

A New Technique for Image Magnification

Carlo Arcelli, Maria Frucci, and Gabriella Sanniti di Baja

Institute of Cybernetics "E. Caianiello", CNR, Pozzuoli (Naples), Italy
c.arcelli@cib.na.cnr.it, m.frucci@cib.na.cnr.it,
g.sannitidibaja@cib.na.cnr.it

Abstract. A discrete technique for image magnification is presented, which produces the resulting image in one scan of the input image and does not require any threshold. The technique allows the user to magnify an image with any integer zooming factor. The performance of the algorithm is evaluated by using the standard criterion based on the Peak Signal to Noise Ratio PSNR. The obtained results are visually good, since artifacts do not significantly affect the magnified images.

1 Introduction

Image magnification is a process that, starting from an input image at a given resolution, virtually generates a higher resolution image. This process is involved in a number of applications, e.g., for matching images captured by sensors with different capturing resolution, satellite image analysis, medical image display, entertainment, and image compression and transmission.

Among the many approaches to image magnification available in the literature, the most common ones are based on nearest-neighbor, bilinear, and bicubic interpolation [1]. The main drawback of the nearest-neighbor interpolation method is the too blocky appearance of the obtained higher resolution image, while bilinear and bicubic interpolations tend to blur the resulting image. Edge sharpness is better maintained if non-linear interpolation is adopted [2], or when more sophisticated methods are used, e.g., following the Bayesian maximum a posteriori approach [3], wavelet-based approach [4], fractal-based approach [5-7], and PDEs-based approach [8,9]. In any case, staircase artifacts affect the resulting image, especially when the magnifying factor is larger than 4. These artifacts can be reduced by adaptive techniques, which however often imply a significant increase in computational cost [10-12].

In the framework of interpolation methods, we introduce here a new technique to associate in the output image a block of size $n \times n$ (n integer) to each pixel p with gray level p in the input image. The gray level q to be assigned to each of the pixels in the $n \times n$ block associated to p is computed by taking into account a neighborhood of size $m \times m$ centered on q , where m is the smallest odd integer larger than n , i.e., $m = n + 1 + \text{mod}(n, 2)$. The $m \times m$ neighborhood is used, once the magnification factor is selected, only to determine the weights for the neighbors of p actually involved in the computation of each q . The gray level of q is computed as the weighted average of p and three suitable neighbors of p . The magnified version is computed in one scan of the input image.

To evaluate the performance of our technique, we use as input an image whose resolution has been reduced starting from an original high resolution image and compute the Peak Signal to Noise Ratio (PSNR) by taking into account the original high resolution image and the high resolution image we generate. Then, we compare the resulting PSNR, to the PSNRs obtained when nearest-neighbor, bilinear and cubic interpolations are used. We show that the new technique generally increases the PSNR with respect to the compared ones, and produces visually appealing results.

The magnification procedure has been devised for gray level images, where the gray levels are in the range [0-255]. Work on its generalization to color images is in progress.

2 The Basic Algorithm

Let LR be a low resolution input image consisting of $I \times J$ pixels, and let n , with n integer, be the selected zooming factor. The reconstructed image HR at higher resolution will consist of $nI \times nJ$ pixels.

The eight neighbors of a pixel $p_{i,j}$ of LR are denoted as tl (top left), t (top), tr (top right), r (right), br (bottom right), b (bottom), bl (bottom left) and l (left), by taking into account their positions with respect to $p_{i,j}$, as shown in Fig. 1.

tl	t	tr
l	$p_{i,j}$	r
bl	b	br

Fig. 1. The eight neighbors of $p_{i,j}$ in the low resolution input image LR

A block of $n \times n$ pixels in HR is associated to each pixel $p_{i,j}$ of LR. The gray level of each pixel $q_{i',j'}$ in the $n \times n$ block is computed in terms of the gray levels of $p_{i,j}$ and its neighbors in LR. To this aim, we introduce a mask of multiplicative weights, to be used for the gray levels of $p_{i,j}$ and its neighbors. To determine the values of the weights, for each pixel $q_{i',j'}$ of the block associated to $p_{i,j}$, we consider a neighborhood of size $m \times m$. The value m is the smallest odd integer larger than n . For simplicity, let us suppose that the selected zooming factor is $n=4$. Hence, it is $m=5$ and the neighborhood centered on any $q_{i',j'}$ has size 5×5 .

With reference to Fig. 2, the nine 4×4 blocks, framed by thick gray lines, correspond in HR to $p_{i,j}$ and to its eight neighbors in LR. The 5×5 neighborhood centered on $q_{i',j'}$ (the gray pixel in the central block) is framed by thick black lines. We observe that the 5×5 neighborhood of $q_{i',j'}$ partially overlaps the four blocks in HR associated with the pixels t , tr , r and $p_{i,j}$ in LR. Namely, the 5×5 neighborhood of $q_{i',j'}$ overlaps the block associated with t by 6 pixels, the block associated to tr by 4 pixels, the block associated to r by 6 pixels and the block associated to $p_{i,j}$ by 9 pixels. The values 6, 4, 6 and 9 are the multiplicative weights to be used for the gray levels of t , tr , r and $p_{i,j}$, respectively. The weighted average gray level is computed and assigned to the pixel $q_{i',j'}$. Of course, when another pixel $q_{i'',j''}$ is selected in the block, its neighborhood will

possibly overlap blocks associated to different neighbors of $p_{i,j}$, or overlapping will involve a different number of pixels. Thus, proper weights have to be computed to determine the gray level to be assigned to each pixel belonging to the $n \times n$ block associated to $p_{i,j}$. The coordinates of each pixel $q_{i',j'}$ in the $n \times n$ block associated to $p_{i,j}$ in HR are computed as $i' = ni+k, j' = nj+h$, for $k=0 \dots n-1$ and $h=0 \dots n-1$.

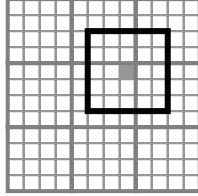


Fig. 2. The $n \times n$ blocks associated in HR to a pixel $p_{i,j}$ of LR and to its neighbors are framed by thick gray lines. The $m \times m$ neighborhood centered pixel $q_{i',j'}$ (gray square) belonging to the $n \times n$ block associated to $p_{i,j}$, is framed by thick black lines.

In LR, let z be any of the nine pixels in the 3×3 window centered on $p_{i,j}$, i.e., $z \in \{tl, t, tr, l, p_{i,j}, r, bl, b, br\}$. For every z , let $z\#$ denote the number of pixels in the intersection between the $n \times n$ block associated to z in HR and the $m \times m$ neighborhood associated to each $q_{i',j'}$ that has to be created in correspondence with $p_{i,j}$. In detail, $z\#$ is given by the following expressions:

$$\begin{aligned} tl\# &= a(k) \times a(h) \\ t\# &= c(k) \times a(h) \\ tr\# &= b(k) \times a(h) \\ l\# &= a(k) \times c(h) \\ p_{i,j}\# &= c(k) \times c(h) \\ r\# &= b(k) \times c(h) \\ bl\# &= a(k) \times b(h) \\ b\# &= c(k) \times b(h) \\ br\# &= b(k) \times b(h) \end{aligned}$$

where $a(x) = \max(0, \lfloor m/2 \rfloor - x)$, $b(x) = \max(0, x + m - n - \lfloor m/2 \rfloor)$ and $c(x) = m - a(x) - b(x)$, with $x \in \{k, h\}$

Then, the gray level of each $q_{i',j'}$ is computed as follows:

$$q_{i',j'} = \sum(z\# \times z) / m \times m$$

Obviously, special care should be taken to correctly compute the gray level $q_{i',j'}$ when $p_{i,j}$ belongs to the first and the last row or column of LR.

2.1 Improved Algorithm

The basic algorithm can be improved as follows, at the expenses of a modest increase in the computational cost. As soon as the block of $n \times n$ pixels associated to $p_{i,j}$ has

been computed, the minimum gray level $qmin$ and the maximum gray level $qmax$ in the block are detected. Then, the gray level of each pixel $q_{i',j'}$ in the block is updated as follows:

if $p_{i,j} \leq qmin$, then $q_{i',j'} = (q_{i',j'} - \Delta)$, for any $q_{i',j'} > p_{i,j}$,

if $p_{i,j} \geq qmax$, then $q_{i',j'} = (q_{i',j'} + \Delta)$, for any $q_{i',j'} < p_{i,j}$,

where $\Delta = \min_{q_{i',j'} \neq p_{i,j}} \{ |q_{i',j'} - p_{i,j}| \}$

In this way, if $qmin$ for a block is larger than or equal to the gray level of the pixel $p_{i,j}$ responsible of the block, the computed gray levels of the pixels $q_{i',j'}$ are lowered. In turn, if $qmax$ for the block is smaller than or equal to the gray level of the pixel $p_{i,j}$, the computed gray levels are increased. In any case, the amount for lowering/increasing is given by the minimal gray level discrepancy in the block with respect to $p_{i,j}$. The effect produced by this updating is to reduce the differences between the gray level of $p_{i,j}$ and the computed gray levels of the pixels $q_{i',j'}$, while preserving the relative gray level differences among the pixels $q_{i',j'}$. As a result, the improved algorithm is preferable to the basic algorithm as far as PSNR is concerned. Moreover, from a visual point of view, the effect produced by the updating is a reduction of blurring with a somehow better edge delineation.

3 Performance Evaluation

The algorithm has been tested on a large number of images with different resolution and by using different zooming factors. Here, the performance evaluation of the proposed technique is illustrated with reference to the set of 15 original images with resolution 512×512 shown in Fig. 3. In particular, the original images are reduced to LR images with resolution 128×128 , to which the magnification procedure is applied with a zooming factor $n=4$ to obtain the reconstructed HR images.

Since the quality of LR images influences the quality of the reconstructed HR images, we use a decimation process that limits the presence of Moiré patterns in the LR images. To this aim, each pixel $p_{i,j}$ in LR is assigned a gray level equal to the average of the gray levels of the pixels in the block of the original 512×512 image from which $p_{i,j}$ is generated.

For each LR image, we compute the reconstructed HR images by our basic and improved algorithms, as well as by using nearest-neighbor, bilinear and cubic interpolation methods. To implement the latter methods, we have resorted to the standard OpenCV library. For each reconstructed image, the corresponding PSNR is computed to compare the original image to the reconstructed HR image. The PSNR is computed as follows:

$$PSNR = 20 \times \log_{10} \left(\frac{255}{\sqrt{MSE}} \right)$$

where $MSE = \frac{1}{H \times K} \sum_{i=1}^H \sum_{j=1}^K (v_{i,j} - w_{i,j})^2$ and $v_{i,j}$ and $w_{i,j}$ belong to the original image and to the reconstructed image of size $H \times K$, respectively.

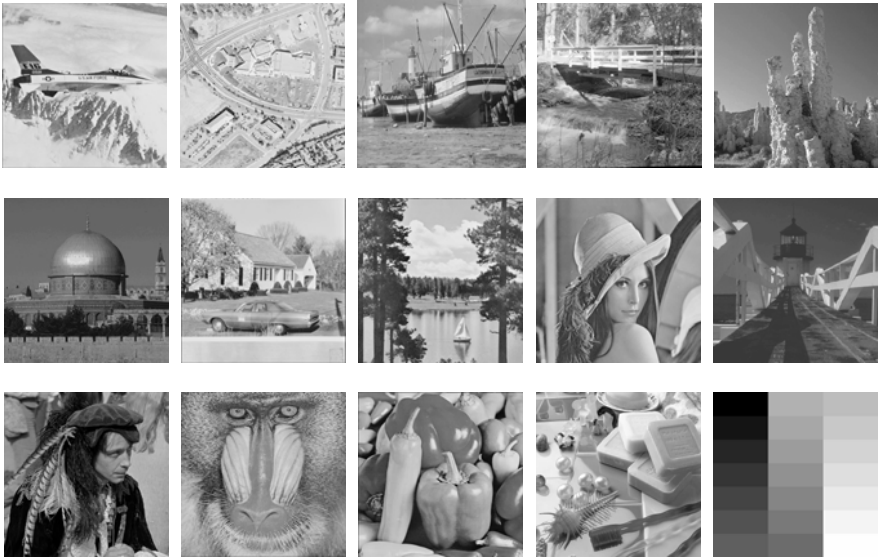


Fig. 3. Test images (512×512)

Table 1 summarizes the results. The highest PSNR value for each test image is in bold. The PSNR computed with the improved technique is always the highest one, except for the peppers image, for which the visual effect of the reconstruction obtained by using both bilinear interpolation and the improved algorithm is illustrated in Fig. 4.

Table 1. PSNR computed by using nearest-neighbor interpolation, bilinear interpolation, bicubic interpolation, the proposed basic algorithm and its improved version

	NN	BL	BC	Basic	Improved
airplane	24.693	25.485	23.158	25.215	25.526
aerial	21.714	22.117	20.348	21.885	22.268
boat	24.605	25.065	23.016	24.828	25.141
bridge	22.282	22.674	21.211	22.500	22.737
desert	22.760	22.854	21.917	22.734	22.907
dome	27.016	27.465	25.855	27.287	27.537
house	23.559	24.115	22.179	23.922	24.148
lake	23.630	24.437	22.070	24.153	24.446
lena	26.921	28.014	24.856	27.699	28.053
lighthouse	28.575	29.341	27.111	29.080	29.414
man	23.884	24.481	22.499	24.252	24.535
mandrill	20.849	20.887	20.095	20.778	20.962
peppers	25.999	27.194	24.418	26.957	27.152
soap	24.249	24.827	22.890	24.607	24.867
wedge	32.444	34.196	32.663	34.226	34.893



Fig. 4. The LR 128×128 input image, left, the 512×512 image reconstructed by bilinear interpolation, middle, and by the improved algorithm, right

The comparison between the two 512×512 images in Fig.4 shows that, notwithstanding the smaller PSNR, the visual effect of our reconstructed image is still good, since one can appreciate a diminished blurring and a better edge delineation. See Fig. 5, where a real size close-up of two details of the two images is given.

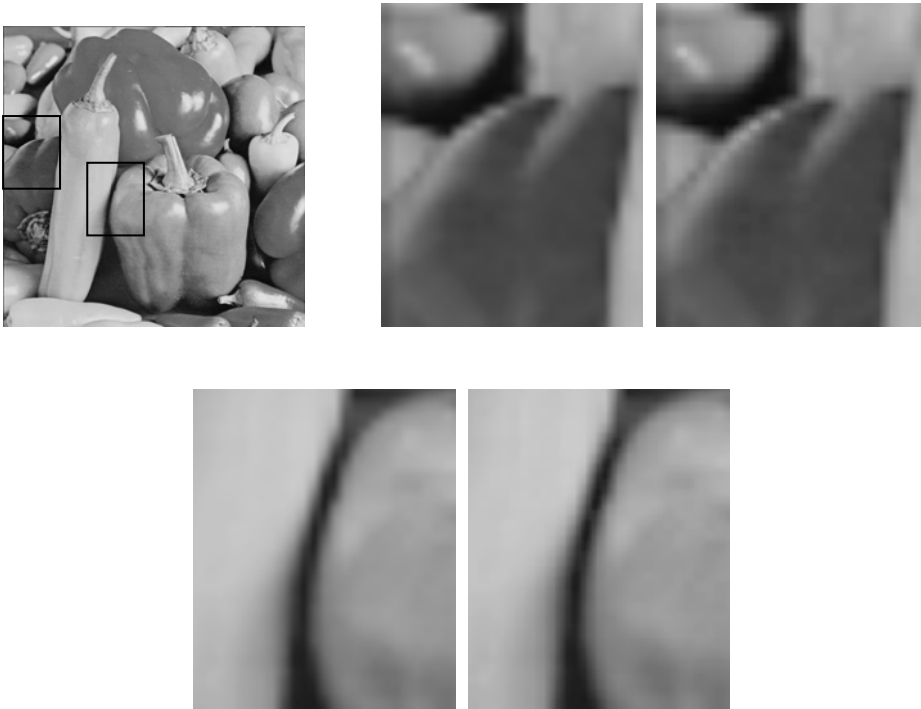


Fig. 5. The original 512×512 image, left, and a close-up of two details of the images, magnified from the 128×128 image, with bilinear interpolation, middle, and our technique, right

Though PSNR is widely used as a quantitative measure to evaluate the performance of magnification algorithms, we think that also qualitative ways should be taken into account. In fact, two reconstructed images with the same PSNR may have a different visual aspect, which is often important at least for display purposes. Thus, we have also computed the difference images, obtained by assigning to each pixel the absolute value of the difference in gray level pertaining that pixel in the original image and the reconstructed HR image. The difference images obtained by alternative magnification techniques are then compared to point out where discrepancies to the original images mainly occur. As an example, in Fig.6, we show two difference images obtained by bilinear interpolation and our improved algorithm. In both cases, for visualization purposes we have set to white all pixels for which the difference in gray level is at most equal to 25. We observe that the amount of pixels whose gray level in the HR images differs from the gray level in the original image is mostly the same in both cases and that differences concentrate along the edges.



Fig. 6. The difference image for the peppers image obtained by bilinear interpolation, left, and by our improved algorithm, right

The difference image can be used to extract a number of features that could provide an alternative quantitative measure to evaluate the performance of a magnification algorithm. Some of the features that can be extracted from the difference image are: the percentage of modified pixels, i.e., pixels whose gray level in the reconstructed HR image differs from the gray level in the original image; the average gray level change of the modified pixels; the ratio between the number of pixels with a given discrepancy (or the number of pixels whose discrepancy has the highest frequency) and the number of modified pixels.

Experiments have been carried out with different integer values for the zooming factor n . As a result, we note that the PSNR obtained when magnification is performed with our technique still results to be generally higher than the PSNR with the other interpolation methods, as it is the case for the zooming factor $n=4$. Of course, to compute the PSNR, the original 512×512 images have to be reduced to LR images whose resolution is $1/n$ of the corresponding original image. As expected, the PSNR of the reconstructed image diminishes when the zooming factor increases. Actually, this is only partially due to the increase in the zooming factor n ; in fact, the quality of the input LR image necessary to build the magnified HR image obviously worsens

when n increases. In turn, we note that the visual aspect is not significantly affected for zooming factors slightly larger than 4, if the same LR image is used as input to our magnification procedure; as an example, in Fig.7, the HR images reconstructed from the 128×128 LR image peppers with $n=5$ and $n=6$ are shown.



Fig. 7. Image reconstructed by the improved algorithm with $n=5$, left, and $n=6$, right

Finally, in Fig. 8 the magnification with a zooming factor $n=4$ of some particularly significant details of the original 512×512 lena image are given. We may note that staircase artifacts are not largely created (Fig. 8 left, portion of the hat and Fig. 8 middle, portion of the arm) in the presence of edges that are not aligned along the horizontal or vertical direction; we also note that thin image subsets with considerable variation of gray levels are reasonably well magnified (Fig. 8 right, feathers).

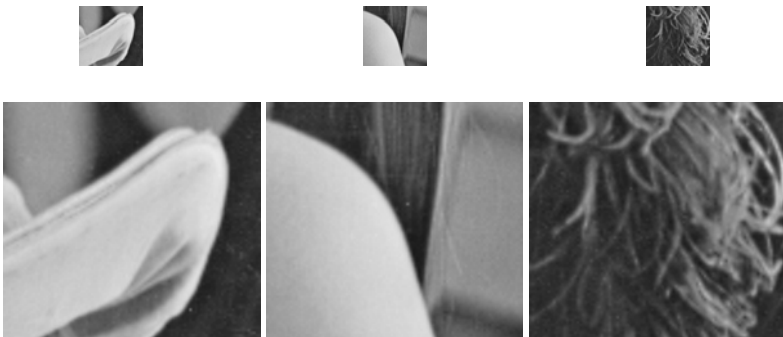


Fig. 8. Some particulars of the 512×512 original lena image, top, and their magnification with a zooming factor $n=4$, bottom

4 Concluding Remarks

We have introduced a discrete technique for image magnification, which is easy to implement and has a limited computational cost, since it requires only one scan of the LR input image. A relevant feature is that no threshold is necessary, so that the technique is fully automatic. Since the technique is discrete, only integer zooming factors can be used. Some blurring, especially with low quality input images, may occur. In turn, the visual aspect of the reconstructed image is appealing, when starting from a high quality low resolution input image, even if zooming factors larger than 4 are used.

Two versions of the algorithm have been presented. The basic algorithm has lower computational cost, while the results obtained by using the improved algorithm are characterized by a larger value of PSNR. The algorithms have been suggested for gray level images, but our approach can be extended to color images and work is currently in progress in this respect.

References

- [1] Lehmann, T.M., Gonner, C., Spitzer, K.: Survey: interpolation methods in medical image processing. *IEEE Trans. on Medical Imaging* 18(11), 1049–1075 (1999)
- [2] 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)
- [3] Schultz, R.R., Stevenson, R.L.: A Bayesian approach to image expansion for improved definition. *IEEE Trans. on Image Processing* 3(3), 233–242 (1994)
- [4] Carey, W.K., Chuang, D.B., Hemami, S.S.: Regularity preserving image Interpolation. *IEEE Trans. on Image Processing* 8(9), 1293–1297 (1999)
- [5] Mitra, S.K., Murthy, C.A., Kundu, M.K.: A technique for image magnification using partitioned iterative function system. *Pattern Recognition* 33(7), 1119–1133 (2000)
- [6] Chung, K.H., Fung, Y.H., Chan, Y.H.: Image enlargement using fractal. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 273–276 (2003)
- [7] Lai, C.M., Lam, K.M., Siu, W.C.: An efficient fractal-based algorithm for image magnification, proceedings. In: *Proc. Int. Symp. on Intelligent Multimedia, Video and Speech Processing*, pp. 571–574 (2004)
- [8] Tsai, A., Yezzi Jr., A.: Curve evolution implementation of the Mumford–Shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Trans. on Image Processing* 10(8), 1169–1186 (2001)
- [9] Tschumperle, D., Deriche, R.: Vector-valued image regularization with PDEs: a common framework for different applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 27(4), 506–517 (2005)
- [10] Thurnhofer, S., Mitra, S.: Edge-enhanced image zooming. *Optical Engineering* 35(7), 1862–1870 (1996)
- [11] Battiato, S., Gallo, G., Stanco, F.: A locally adaptive zooming algorithm for digital images. *Image and Vision Computing* 20, 805–812 (2002)
- [12] Zhang, X., Lam, K.-M., Shen, L.: Image magnification based on a blockwise adaptive Markov random field model. *Image and Vision Computing* 26, 1277–1284 (2008)