtain  $F''$ , and up-sampling  $F''$  by a factor of  $k = 2$  leads us to  $F''_{up}$ .

In this case, we know all the entries of  $F$ . Equation (7) can be applied to each row and column of  $F'$ . As an illustration, let us consider the following statement for the first row:  $C$  is the first row of matrix  $F'$ ,  $B$  is obtained using (9), and matrix  $A$  is unknown. As a result, we should use (7) eight times, four times for the rows, and four times for the columns of matrix  $F$ . In as much as (7), itself, contains  $n$  equations, where  $n$  is the size of vector  $C$ , we have  $4 \times 8 = 32$  equations (where four is the number of equations:  $[\frac{N}{k}] - [\frac{N}{k^2}] = [\frac{16}{2}] - [\frac{16}{4}]$ ). We use matrices  $B$  and  $C$  that are obtained from (7) as the blocks of new matrices named  $B'$  and  $C'$ , respectively. Then, these 32 equations are solved using linear least square method; moreover,  $A$  can be calculated as follows:

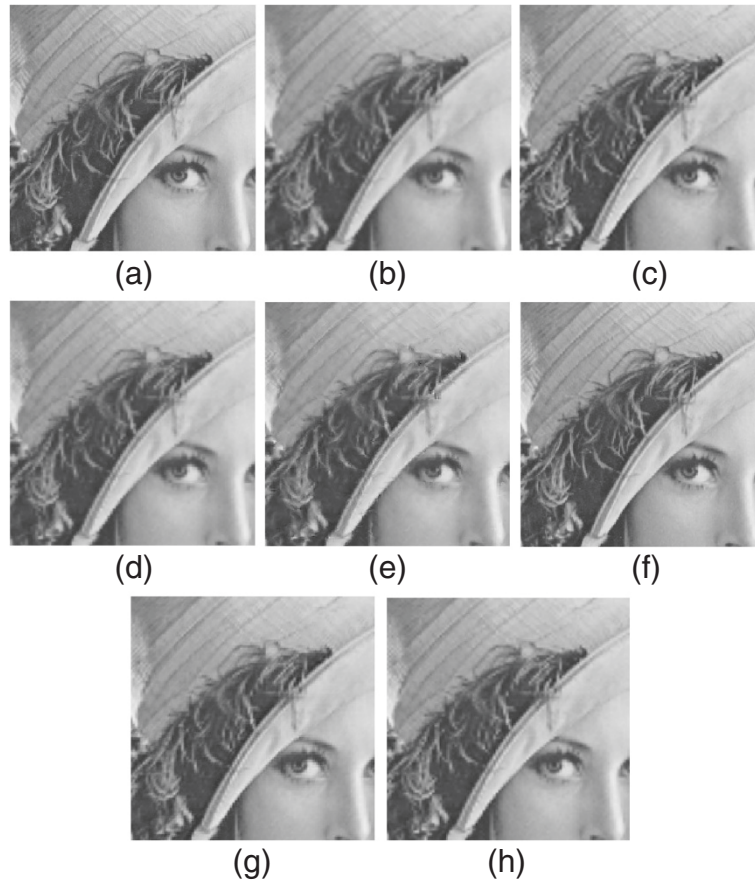
$$B' \times A = C' \quad (14)$$

Hence:

$$A = (B'^T B')^{-1} B'^T C' \quad (15)$$

**Table 2 SSIM**

Images	Bilinear	Bicubic	WZP-CS	SAI	NEDI	Optimized spline	Proposed method
Lena	0.8735	0.8737	0.8728	0.8843	0.9076	0.9046	0.9075
Couple	0.7809	0.7863	0.7825	0.8062	0.8304	0.8306	<b>0.8356</b>
Barbara	0.7346	0.7414	0.7397	0.7646	0.7473	0.7885	0.7592
Fishing boat	0.7868	0.7882	0.7871	0.8115	0.8252	0.8339	<b>0.8348</b>
Cameraman	0.937	0.9448	0.9374	0.9368	0.9568	0.9621	<b>0.9648</b>
Crowd	0.7091	0.7229	0.7129	0.7718	0.783	0.801	0.7984
Lake	0.8032	0.802	0.8028	0.8232	0.8374	0.8474	0.8392
Baboon	0.6182	0.6367	0.6262	0.6899	0.7195	0.6766	0.7026
Girl	0.7486	0.7337	0.7462	0.7491	0.7802	0.7616	0.7648
Average	0.7768	0.7810	0.7786	0.8041	0.8208	0.8228	<b>0.8229</b>



**Figure 3** Comparison of different methods for the Lena image: (a) the original image, (b) bilinear interpolation, (c) bicubic interpolation, (d) WZP cycle-spinning [6], (e) SAI [5], (f) NEDI [3], (g) optimized spline [7], and (h) the proposed method.

Now, using  $A$ , we zero pad matrix  $F'$  by factors of  $k_1$  and  $k_1$ , through the following steps:

$$F'_{8 \times 8} = \begin{pmatrix} f_{1,1} & f_{1,3} & \cdots & f_{1,15} \\ f_{3,1} & f_{3,3} & \cdots & f_{3,3} \\ \vdots & \vdots & \ddots & \vdots \\ f_{15,1} & f_{15,3} & \cdots & f_{15,15} \end{pmatrix} \quad (16)$$

$$F''_{4 \times 4} = \begin{pmatrix} f_{1,1} & f_{1,5} & f_{1,9} & f_{1,13} \\ f_{5,1} & f_{5,5} & f_{5,9} & f_{5,13} \\ f_{9,1} & f_{9,5} & f_{9,9} & f_{9,13} \\ f_{13,1} & f_{13,5} & f_{13,9} & f_{13,13} \end{pmatrix} \quad (17)$$

$$F''_{up} = \begin{pmatrix} f_{1,1} & 0 & f_{1,5} & 0 & \cdots & f_{1,13} \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ f_{5,1} & 0 & f_{5,5} & 0 & \cdots & f_{5,13} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ sf_{13,1} & 0 & f_{13,5} & 0 & \cdots & f_{13,13} \end{pmatrix} \quad (18)$$

Figure 1 shows an  $[\frac{N}{k}] \times [\frac{N}{k}]$  image that is zero-padded by factors of  $k_1 = k_2 = 2$ .

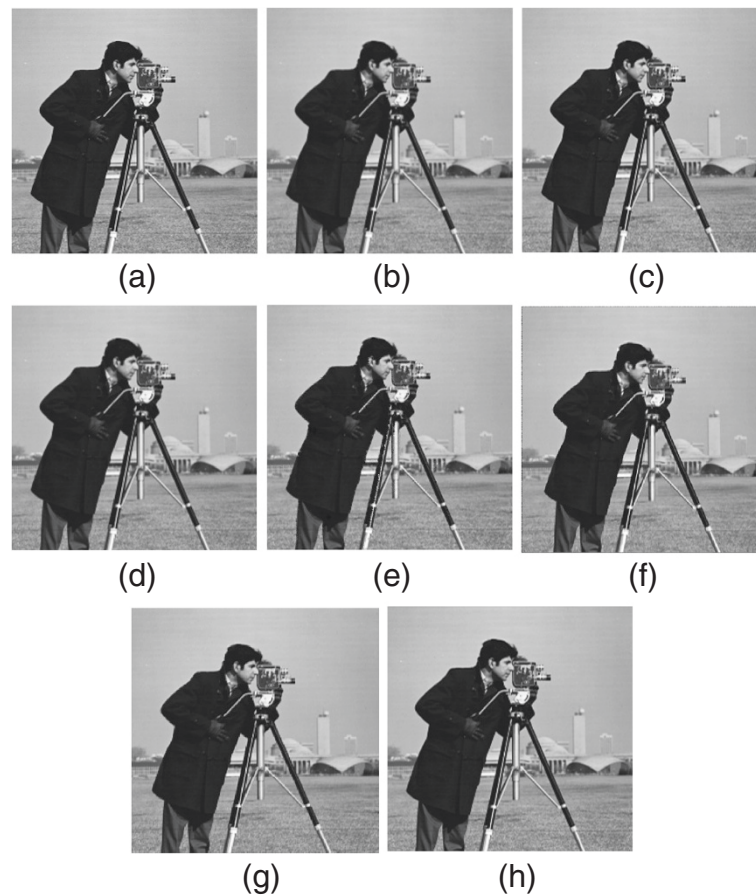
The gray blocks in Figure 1 contain the pixels of the main  $N \times N$  image and the remaining blocks contain zero. Now, in order to apply the proposed method to this figure, we need to take the following steps:

**Step (2-1):** We first inspect the odd rows and then apply the rule mentioned before to each one to determine the amounts which must be replaced with zeros in the white blocks in Figure 1, using the amounts of adjacent blocks.

**Step (2-2):** Next, we apply step (1) to the odd columns.

As can be seen from Figure 1, even rows and even columns contain zeros; therefore by using steps (1) and (2), the values that some of these blocks contain can be calculated. In Figure 2, blocks that contain circles are obtained by applying step (1), and those containing diamonds are obtained by applying step (2) to Figure 1.

Consequently, only the blocks in even rows and columns hold zeros (white blocks in Figure 2), and the goal is to



**Figure 4** Comparison of different methods for the Cameraman image: (a) the original image, (b) bilinear interpolation, (c) bicubic interpolation, (d) WZP cycle-spinning [6], (e) SAI [5], (f) NEDI [3], (g) optimized spline [7], and (h) the proposed method.

determine the values of these blocks. To overcome this challenge, we can use one of the following approaches:

**Method (1):** After applying step (1), the blocks that contain circles in Figure 2 are identified. Now, by applying step (2) to the same columns, we can obtain the remaining unknown blocks.

**Method (2):** After applying step (2), the blocks that contain diamonds in Figure 2 are spotted. Then, by applying step (1) to the same rows, the remaining unknown blocks are extracted.

**Method (3):** We can average the amounts obtained through methods (1) and (2) to each block to work out a new value.

In this way, the proposed method can provide us a high resolution image from a low resolution one. We can find the interpolator multipliers by

employing the downsampled version of the original image. Then, the HR image can be estimated by using the original LR image and the computed multipliers.

### Simulation results

The proposed image interpolation algorithm was implemented and its performance was compared with some existing methods. The results of the proposed method simulation are benchmarked against some classic methods such as bilinear, bicubic, and some recent methods such as SAI [5] and WZP-CS [6] that are based on wavelet transform, and the results are tabulated in Tables 1 and 2. Table 1 shows the PSNR of implementing the mentioned methods on nine  $256 \times 256$  images, in order to enlarge them by a factor of 2, and the SSIM of them is indicated in Table 2. The simulations are implemented using Matlab 7.10.0.499(R2010a) software on a VAIO SR 490 DCB laptop with an Intel Core2 Duo P8800 2.66 GHz CPU and a 4 GB RAM.

**Table 3 Average calculation time for enlarging a 256 \* 256 image by a factor of 2**

Images	Bilinear	Bicubic	WZP-CS	SAI	NEDI	Optimized spline	Proposed method
Average calculation time (in seconds)	0.0112	0.0137	2.85	8.6846	1.4103	3.7137	1.7579

As can be seen from these two tables, the results achieved by our proposed approach outperform those from other solutions. The bold values given in each row of these two tables indicate that the proposed method outperforms the other compared methods when applied to the images of the mentioned rows. The average PSNR of implementing the proposed method, is 29.0030 dB, which is higher than the other values of the compared solutions. On the other hand, the average SSIM of implementing the proposed method on the mentioned images is 0.8229 dB which is higher than the average SSIM for the other compared algorithms. In order to compare the methods objectively, Figure 3 shows that, applying the bilinear method yields blurring and also has less implementation speed than the proposed method. In addition, the scheme based on WZP-CS causes damages at the edges of the image. In the Cameraman image (Figure 4), applying the proposed method produces a better visual quality compared to other known models.

Also, since we do not use blocks in this presentation, unlike the SAI method, there is no blocking result in the obtained image.

Moreover, one of the most important criteria that should be taken into account for comparison is computational complexity of the algorithms, which can be computed by considering the number of operations in the algorithms. In the proposed method, the number of multiplications and summations in order to calculate the matrix  $A$  from expression (15) for an  $N \times N$  image to be enlarged by a factor of  $K$  are  $[(\frac{N}{K})(2MK - 1)(2MK) + (2MK - 1)! + (2MK - 1)(6MK - 1)]$  and  $[(\frac{N}{K})(2MK - 1)(2MK) + 0.5(2MK - 1)! - 3(2MK - 1)]$ , respectively; where  $2M$  is the number of entries of the LR image that are used for interpolation of each pixel. By assuming  $M = K = 2$  in our experiments, there will be  $(28N + 5201)$  multiplications and  $(28N + 2499)$  summations for calculating the matrix  $A$ . On the other hand, after calculating the matrix  $A$ , the number of multiplications and summations to estimate the value of each pixel are  $2M(N \times N - \frac{N}{K} \times \frac{N}{K})$  and  $(2M - 1)(N \times N - \frac{N}{K} \times \frac{N}{K})$ , respectively. By assuming  $M = K = 2$  in our experiments, there will be  $3N^2$  multiplications and  $\frac{9}{4}N^2$  summations for estimating the value of each pixel. The number of operations to estimate the value of each pixel is approximately on the same order in the compared methods, and the difference is in the number of operations in order to calculate the multipliers of

the linear combination, hence, in order to compare the proposed method with the other mentioned ones, we consider the number of operations in order to calculate the matrix  $A$ , where, this number of operations is acceptable for our purpose.

In addition to this, since all algorithms are simulated in the same situation, running time may have a better sense of complexity. As it is indicated in Table 3, the proposed method outperforms the high performance algorithms such as SAI and optimized spline in terms of the average time that is needed to simulate the algorithm, and features a higher implementation speed.

The linear least square study unveiled in this article also offers an advantage when transmitting an image where we have memory limitation; i.e. instead of transmitting the enlarged version of the original image by a factor of  $k$ , we can transmit the original image and the interpolator matrix  $A$ , and then use the interpolator matrix to enlarge the original image by a factor of  $k$  at the receiving end. Accordingly, less memory is used in transmitting the image. In the images that are simulated, the interpolator matrix entries are decimal numbers with three digits to the right of the decimal point; hence, ten bits would be sufficient in order to send the interpolator matrix as side information.

## Conclusion

In this article, we presented a novel image interpolation method. For each image to be interpolated, we found an interpolator function which was applied to each row and column of the image separately by convolving this interpolator function with the vector considered for interpolation. PSNR and SSIM criteria were then used to compare our results versus those obtained by other algorithms. Experimental results demonstrated that the solution revealed in this article achieves better results both objectively and subjectively.

Another key feature of the proposed method is that it allows enlargement of the image by any factor of  $k = 2, 3, 4, 5, \dots$  by simply setting this parameter to the desired value.

## Competing interests

The authors declare that they have no competing interests.

## Acknowledgements

The authors would like to thank Seyyed Amirhossein Hosseini for editing this article and his helpful comments.

#### Author details

<sup>1</sup>Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran. <sup>2</sup>Department of Electrical and Computer Engineering, Rice University, Houston, TX, USA.

Received: 2 March 2012 Accepted: 30 July 2012

Published: 7 September 2012

#### References

1. R Keys, Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoust. Speech Signal Process.* **29**(6), 1153–1160 (1981)
2. HS Hou, HC Andrews, Cubic splines for image interpolation and digital filtering. *IEEE Trans. Acoust. Speech Signal Process.* **26**(6), 508–517 (1978)
3. X Li, MT Orchard, New edge-directed interpolation. *IEEE Trans. Image Process.* **10**(10), 1521–1527 (2001)
4. L Zhang, X Wu, An edge guided image interpolation algorithm via directional filtering and data fusion. *IEEE Trans. Image Process.* **15**(8), 2226–2238 (2006)
5. X Zhang, X Wu, Image interpolation by adaptive 2-D autoregressive modeling and soft-decision estimation. *IEEE Trans. Image Process.* **17**(6), 887–896 (2008)
6. A Temizel, T Vlachos, W Visioprime, Wavelet domain image resolution enhancement using cycle-spinning. *Electron. Lett.* **41**(3), 119–121 (2005)
7. R Madani, A Ayremlou, A Amini, F Marvasti, Optimized compact-support interpolation kernels. *IEEE Trans. Signal Process.* **60**(2), 626–633 (2012)
8. A Giachetti, N Asuni, Real time artifact-free image upscaling. *IEEE Trans. Image Process.* **20**(10), 2760–2768 (2011)
9. L Shao, H Hu, G de Haan, Coding artifact robust resolution up-conversion. in *IEEE International Conference on Image Processing*, vol. 5 (San Antonio, Texas, USA, Sept 2007), pp. 409–412
10. L Shao, H Zhang, G de Haan, An overview and performance evaluation of classification based least squares trained filters. *IEEE Trans. Image Process.* **17**(10), 1772–1782 (2008)
11. Z Wang, AC Bovik, HR Sheikh, EP Simoncelli, Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)

doi:10.1186/1687-6180-2012-197

**Cite this article as:** Bayat et al.: A novel method in adaptive image enlargement. *EURASIP Journal on Advances in Signal Processing* 2012 **2012**:197.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---