Real time artifact-free image upscaling

Andrea Giachetti and Nicola Asuni

***Abstract*—The problem of creating artifact-free upscaled images appearing sharp and natural to the human observer is probably more interesting and less trivial than it may appear. The solution to the problem, often referred to also as ”single image super-resolution”, is related both to the statistical relationship between low resolution and high resolution image sampling and to the human perception of image quality.**

In many practical applications, simple linear or cubic interpolation algorithms are applied for this task, but the results obtained are not really satisfactory, being affected by relevant artifacts like blurring and jaggies.

Several methods have been proposed to obtain better results, involving simple heuristics, edge modeling or statistical learning. The most powerful ones, however, present a high computational complexity and are not suitable for real time applications, while fast methods, even if edge-adaptive, are not able to provide artifacts-free images. In this paper we describe a new upscaling method (ICBI, Iterative Curvature Based Interpolation) based on a two step grid filling and an iterative correction of the interpolated pixels obtained by minimizing an objective function depending on the second order directional derivatives of the image intensity. We show that the constraints used to derive the function are related with those applied in another well known interpolation method providing good results but computationally heavy (i.e., NEDI [11]). The high quality of the images enlarged with the new method is demonstrated with objective and subjective tests, while the computation time is reduced of 1-2 orders of magnitude with respect to NEDI, so that we were able, using a GPU implementation based on the nVidia CUDA technology, to obtain real time performances.

I. INTRODUCTION

Image upscaling, or single image super-resolution has recently become a hot topic in computer vision and computer graphics communities due to the increasing number of practical applications of the algorithms proposed.

Image upscaling (and more generally image interpolation) methods are implemented in a variety of computer tools like printers, digital TV, media players, image processing packages, graphics renderers and so on. The problem is quite simple to be described: we need to obtain a digital image to be represented on a large bitmap from original data sampled in a smaller grid, and this image should look like it had been acquired with a sensor having the resolution of the upscaled image or, at least, present a ”natural” texture.
Methods that are commonly applied to solve the problem (i.e. pixel replication, bilinear or bicubic interpolation) does not fulfil these requirements, creating images that are affected by visual artifacts like pixelization, jagged contours, oversmoothing. For this reason a lot of improved algorithms have been presented in literature (see [17] for a review). They obviously rely on the assumption that, in natural images, high frequency components are not equally probable if low frequency components are known and a good algorithm is able to guess the image pattern that would have been created by a higher resolution sensor better than other methods.

The relationship between high resolution and low resolution patterns can be learned from examples and, for this reason, several researchers proposed to recover a statistical model of it from a training set. Approaches like those presented in [2], [16] try to classify patches according to the local edge appearance, applying different interpolation strategies depending on the results. More sophisticated techniques learn the correspondence between high resolution and low resolution image patches solving the problem of locally merging different results to generate a continuous output. Algorithms of this kind (example-based super resolution) can provide very good results (see, for example, [6], [9], [10]), even if they need a sufficiently representative set of examples. A possible way to avoid the use of training images has been proposed in [8] where patch recurrence in single images at different scales and with different subpixel alignment is used in a framework similar to classic multi-frame super-resolution.

Other problems of learning based approaches are related to the fact that the a priori information used is not usually valid for arbitrary scaling factors and to the fact that they are computationally expensive.

Realistic high frequency reconstruction is not the only issue to be considered in choosing an interpolation approach: the computational efficiency of the methods should also be taken into account, especially in the case of real time applications (i.e. to improve the perceived quality of video streaming).

Fast super resolution methods trying to obtain better results than simple polynomial interpolators are not usually based on statistical modelling, but simply adapt the local interpolation function to a low resolution estimate of local edge behaviour. Simplest edge-adaptive methods ([3], [4], [13]), that could easily reach real-time performances, are not, however, able to create natural looking images, and often introduce relevant artifacts.

On the other hand, more effective non-iterative edge-adaptive methods like NEDI (New Edge Directed Interpolation [11] or iNEDI (improved NEDI) [1], present a relevant computational complexity, even higher than that of many learning-based methods.

In this paper we propose a new image upscaling method able to obtain artifact-free enlarged images preserving relevant image features and natural texture. The method, as several edge-directed ones, approximately doubles the image size every

A.Giachetti is with the Department of Computer Science, University of Verona, Strada Le Grazie 15, 37134 Verona, Italy. e-mail: andrea.giachetti@univr.it.

N.Asuni is with Tecnick.com, Via Della Pace, 11, 09044 Quartucciu (CA), Italy

time is applied by putting original pixels in an enlarged grid then filling holes. The hole filling is done in two steps, linearly interpolating closest points in the direction along which the second order derivative of the image brightness is lower. After each hole filling step an iterative refinement is performed, updating the values of the newly inserted pixels by minimizing the local variations of the second order derivatives of the image intensity while trying to preserve strong discontinuities.
Other optimization based methods, with different constraints, have been proposed in literature. For example in [14] a gradient profile prior derived from the analysis of natural images and relating gradient profiles at different scales is used to enhance sharpness; in [5] a statistical dependence relating edge features of two different resolutions is forced; in [12] a constraint related to the smoothness of isophote curves is applied. In [15] the Gaussian Point Spread Function in the classical backprojection scheme is locally modified according to a local multiscale edge analysis.
These methods are often able to obtain good edge behavior, even if sometimes at the cost of texture flattening. The constraint used in our technique, based on the continuity of the second order derivatives (that we prove to be related to the NEDI constraint) is simple and extremely effective in removing artifacts; furthermore we use a smart initialization of the interpolated pixels using second order derivatives information that ensures a fast convergence, so that, implementing a GPU version of the algorithm with CUDA technology, we were able to obtain a real time high quality image upscaling.

The main contributions of our paper can be summarized in the following items:

- A review of constant covariance constraint used in the NEDI method with the proof of the relationship of that constraint with the second order derivatives smoothness used in our algorithm
- A new algorithm for image upscaling based on the iterative smoothing of second order derivatives (ICBI, Iterative Curvature-Based Interpolation). The algorithm is initialized using a simple filling rule based on second order derivatives (FCBI, Fast Curvature-Based Interpolation) that can be considered an edge directed interpolation algorithm too.
- A framework with test images and code for objective and subjective image quality evaluation. In our tests against similar edge directed methods, our method demonstrated to be an improvement over previous ones, providing a reasonable reconstruction of the missing information and requiring considerably less computational power than other methods achieving good scores. Images and scripts are available at the web site *http://www.andreagiachetti.it/icbi* so that other interpolation techniques can be compared with those presented here.
- A GPU implementation of the ICBI method able to enlarge images at interactive frame rates.

The paper is organized as follows: Section II gives the basic description of the particular class of image upscaling methods based on grid doubling and hole filling, Section III describes the NEDI algorithm, showing that some of its drawbacks can be removed by changing the constant covariance constraint with a more restrictive one, then Section IV demonstrates the relationship between this constraint and the hypothesis of second order derivatives continuity used in out new ICBI method. Section V describes the new method in detail and the experimental tests showing its advantages are reported in Section VI. The GPU implementation realized using the CUDA technology is described in Section VII.

II. Interpolation from 4 neighbors: fast methods and the NEDI algorithm

We focused our analysis on the ”edge-directed” interpolation algorithms that, each time they are applied, approximately duplicate the image size by copying original pixels (indexed by i, j) into an enlarged grid (indexed by 2i, 2j) and then filling the gaps with ad with ad hoc rules obtaining the missing values as weighted averages of valued neighbors, with weights derived by a local edge analysis. Algorithms of this kind are the well known Data Dependent Triangulation [13] and NEDI [11], but other similar techniques are, for example, described in [3], [4], [7].

In these methods the higher resolution grid is usually filled in two steps: in the first one, pixels indexed by two odd values (e.g. darker pixel in Figure 1 A) are computed as a weighted average of the four diagonal neighbors (corresponding to pixels of the original image); in the second the remaining holes (e.g. black pixel in Figure 1 B) are filled with the same rule, as a weighted average of the 4 nearest neighbors (in horizontal and vertical directions).

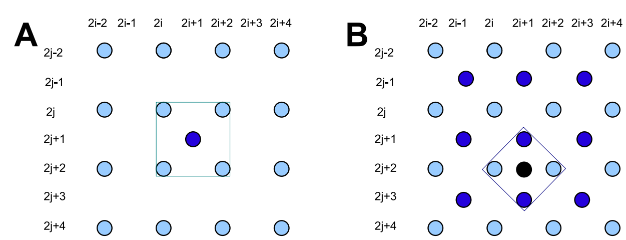

Fig. 1. Two steps interpolation based on a weighted average of four neighbors.

For example, for the first step, the interpolated value is usually computed as:

$$I_{2i+1,2j+1} = \vec{\alpha} \cdot (I_{2i,2j}, I_{2i,2j+2}, I_{2i+2,2j}, I_{2i+2,2j+2}). \quad (1)$$

and specific algorithms of this kind differ for the way they estimate the coefficients vector $\vec{\alpha} = (\alpha_0, \alpha_1, \alpha_2, \alpha_3)$ from the neighboring valued pixels in the grid.

In the Data Dependent Triangulation the weighted average is computed setting to zero the weights of the two diagonally opposite pixels that differs more among themselves, and to 0.5 those of the other two. In the NEDI method [11] the weights are computed by assuming the local image covariance (i.e. the vector α) constant in a large window and at different scales. With this constraint, an overconstrained system of equations can be obtained and solved for the coefficients. Images upscaled with this method are visually better than those obtained with the previously described methods, especially if some tricks are used to adapt window size and to handle matrix conditioning, as done in [1]. However, even applying the rule only in non-uniform regions and using instead a simple linear interpolation elsewhere (as actually done in [11], [1]), the computational cost of the procedure is quite high.

III. Constant covariance condition revised: a modified, well-conditioned NEDI

If we analyze the locally constant covariance assumption used in NEDI, we clearly see that it is not ideal to model a classical step edge profile. In this case the brightness changes only perpendicularly to the edge and it means that the overconstrained system solved to obtain the parameters is badly conditioned due to the rank deficiency of the problem (the expected rank of the matrix to be inverted is 2 and not 4). The simple solution we applied in [1] to avoid computational problems consists of finding the minimum norm solution using the pseudoinverse. Finding a different constraint leading to a well-conditioned problem would be, however, more satisfactory, as in the ill-conditioned case it would be possible to have a completely absurd pattern satisfying exactly the condition imposed to the local intensity.

We can obtain easily a better constraint by assuming that coefficients in α multiplying opposite neighbors are equal. In this case, we can write:

$$I_{2i+1,2j+1} = \vec{\beta} \cdot (I_{2i,2j} + I_{2i+2,2j+2}, I_{2i,2j+2} + I_{2i+2,2j}). \quad (2)$$

and, assuming that this relationship is true with the same coefficients in a neighborhood of the point and also at the coarser scale, we can, as in the NEDI algorithm, write an overconstrained system and solving it to find β_1 and β_2. In this case, the inverted matrix if full-ranked. The solution is clearly faster (about 35% in our experiments) and, most important, the quality of the interpolation is the same obtained with the NEDI method (see Section VI).

IV. NEDI constraint and the Iterative Curvature Based Interpolation

If the condition 2 holds in a neighborhood and across scales, it is reasonable to think that an algorithm iteratively refining interpolated pixels by locally minimizing a function that should be zero when the constraint is valid would be effective in obtaining a good result. From 2, we have:

$$\begin{aligned} &\beta_1(I_{2i,2j} - 2I_{2i+1,2j+1} + I_{2i+2,2j+2}) + \\ &\beta_2(I_{2i,2j+2} - 2I_{2i+1,2j+1} + I_{2i+2,2j}) = \\ &(1 - 2(\beta_1 + \beta_2))I_{2i+1,2j+1} \end{aligned} \quad (3)$$

One way to guarantee that this condition is locally true is to assume that local approximations of the second order derivatives along the two perpendicular directions, $(I_{2i,2j} - 2I_{2i+1,2j+1} + I_{2i,2j})$, $(I_{2i,2j+2} - 2I_{2i+1,2j+1} + I_{2i+2,2j})$ divided by the local intensity $I_{2i+1,2j+1}$ are constant. If we assume also that the local gain is null ($\beta_1 + \beta_2 = 1/2$), we can impose simply the constancy of the second order derivative estimates. This is actually the condition we introduced in our new ICBI (Iterative Curvature Based Interpolation) method.

The idea of ICBI is rather simple: in the two step filling method described in Section II, after the computation of the new pixel values with a simple rule (in our case we take the average of the two neighbors in the direction of lowest second order derivative, an algorithm we called FCBI, Fast Curvature Based Interpolation), we define an energy component at each new pixel location that is locally minimized when the second order derivatives are constant. We then modify the interpolated pixel values in an iterative greedy procedure trying to minimize the global energy. The same procedure is repeated after the second interpolation step.
Images obtained with this method do not present the evident artifacts; adding additional terms to reduce the image smoothing and heuristics to deal with sudden discontinuities, we obtained results that compare favourably with other "edge based" techniques, with a computational cost that is compatible with real time applications (see Section VII).

V. ICBI in details

Let us describe the algorithm in details. The two filling steps, as written before, are performed by first initializing the new values with the FCBI algorithm, i.e., for the first step, computing local approximations of the second order derivatives $\tilde{I}_{11}(2i+1,2j+1)$ and $\tilde{I}_{22}(2i+1,2j+1)$ along the two diagonal directions using eight valued neighboring pixels (see Fig. 2):

$$\begin{aligned} \tilde{I}_{11}(2i+1,2j+1) &= I(2i-2,2j+2) + I(2i,2j) + \\ &+ I(2i+2,2j-2) - 3I(2i,2j+2) - 3I(2i+2,2j) + \\ &+ I(2i,2j+4) + I(2i+2,2j+2) + I(2i+4,2j) \\ \tilde{I}_{22}(2i+1,2j+1) &= I(2i,2j-2) + I(2i+2,2j) + \\ &+ I(2i+4,2j+2) - 3I(2i,2j) - 3I(2i+2,2j+2) + \\ &+ I(2i-2,2j) + I(2i,2j+2) + I(2i+2,2j+4) \end{aligned} \quad (4)$$

and then assigning to the point (2i+1,2j+1) the average of the two neighbors in the direction where the derivative is lower:

$$\begin{cases} \frac{I(2i,2j)+I(2i+2,2j+2)}{2} & \text{if } \tilde{I}_{11}(2i+1,2j+1) < \tilde{I}_{22}(2i+1,2j+1) \\ \frac{I(2i+2,2j)+I(2i,2j+2)}{2}; & \text{otherwise.} \end{cases}$$

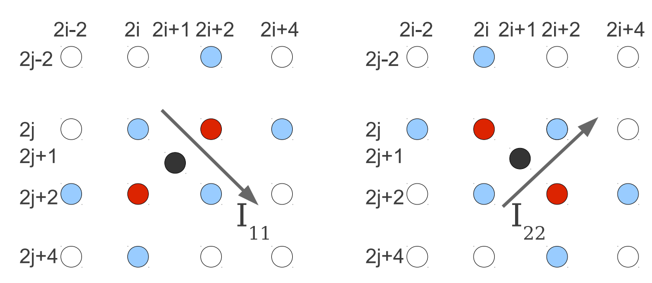

Fig. 2. At each step (here it is shown the first), the FCBI algorithm fills the central pixel (black) with the average of the two neighbors in the direction of lowest second order derivative (I_{11} or I_{22}). I_{11} and I_{22} are estimated using for each one the 8 valued neighboring pixels (evidentiated with different colors).

Interpolated values are then modified in an iterative procedure trying to minimize an ”energy” function. This function is obtained by adding a contribution for each interpolated pixel, depending on the local continuity of the second order derivatives and on other quantities that are minima when desired image properties are reached.

The sum of these pixel components should be minimized globally by varying the interpolated pixel values. It is clear that the computational cost of the procedure could be high. We apply, however, a greedy strategy just iterating the local minimization of each pixel term. Being the initial pixel value guess obtained with FCBI reasonable, the procedure leads quickly to a local minimum that appears to be reasonable for our task.

We said that the main energy term defined for each interpolated pixel should be minimized by small changes in second order derivatives. For the first interpolation step (filling gaps in the enlarged grid at locations $(2i+1,2j+1)$), we defined this term as:

$$\begin{aligned} U_c(2i+1,2j+1) = \quad & (5) \\ w_1(|(I_{11}(2i+1,2j+1) - I_{11}(2i+2,2j+2))| + \\ |(I_{22}(2i+1,2j+1) - I_{22}(2i+2,2j+2))|) + \\ w_2(|(I_{11}(2i+1,2j+1) - I_{11}(2i+2,2j))| + \\ |(I_{22}(2i+1,2j+1) - I_{22}(2i+2,2j))|) \\ w_3(|(I_{11}(2i+1,2j+1) - I_{11}(2i,2j+2))| + \\ |(I_{22}(2i+1,2j+1) - I_{22}(2i,2j+2))|) + \\ w_4(|(I_{11}(2i+1,2j+1) - I_{11}(2i,2j))| + \\ |(I_{22}(2i+1,2j+1) - I_{22}(2i,2j))|) \quad & (6) \end{aligned}$$

where I_{11}, I_{22} are local approximations of second order directional derivatives, computed as:

$$I_{11}(2i+1,2j+1) = I(2i-1,2j-1) + I(2i+3,2j+3) - 2I(2i+1,2j+1) \tag{7}$$

$$I_{22}(2i+1,2j+1) = I(2i-1,2j+3) + I(2i+3,2j-1) - 2I(2i+1,2j+1) \tag{8}$$

This energy term sums local directional changes of second order derivatives. Weights w_i are set to 1 when the first order derivative in the corresponding direction is not larger than a threshold T and to 0 otherwise. In this way smoothing is avoided when there is a strong discontinuity in the image intensity. Assuming that the local variation of the gray level is small, second order derivatives can also be considered an approximation of the intensity profiles curvature. This is why we call this term a ”curvature smoothing” term, and defined the algorithm ”Iterative Curvature Based Interpolation” (ICBI).

The optimization procedure minimizing the sum of the curvature smoothing terms is really effective in removing artifacts, but tends to create oversmoothed image. The smoothing effect can be only slightly reduced by replacing the second order derivative estimation with the actual directional curvature. In our experiments we found more effective the addition of another energy term enhancing the absolute value of the second order derivatives:

$$U_e(2i+1,2j+1) = -(|I_{11}(2i+1,2j+1)| + |I_{22}(2i+1,2j+1)|) \tag{9}$$

This term creates sharper images, but can introduce artifacts, so its weight should be limited. Another term we tested to reduce artifacts is related to isophotes (i.e. isolevel curves) smoothing. This is derived from [12], where an iterative isophote smoothing method is presented, based on a local force defined as

$$f(I) = -\frac{I_1(i,j)^2 I_{22}(i,j) - 2I_1(i,j)I_2(i,j)I_{12}(i,j) + I_{11}(i,j)^2 I_2(i,j)}{I_1(i,j)^2 + I_2(i,j)^2}$$

with $I_{11}, I_{22}, I_{12}, I_1, I_2$ being local approximations of first and second order directional derivatives. The related energy term we applied is:

$$U_i(2i+1,2j+1) = f(I)|_{2i+1,2j+1} I(2i+1,2j+1) \tag{10}$$

with I_{11}, I_{22} computed as before and

$$I_{12}(2i+1,2j+1) = 0.5(I(2i+1,2j-1) + I(2i+1,2j+3) - I(2i-1,2j+1) - I(2i+3,2j+1)) \tag{11}$$

$$I_1(2i+1,2j+1) = 0.5(I(2i,2j) - I(2i+2,2j+2)) \tag{12}$$

$$I_2(2i+1,2j+1) = 0.5(I(2i,2j+2) - I(2i+2,2j)) \tag{13}$$

Actually this term has a very small influence in improving the perceived and measured image quality.

The complete energy function for each pixel location $(2i+1,2j+1)$, sum of the ”curvature continuity”, ”curvature enhancement” and ”isophote smoothing” terms becomes therefore:

$$U(2i+1,2j+1) = aU_c(2i+1,2j+1) + bU_e(2i+1,2j+1) + cU_i(2i+1,2j+1) \tag{14}$$

Using this pixel energy, the first step of the iterative interpolation correction (adjusting pixel values with two odd indexes) is finally implemented as a simple greedy minimization as follows: after the placement of the original pixels at locations $(2i,2j)$ and the insertion of rough interpolated ones at locations $(2i+1,2j+1)$, we compute, for each new pixel, the energy function $U(2i+1,2j+1)$ and the two modified energies $U^+(2i+1,2j+1)$ and $U^-(2i+1,2j+1)$, i.e. the energy values obtained by adding or subtracting a small value

Fig. 3. Selected images from the test database.

δ to the local image value $I(2i+1, 2j+1)$. The intensity value corresponding to the lower energy is then assigned to the pixel. This procedure is iteratively repeated until the sum of the modified pixels at the current iteration is lower than a fixed threshold, or the maximum number of iterations has been reached. The number of iterations can be also fixed in order to adapt the computational complexity to timing constraints. In our implementation we change the value of δ from an initial value of 4 to the unit value during the iteration cycle in order to speed up the convergence. a, b and c and T were chosen by trial and error in order to maximize the perceived and measured image quality. Note that the value of c and T are not critical (if $T = I_{max}$ and $c = 0$) results are only slightly worse. If too large, the isophote smoothing term can introduce a bit of false contouring, flattening texture. The ratio between a and b determines a tradeoff between edge sharpness and artifacts removal. Actually, it may be also a reasonable option to use only the derivative-based constraint and to enhance contrast in post processing.

After the second hole-filling step (assigning values to all the remaining empty pixels), the iterative procedure is repeated in a similar way, just replacing the diagonal derivatives in the energy terms with horizontal and vertical ones and iteratively modifying only the values of the newly added pixels.

VI. Experimental results

We tested the algorithms proposed on a database of 25 natural images selected from the morgueFile online archive (http://morguefile.com). All images are subject to the license agreement available at the web page http://morguefile.com/archive/terms.php. For our experimental needs, we used images representing various objects, animals, flowers and buildings. These categories were chosen because they provide a wide range of colors and natural textures. Selected files were RGB color images with a depth of eight bits per channel. In all the previous equations we considered grayscale images; color images can be enlarged in the same way by repeating the procedures independently on each color channel or by computing interpolation coefficients on the image brightness and using them also for the other channels, reducing the computational cost and avoiding color artifacts.

The high quality of the images obtained with the new method can be clearly seen comparing the images upscaled of the same factor with different methods (see Figures 4,5). However, we also performed both subjective and objective tests in order to compare quantitatively the quality of the images created with different methods and the related computational cost.

A. Objective test

The objective test compares images obtained by downsampling the original images and then enlarging them with different methods, with reference images obtained just downsampling the original ones to the corresponding size. We performed this test on images converted to 8 bit grayscale, being the use of all three color channel not relevant to this test.

We created 128×128 and 256×256 subsampled version of the original images and the downsampled reference images. Different classes of algorithms required different reference images to compensate the slightly different zoom factors and translation created by the algorithms. Methods described in Section II do not enlarge exactly the images by $2\times/4\times$ factors, being the exact enlargement at each step equal to $(2width - 1)/width$ horizontally and $(2height - 1)/height$ vertically.

In any case, we applied the exact or approximate $2\times$ enlargement to the 256×256 images and the $4\times$ enlargement to the 128×128 ones. Finally we measured the differences between the upscaled images and the reference ones by evaluating the Peak Signal to Noise Ratio, defined as:

$$PSNR = 20 \log_{10} \frac{MAXPIX}{\frac{\sum_{i=1}^{W} \sum_{j=1}^{H} (I_{up}(i,j) - I_{orig}(i,j))^2}{(W*H)}} \qquad (15)$$

where $I_{up}(i,j)$ is the upscaled subsampled image, I_{orig} the original one, W and H the image dimensions and $MAXPIX$ the end scale value of the pixel intensity. The results for a $2\times$ enlargement of 256×256 images and $4\times$ enlargement of 128×128 images are summarized in Table I.

Algorithms tested are bicubic interpolation, and well known edge based methods, i.e. the technique described in [4], an iterative methods based on isophotes smoothing derived from [12], NEDI [11], the ”well conditioned” NEDI described in Section IV, the improved NEDI described in [1], and the FCBI and ICBI methods here described.

The choice of these algorithms (e.g NEDI-like and fast edge directed methods) is related to the focus of the paper (showing a fast algorithm related with the NEDI constraint). Any other method could, however, be tested with the same experimental setup presented here: all the images used and the evaluation scripts are available at the web site *http://www.andreagiachetti.it/icbi*.

Bicubic interpolation was obtained with the Matlab builtin function, the original NEDI Matlab code was kindly provided us by prof. Xin Li, Chen’s method was implemented by us in Matlab following the the description given in the cited paper. All the others algorithms have been coded in Matlab for this test.

The values obtained with the well conditioned NEDI (wNEDI) method are inserted just to show that the algorithm

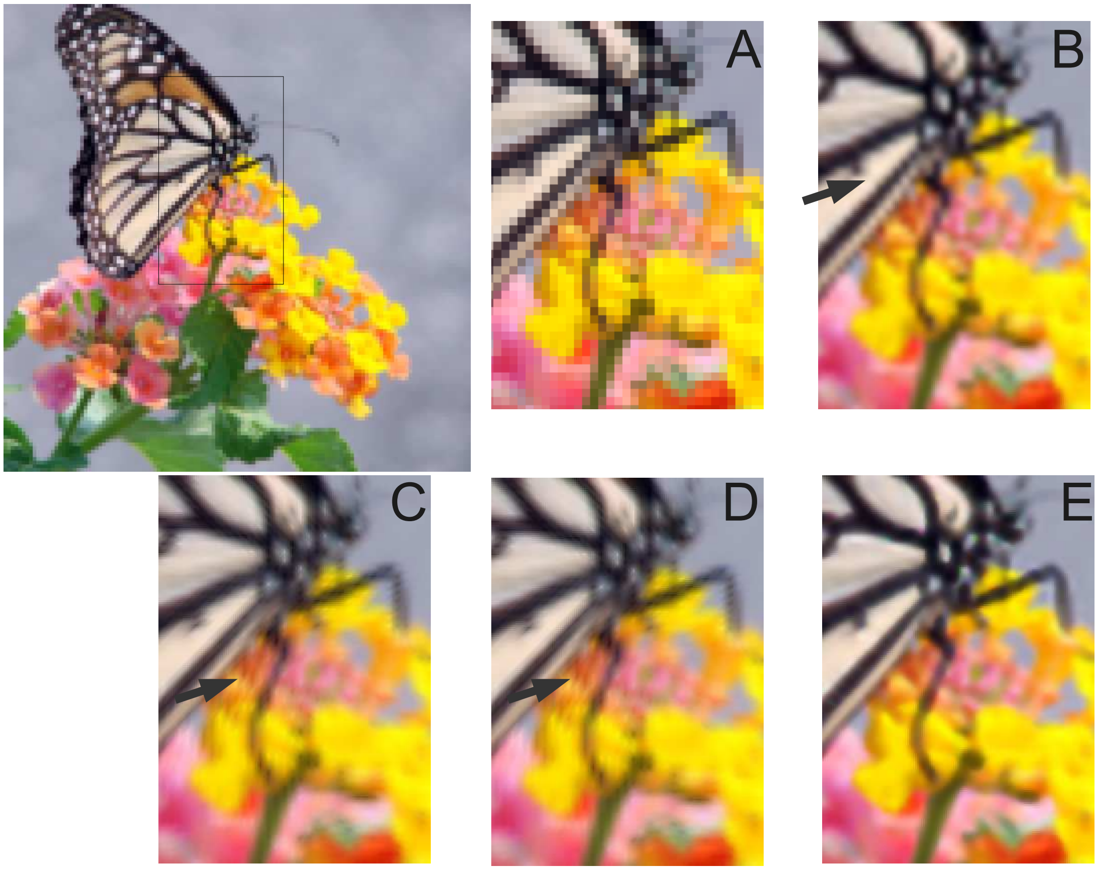

Fig. 4. The enlargement of a natural image using pixel replication (in this case by a $4\times$ factor), creates obviously a pixelized result (see detail in A). Bicubic interpolation (detail in B) removes this effect, but creates evident jagged artifacts (see near the arrow tip). Techniques like NEDI (C,D) provide better results (even if at the cost of a high computational complexity), but still create evident artifacts due to effect of edge discontinuities in the window used to estimate the covariance (see near the arrow tip). Images in C and D, that appears identical, are obtained with the original NEDI constraint and the modified constraint introduced in Section 3. The result obtained with the ICBI technique (E) does not present relevant artifacts.

Fig. 5. Comparison of different "edge directed" interpolation methods. A: Image detail enlarged ($4\times$) with pixel replication. B: the same detail enlarged by a $4\times$ factor with Chen's edge directed method. C: the same detail enlarged with NEDI method. D: the same enlargement obtained with our fast curvature based interpolation method (FCBI). E: the same enlargement obtained with our iterative curvature based iterative interpolation method (ICBI).

	ICBI	FCBI	iNEDI	wNEDI	NEDI	Chen	Iso.	Bic.	NN
$2\times$ dB	**31.07**	29.82	30.64	29.71	29.71	29.50	29.47	30.36	28.13
$4\times$ dB	**25.33**	24.46	25.18	24.30	24.30	24.19	24.11	24.91	23.43

TABLE I
PEAK SIGNAL TO NOISE RATIOS (dB) OBTAINED BY COMPARING IMAGES UPSCALED BY APPROXIMATELY $2\times$ AND $4\times$ FACTORS WITH REFERENCE IMAGES. THE ICBI METHOD PROVIDES THE BEST RESULTS.

indeed produces similar images with a lower computational cost (see Table II).

The iterative method proposed provides the best results. The accuracy is not much higher than iNEDI for $4\times$ enlargement, but if we consider the computation time (Table II) the new method appears clearly superior. Computation times reported in tables are obtained with non optimized Matlab implementations on a Dell XPS M1210 laptop with an Intel Core2 Duo T7200 2.0 GHz CPU.

	ICBI	FCBI	iNEDI	wNEDI	NEDI	Chen
$2\times$ time(s)	12.91	0.17	312.44	145.01	221.64	0.11
$4\times$ time(s)	13.30	0.18	372.83	189.97	293.36	0.12

TABLE II
AVERAGE COMPUTATION TIMES OBTAINED WITH NON-OPTIMIZED MATLAB IMPLEMENTATIONS OF THE ALGORITHMS.

The iterative nature of the ICBI algorithm can be also used to adapt image quality to the available hardware performances. In fact, by limiting the number of iterations we can obtain good quality and artifact-free images with a reduced computational cost. Table III shows the image quality (PSNR) and time performances obtained with the ICBI algorithm varying the maximum number of iterations. With just 5 iterations it is possible to have a PSNR close to the best one obtained. With more than 20 iterations the difference in PSNR becomes negligible.

	2 it.	5 it.	10 it.	20 it.	30 it.
2× time (s)	1.57	3.69	7.16	12.91	20.76
2× PSNR	30.62	30.93	31.05	31.07	31.07
4× time (s)	1.91	4.43	7.28	13.30	19.38
4× PSNR	24.92	25.23	25.31	25.33	25.33

TABLE III
BY FIXING A NUMBER OF ITERATIONS WITH THE ICBI ALGORITHM IT IS POSSIBLE TO MEET TIMING CONSTRAINTS AND STILL OBTAIN GOOD RESULTS. FIVE ITERATIONS ARE SUFFICIENT TO OBTAIN GOOD IMAGES.

Looking at the quantitative results, it must be also considered that some of the methods here tested are parameter dependent and it is possible to obtain different results changing parameters' values. We tried to optimize the results on for all the algorithms tested tuning parameters by trial and error (on a different image set), but slightly different results could surely be obtained with the different methods.

It should also be noted that optimizing methods in order to achieve maximum PSNR is not necessarily the best thing to do to have very good images, being PSNR not necessarily corresponding to visually perceived quality. For example, using ICBI we found that the PSNR values can be increased by adding more weight to the sharpening term, but at the cost of creating visible artifacts. We tested also different image quality measurements proposed in literature (e.g. those used in [17] or the mean structural similarity [19]) to try to have a better correlation between visible artifacts and quality measure, but we did not see relevant differences in algorithm ranking or in visual artifacts characterization. We preferred therefore to test the perceived quality by making experiments with human subjects.

B. Subjective test

In order to compare perceived image quality, we have taken a subset of 10 of the previously described RGB images and enlarged them by a $4\times$ factor with six different algorithms (NEDI, iNEDI, bicubic, Chen's, FCBI, ICBI). We then asked a group of 12 people to compare them, in order to select the method providing the best average ”perceived quality”. All the different possible couples of corresponding images were presented (in random order) to the subjects involved in the test, who were asked to choose the preferred image for each of them. An LCD display was used to represent image couples at full resolution on a screen surface of about 17×17 centimeters.

The sum of the successful comparisons for each interpolation method was then taken as the quality score of the method itself. The average scores (total number of preferences divided by the number of images multiplied by the number of subjects) are reported in Figure 6.

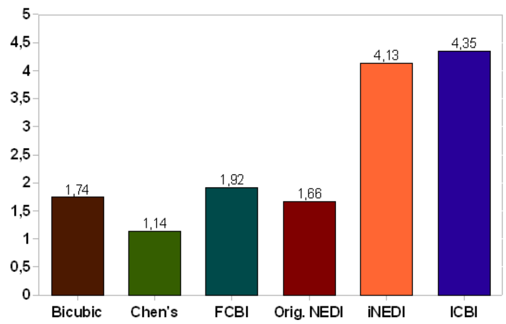

Fig. 6. Average qualitative scores obtained by a group of 12 people comparing pairs of differently enlarged ($4\times$) images and selecting for each pair the preferred one. The score of each algorithm is the average number of preferences of the algorithm on the 5 comparisons made by each person on each image.

We can observe some differences between the results obtained and the results of the objective test. FCBI is now preferred to bicubic interpolation, even if this one obtained better results in the objective test. The reason is that edge-directed interpolation methods are able to remove jagged artifacts that strongly affect bicubic interpolation.

The low number of subjects limits the statistical relevance of the test, but it is clear that images enlarged with ICBI and iNEDI appear evidently of higher quality than those interpolated with other methods compared. ICBI is usually preferred except when the image is not characterized by high frequency textures (causing the former method to create artifacts like those visible in Fig 5 C) and the user judgement of “high quality” is more related to sharpness than on artifact removal. This is somehow expected, we have, in fact, shown that the two methods are based on strongly related constraints. iNEDI is subject to artifacts when the second order derivatives are not continuous due to the assumption of covariance constancy in a large area.

C. Image sharpness and artifacts

Subjective tests reveals that quality scores should be analyzed with care, being the perception of image quality related to image contents and to different factors that may be weighted differently according to the user's needs. It has been shown (see [18]) that the decrease in the perceived ”image quality” is related to a linear combination of blurriness and artifacts, with higher weight given to blurriness (most people seem to prefer an increase in sharpness rather than a similarly noticeable artifact removal). This is probably one of the reasons why, for the enlargement of high resolution images for printing enhancement, photographers often use software that does not create natural detail or maximize similarity between high resolution patches and low resolution upsampled ones. Default options of professional photo zooming software usually strongly enhance contrast and straight lines, locally flattening texture. Of course, this is not necessarily a good choice if the enlarged bitmap should preserve detail recognition, realism and a correct depth perception from defocus.

Learning based methods are also able to reconstruct sharp detail at the risk of creating ”hallucinated” objects, and the perceived quality may be good or bad according to the fact that the detail is realistic or not in that position.

It seems, therefore, a reasonable statement to say that there is not an interpolation method that is ideal in any condition: the choice of the algorithm is largely dependent on the application.

The ICBI method proposed here is, in our opinion extremely effective in removing sampling artifacts, even if it does not enhance strongly lines and contrasted edges and results appear a bit oversmoothed. If the user wants to obtain images with less defocusing and enhanced contours, the final result can be, however, post-processed with sharpening filters to obtain a more contrasted image or clearer lines, without creating texture appearing too artificial or “painted” (see Figure 7).

The other good feature of the method here proposed is the low computational complexity that allowed us to obtain real time performances with a GPU implementation, as we will see in the next section.

VII. CUDA IMPLEMENTATION AND REAL TIME INTERPOLATION

CUDA is a technology developed by nVidia allowing programmers to write code that can be uploaded and executed in recent nVidia graphics cards, exploiting their massively parallel architecture in order to obtain a relevant reduction of the computing time. C++ developers can write particular functions called ”kernels” that can be called from the host and executed on the CUDA device simultaneously by many threads in parallel.

Using this technology, we implemented the ICBI algorithm by creating several CUDA kernels corresponding to the different steps of the algorithm. In this way computation performed in different blocks of the image can be executed in parallel, while the execution of the different steps is synchronized (see Figure 8). A first kernel creates the high resolution image from the low resolution one, a second fills odd pixels with the FCBI method, then two kernels computing derivatives and correcting the interpolated values are executed repeatedly. The second interpolation step is implemented in the same way, with a first kernel inserting new pixel values, and the iterative call of the two kernels computing derivatives and locally changing the interpolated values optimizing the energy function.

With this implementation, we obtained the $4\times$ enlargement of 128×128 color images in 16.2 ms on average, corresponding to a ideal frame rate of 62 frames per second and the $2\times$ enlargement of 256×256 images in 12.3ms on average using a nVidia GeForce GTX280 graphic card (240 cores) and obtaining the same image quality of the Matlab and C version of the code.

This example implementation clearly shows the possibility of applying ICBI for real time applications.

VIII. DISCUSSION

In this paper we discussed several issues related to the problem of creating high quality upscaled images from low resolution original data. First we showed that the well known NEDI method can be slightly modified removing the necessity of solving ill conditioned overconstrained systems of equations and obtaining the same image quality. Then we showed how the modified NEDI constraint is related to the constraint used in our new ICBI (Iterative Curvature Based Interpolation) technique. This technique uses mainly the assumption that the second order derivatives of the image brightness are continuous along the interpolation directions and is able to obtain very good results, especially for its ability of removing artifacts without creating ”artificial” detail, as proved by our objective and subjective tests. The new technique, based on a greedy minimization of an energy function defined at the interpolated pixel locations, is not computationally expensive like example based methods or the NEDI procedure and it is easily parallelizable. This allowed us to implement, exploiting the nVidia CUDA Technology, a version of the algorithm able to work at interactive frame rates on commodity graphics cards.

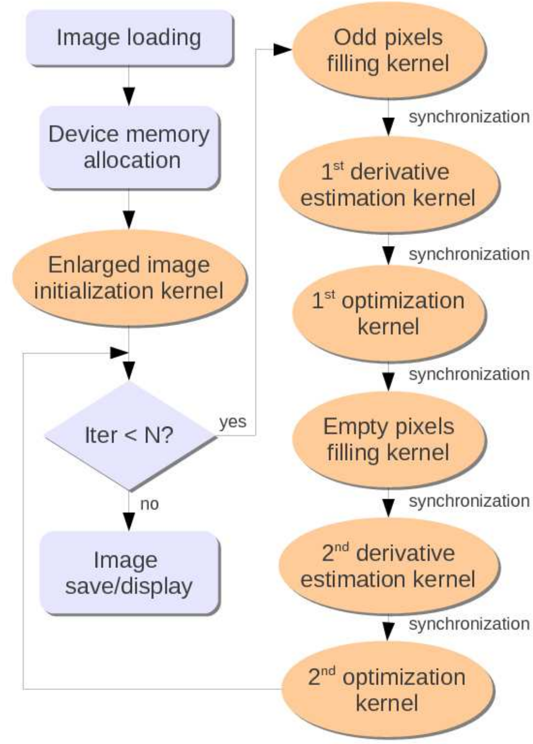

Fig. 8. Flow chart representing the execution of the CUDA ICBI implementation. Ellipses represent kernels where matrices are processed in parallel creating multiple threads each one processing a separate block.

ACKNOWLEDGMENTS

Special thanks to Enrico Gobbetti and José Antonio Iglesias Guitiàn for useful hints on CUDA technology and to Vittorio Murino for useful discussions.

REFERENCES

[1] N. Asuni and A. Giachetti. Accuracy improvements and artifacts removal in edge based image interpolation. In *Proc. 3rd Int. Conf. Computer Vision Theory and Applications (VISAPP’08)*, 2008.

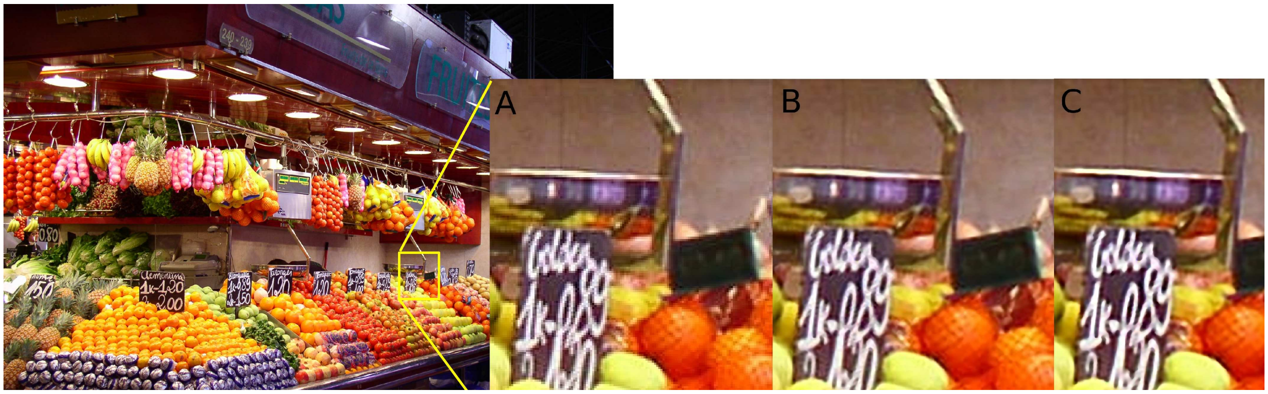

Fig. 7. $4\times$ upscaling of a 4 Megapixel image (not downsampled). A: Nearest neighbor enlargement showing a small detail at the original resolution. B: Same detail enlarged with ICBI: pixelization is removed without creating evident jaggies or artifacts, but the image appears oversmoothed. C: The same upscaled detail in B after a simple post-processing (selective smoothing and sharpening) enhancing the perceived quality of the printed image.

[2] C. B. Atkins, C. A. Bouman, and J. P. Allebach. Optimal image scaling using pixel classification. In *Proc. IEEE Int. Conf. Im. Proc.*, volume 3, pages 864–867, 2001.

[3] S. Battiato, G. Gallo, and F. Stanco. A locally-adaptive zooming algorithm for digital images. *Image and Vision Computing*, 20:805–812, 2002.

[4] M.J. Chen, C.H. Huang, and W.L. Lee. A fast edge-oriented algorithm for image interpolation. *Image and Vision Computing*, 23:791–798, 2005.

[5] R. Fattal. Image upsampling via imposed edge statistics. *ACM Transactions on Graphics*, 26(3):95, 2007.

[6] W.T. Freeman, T.R. Jones, and E.C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, 2002.

[7] A. Giachetti and N. Asuni. Fast artifact free image interpolation. In *Proc. BMVC 2008*, 2008.

[8] D. Glasner, S. Bagon, and Michal Irani. Super-resolution from a single image. In *proc. 12th International Conference on Computer Vision*, pages 349–356. IEEE, 2009.

[9] Kenji Kamimura, Norimichi Tsumura, Toshiya Nakaguchi, Yoichi Miyake, and Hideto Motomura. Video super-resolution using texton substitution. In *ACM SIGGRAPH 2007 posters*, page 63, New York, NY, USA, 2007. ACM.

[10] K. I. Kim and Y. Kwon. Example-based learning for single-image super-resolution. In *Proceedings of the 30th DAGM symp. on Patt. Rec.*, pages 456–465, Berlin, Heidelberg, 2008. Springer-Verlag.

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

[12] B.S. Morse and D. Schwartzwald. Image magnification using level-set reconstruction. In *Proc. IEEE Conf. Comp. Vis. Patt. Rec.*, volume 3, pages 333–340, 2001.

[13] D. Su and P. Willis. Image interpolation by pixel level data-dependent triangulation. *Computer Graphics Forum*, 23, 2004.

[14] J. Sun, Z.B. Xu, and H.Y. Shum. Image super-resolution using gradient profile prior. In *CVPR08*, pages 1–8, 2008.

[15] YW Tai, WS Tong, and CK Tang. Perceptually-inspired and edge-directed color image super-resolution. In *Proc.IEEE Conference on Comp. Vision and Patt. Recognition*, 2006.

[16] S. Thurnhofer and S.K. Mitra. Edge-enhanced image zooming. *Optical Engineering*, 35:1862–1870, 1996.

[17] J.D. van Ouwerkerk. Image super-resolution survey. *Image and Vision Computing*, 24:1039–1052, 2006.

[18] E. Vansteenkiste, D.Van der Weken, W. Philips, and E.E. Kerre. Evaluation of the perceptual performance of fuzzy image quality measures. In *proc. KES, 10th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems*, pages 623–630, 2006.

[19] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004.

PLACE PHOTO HERE

Andrea Giachetti received his Master's degree in Physics in 1993 and his Ph.D. degree in Physics in 1997 from the University of Genova, Italy. From Febrauy 1997 to January 2004, he was Expert Researcher at CRS4 (Center for Advanced Studies, Research and Development, Sardinia), where he became head of the Medical Image Processing group and was involved in several EU-funded research projects. From January 2004 to November 2006 he was Assistant Professor at the the University of Cagliari. Since November 2006 he has been Associate Professor at the Department of Computer Science of the University of Verona. His research activity is mainly focused on computer vision, computer graphics, medical image processing and shape analysis.

PLACE PHOTO HERE

Nicola Asuni received his Master's degree (Laurea) in Computer Science from the University of Cagliari in 2007. He is the founder and president of Tecnick.com S.r.l., a company providing IT consulting, implementation services and custom software development. He is author of several award-winning open source software projects.