ELSEVIER

Image and Vision Computing 20 (2002) 805–812

image AND vision COMPUTING

www.elsevier.com/locate/imavis

A locally adaptive zooming algorithm for digital images

S. Battiato, G. Gallo, F. Stanco*

Dipartimento di Matematica ed Informatica, Università di Catania, Viale A. Doria 6, Catania 95125, Italy

Abstract

In this paper we address the problem of producing an enlarged picture from a given digital image (zooming). We propose a method that tries to take into account information about discontinuities or sharp luminance variations while doubling the input picture. This is realized by a nonlinear iterative procedure of the zoomed image and could hence be implemented with limited computational resources. The algorithm works on monochromatic images, RGB color pictures and Bayer data images acquired by CCD/CMOS camera sensor.

Our experiments show that the proposed method beats in quality classical simple zooming techniques (e.g. pixel replication, simple interpolation). Moreover our algorithm is competitive both for quality and efficiency with bicubic interpolation. © 2002 Published by Elsevier Science B.V.

Keywords: Image processing; Zooming; Interpolation; Nonlinear DSP; Bayer pattern

1. Introduction

In this paper we address the problem of producing an enlarged picture from a given digital image (zooming). This problem arises frequently whenever a user wishes to zoom in to get a better view of a given picture. There are several issues to take into account about zooming: unavoidable smoothing effects, reconstruction of high frequency details without the introduction of artifacts and computational efficiency both in time and in memory requirements. Several good zooming techniques are nowadays well known [5, 7–10,14,19].

A generic zooming algorithm takes as input an RGB picture and provides as output a picture of greater size preserving as much as possible the information content of the original image. For a large class of zooming techniques this is achieved by means of some kind of interpolation; replication, bilinear and bicubic are the most popular choices and they are routinely implemented in commercial digital image processing software.

Unfortunately, these methods, while preserving the low frequencies content of the source image, are not equally able to enhance high frequencies in order to provide a picture whose visual sharpness matches the quality of the original image. The method proposed in this paper, similar to other published algorithms [3,6,19], tries to take into account information about discontinuities or sharp luminance variations while increasing the input picture. A brief mention to an adaptive technique is presented in Ref. [20] where a prior classification of pixel neighborhood is used to apply the most suitable interpolation strategy. Different from Ref. [20] our technique realizes a nonlinear multiple scan of the original image and could hence be implemented with limited computational resources.

The research of new heuristic/strategies able to outperform classical image processing techniques is nowadays the key-point to produce digital consumer engine (e.g. Digital Still Camera, 3G Mobile Phone, etc.) with advanced imaging application [2].

The basic idea of the new algorithm is to perform a gradient-controlled, weighted interpolation. The method speed up the entire adaptive process without requiring a preliminary gradient computation because the relevant information is collected during the zooming process. A preliminary and partial version of the results reported here has been presented in Ref. [1]. Although not linear, our technique is simpler and hence much faster than fractal-based zooming algorithms [14,17].

Our experiments show that the proposed method beats, in subjective quality, pixel replication and bilinear interpolation. Moreover our algorithm is competitive both for quality and efficiency with bicubic interpolation.

The rest of the paper is organized as follows. Section 1 provides a detailed description of the basic algorithm together with some considerations about its complexity. Section 2 describes how to adapt the proposed algorithm to color images. Section 3 reports the results obtained together with a detailed discussion about related performance and

* Corresponding author.
E-mail addresses: fstanco@dmi.unict.it (F. Stanco), battiato@dmi.unict.it (S. Battiato), gallo@dmi.unict.it (G. Gallo).

0262-8856/02/$ - see front matter © 2002 Published by Elsevier Science B.V.
PII: S0262-8856(02)00089-6

weakness. Section 4 shows an interesting extension of the proposed strategy able to work directly on Bayer data [4]. Also in this case experiments are reported and briefly discussed. Section 5 concludes the paper.

2. The basic algorithm

In this section we give a detailed description of the proposed algorithm. The procedure is summarized in pseudo-code in box 1. First we describe the algorithm in the case of gray scale picture. For sake of simplicity the magnifying factor applied here is two in both horizontal and vertical directions. The algorithm works in four successive stages as described in the following.

2.1. First stage: simple enlargement

The first stage is the simplest one and requires to expand the source $n \times n$ pixels image onto a regular grid of size $(2n-1)\times(2n-1)$ (Fig. 1). More precisely if $S(i,j)$ denotes the pixel in the ith row and jth column of the source image and $Z(l,k)$ denotes the pixel in the lth row and kth column in the zoomed picture the expansion is described as a mapping $E : S \rightarrow Z$ according to the equation:

$$E(S(i,j)) = Z(2i-1, 2j-1) \quad i,j = 1,2,\dots,n.$$

The mapping E leaves undefined the value of all the pixels in Z with at least one even coordinate (white dots in Fig. 1).

2.2. Second stage: filling the holes (part I)

In the second stage the algorithm scans line by line the pixels in Z whose coordinates are both even (e.g. the pixels denoted by a gray dot, labeled with an X in Fig. 2a). For reference, in the following description we will use the capital letters A, B, C and D as in Fig. 2a, to denote the pixels surrounding the pixel X and that have already been assigned a value in the previous stage (black dots). With letters a, b, c and d we denote the luminance value of pixels A, B, C, D, respectively.

For every X pixel one of the following mutual exclusive conditions is tested and a consequential action is taken:

- *Uniformity*: if $|\mathrm{range}(a,b,c,d)| < T_1$ the value $(a+b+c+d)/4$ is assigned to pixel X.

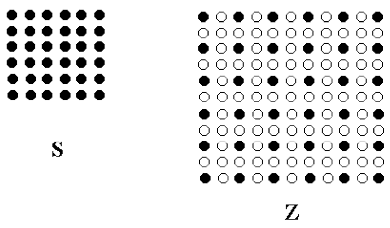


Fig. 1. The picture shows the first stage of zooming (enlargement).

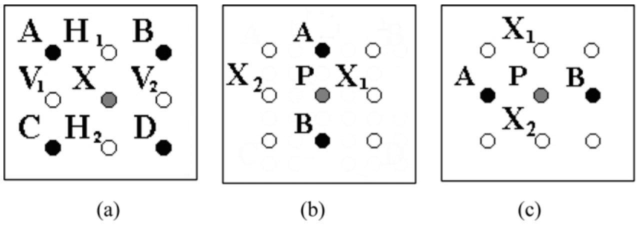


Fig. 2. (a) The layout of the pixel to reconstruct; (b) and (c) the layout and the notation referred in the description of the algorithm.

- *Edge in the SW–NE direction*: if $|a-d| > T_2$ and $|a-d| \geqslant |b-c|$ the value $(b+c)/2$ is assigned to pixel X.
- *Edge in the NW–SE direction*: if $|b-c| > T_2$ and $|b-c| \geqslant |a-d|$ the value $(a+d)/2$ is assigned to pixel X.
- *Edge in the NS direction*: if $|a-d| > T_1$ and $|b-c| > T_1$ and $(a-d)\times(b-c) > 0$ the value $(a+b)/2$ to pixel H_1 and $(c+d)/2$ to pixel H_2 and leave the value at pixel X undefined.
- *Edge in the EW direction*: if $|a-d| > T_1$ and $|b-c| > T_1$ and $(a-d)\times(b-c) < 0$ the value $(a+c)/2$ to pixel V_1 and $(b+d)/2$ to pixel V_2 and leave the value at pixel X undefined.

T_1, T_2 denote suitable threshold values that are described in Section 3. At the end of this stage there may be several X pixels, as well as several H_1, H_2, V_1 and V_2 pixels, left with an undefined value. Successive scanning of the zoomed picture will take care of these 'holes'.

```
Z= enlarge (E);                         // First Stage

For all pixel X in Z do                 // Second Stage
        Read ( A, B, C, D)
        If A, B, C, D are uniform then
                X=(a + b + c + d)/4;
        else if there is edge SW-NE then
                X=(b + c)/2;
        else if there is edge NW-SE then
                X=(a + d)/2;
        else if there is edge NS
                H1=(a + b)/2, H2=(c + d)/2;
        else if there is edge EW
                V1=(a + c)/2, V2=(b + d)/2;
        end if;
end For;

For all pixel P in Z do                 // Third Stage
        Read (A, B);
        if X1, X2 are assigned then
                if A, B are uniform then
                        P=(a + b)/2;
                end if;
        else if there is edge X1X2 then
                P=(x1 + x2)/2;
        else if there is edge AB then
                P=(a + b)/2;
        end if;
end For;

rebinning(Z);                           // Final Stage
end.
```

BOX 1: *A pseudo code description of the algorithm*

2.3. Third stage: filling the holes (part II)

In the third stage the algorithm scans line by line image Z looking for pixels left undefined in the previous stage and with at least one odd coordinate. These are the pixels denoted by the gray dot with label P in Fig. 2b and c. The two cases illustrated in the two sub-pictures will be treated similarly in the following discussion. For reference we will use the letters A and B as in Fig. 2b (or Fig. 2c), to denote the pixels above and below (left and right, respectively) of pixel P and that have already been assigned a value in the expansion stage (black dots).

Observe that the values of pixels X_1 and X_2 may, or may not, have been assigned in the previous stage. If they have been assigned they will be referred, respectively, with x_1 and x_2. Accordingly the algorithm considers the following two cases and take consequential actions:

- *X_1 or X_2 have not been assigned a value*. In this case the algorithm checks if $|a-b|<T_1$, with the same threshold T_1 introduced in the previous stage. If it is so, the value of the pixel P is set equal to $(a+b)/2$, otherwise the value of pixel P is left undefined.
- *X_1 and X_2 have both been assigned a value*. In this case the algorithm checks for the presence of a vertical or a horizontal edge. The following sub-cases arise:
  - Presence of an edge in direction X_1X_2. $|a-b|>T_2$ and $|a-b|\gg|x_1-x_2|$. In this case the value $(x_1+x_2)/2$ is assigned to pixel P.
  - Presence of an edge in direction AB. $|x_1-x_2|>T_2$ and $|x_1-x_2|\gg|a-b|$. In this case the value $(a+b)/2$ is assigned to pixel P.
  - None of above. No action is taken and the value of P is left undefined.

At the end of this third stage all the pixels whose spatial dependence from the neighborhood values is 'simple' have been assigned a value. Using the information gathered insofar, in the next stage the remaining 'holes' are eventually filled.

2.4. The final stage: rebinning

The final stage of the algorithm scans once more picture Z looking for pixels with undefined value and fix the hole using a suitably weighted mean value as described below. Preliminarily the gray value scale (typically ranging from 0 to 255) is subdivided into a rougher scale of $m=256/q$ bins B_i where q is a suitable, user selected integer value. The ith bin includes the gray values ranging from $q(i-1)$ up to $qi-1$.

The algorithm first looks for every pixel X, with both even coordinates and with an undefined value. In this case the pixels A, B, C, D surrounding X as in Fig. 2a fall within at least one and at most four bins B_j. The representative values of these bins (typically the median) are averaged to produce a value for X. Observe that in this way the more frequent values are weighted less: this 'trick' guarantees, as shown in the experiments, a better detail preservation in the zoomed picture. In order to preserve visual sharpness is fundamental reducing as much as possible the low-pass operation. It is also important to note that the values considered to assign a value to X are coming from the original values in the source and hence are not affected by the smoothing operations done in the three previous stages.

Eventually the algorithm looks for every pixel P, with at least one odd coordinate whose value is still left undefined. In this case the same rebinning procedure described for X pixels is performed starting from the values A, B, X_1, X_2 as in Fig. 2b and c. Observe that at this step, all four of these values have been already set.

The proposed approach requires $O(N)$ steps to zoom out of a factor 2 a digital picture of N pixels. As for space requirement the algorithm requires only the storage space for the zoomed picture.

The same complexity $O(N)$ steps is required by classical zooming algorithms where indeed the multiplicative constants are sensibly greater. This is particularly true, having a strong impact on the overall performance, when the dimensions of the input image are extremely large (e.g. 3/4 Mpixel acquired by a High Quality Digital Camera).

3. Zooming color pictures

The basic algorithm described above for gray scale pictures can be easily generalized to the case of RGB colored digital images. To do so we take advantage of the higher sensitivity of human visual system to luminance variations with respect to chrominance values. Hence it makes sense to allocate larger computational resources to zoom luminance values, while chrominance values may be processed with a simpler and more economical approach. This simple strategy is inspired by analogues techniques used by classical lossy image compression algorithms like JPEG and/or JPEG2000 [13,16,21] vastly implemented in almost digital still camera engines.

Accordingly we propose to operate as follows:

- Translate the original RGB picture into the YUV color model.
- Zoom the luminance values Y according with the basic algorithm described above.
- Zoom the U and V values using a simpler pixel replication algorithm.
- Back translates the zoomed YUV picture into an RGB image.

The results obtained with this basic approach are qualitatively comparable with the results obtained using bicubic interpolation over the three color channels. From the computational point of view, it is important to note how no significant difference in terms of timing response has been observed between the simple application of our approach to the three RGB planes and the approach described above (RGB–YUV conversion, Y zooming U, V replication, YUV–RGB conversion).

Yet, in real applications (DSC, 3G Mobile phone,…) the zooming process inside typical *Image Generation Pipeline* if present is realized just before compression [2]: the YUV conversion is always performed as a crucial step to achieve visual lossless compression. In this case the color conversion itself does not introduce further computational costs.

4. Experimental results

The validation of a zooming algorithm requires the assessment of the visual quality of the zoomed pictures. Fig. 3 shows two examples of zooming pictures obtained with the proposed algorithm. Unfortunately this qualitative judgment involves qualitative evaluation of many zoomed pictures from a large pool of human observers and it is hard to be done in a subjective and precise way. For this reason several alternative quantitative measurements related to picture quality have been proposed and widely used in the literature. To validate our algorithm we have chosen both the approaches proposed in Refs. [11,12,15], classical metrics and subjective tests. In particular we have used the cross-correlation and the PSNR between the original picture and the reconstructed picture to assess the quality of reconstruction.

In our experimental contest we have first collected a test pool of 30 gray scale pictures. For each image I in this set we have first performed the following operations:

- reduction by decimation: I_{d};
- reduction by averaging: I_{a}.

The size reduction technique adopted may have influences on the quality of the zoomed picture. Starting from I_{d} and I_{a} we have obtained the zoomed images listed as follows:

- I_{dr}, the picture I_{d} zoomed by a factor 2 using a simple pixel replication technique;
- I_{ar}, the picture I_{a} zoomed by a factor 2 using a simple pixel replication technique;
- I_{db}, the picture I_{d} zoomed by a factor 2 using bicubic interpolation;
- I_{ab}, the picture I_{a} zoomed by a factor 2 using bicubic interpolation;
- I_{do}, the picture I_{d} zoomed by a factor 2 using our algorithm;
- I_{ao}, the picture I_{a} zoomed by a factor 2 using our algorithm.

We have chosen simple replication and bicubic interpolation as the two comparing stones to assess the quality of our technique. It is generally accepted that replication provides the worst quality-wise zooming algorithm while bicubic interpolation is considered one of the best options available. It should also be observed that our technique requires the choice of two threshold parameters that could be in principle, user-selected. We have performed our tests with a fixed default setting of the thresholds ($T_1 = 4\ T_2 = 3$) These threshold values could be chosen in a more proper way using local adaptive measures of activity in a local neighborhood (e.g. local variance and/or local edge sensing [18]). Our heuristic values have been selected because they provided in preliminary experiments the best zooming quality. Preliminary experiments, moreover, have shown that small variations in these parameters do not produce large quality changes in the zoomed picture.

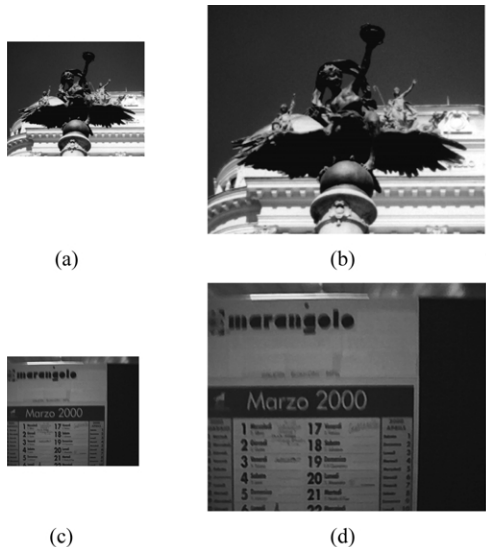


Fig. 3. Two examples of zoomed pictures obtained with the proposed algorithm. (a) and (b) relative to a natural scene, (c) and (d) relative to a printed text.

Let the cross-correlation coefficient C between two pictures A, B, be:

$$C = \left| \frac{\left(\sum_{k,l} A(k,l)B(k,l) - KLab\right)}{\sqrt{\left(\sum_{k,l} A^2(k,l) - KLa^2\right)\left(\sum_{k,l} B^2(k,l) - KLb^2\right)}} \right|$$

where a and b denote, respectively, the average value of picture A and B, K and L denote, respectively, width and length, in pixels, of images A and B. Notice that cross-correlation coefficients is between 0 and 1. The more the coefficient approaches 1, the better the reconstruction quality.

For every pair (I, I_{dr}), (I, I_{db}), (I, I_{do}), (I, I_{ar}), (I, I_{ab}),

Table 1

C_{do}	C_{db}	C_{dr}	C_{ao}	C_{ab}	C_{ar}
0.918775	0.899981	0.834393	0.97219	0.97386	0.97454

(I, I_{ao}) we have computed the relative cross-correlation coefficients c_{dr}, c_{db}, c_{do}, c_{ar}, c_{ab}, c_{ao}.

The coefficients have been averaged over the test pool to obtain the average coefficients: C_{dr}, C_{db}, C_{do}, C_{ar}, C_{ab}, C_{ao}. The values obtained are reported in Table 1.

The table promptly shows that the cross-correlation coefficients obtained with the proposed technique are better or very close to the analogous coefficients obtained using bicubic interpolation when the zooming is done from a decimated picture. This proves the better or equal ability of our algorithm to fill in with proper details the missing information in the larger picture. Setting to 0 the score for the pixel replication algorithm and to 1 the score for bicubic interpolation, the proposed technique scores a value of 1.286545.

The results in the case of a smaller picture obtained by averaging are essentially inconclusive because in this case the cross-correlation coefficient is not a good indicator of reconstruction quality. The results, indeed, seem to prize the pixel replication technique above the other two, even if it is very well known that this technique does not produce perceptually good images. This apparently contradictory result only comes from the statistical properties of cross-correlation coefficient. This statistical indicator turns out closer to one whenever a square of two time two pixels is substituted by a simple replication of their average. The behavior in the case of zoomed picture obtained from averaged smaller picture should hence be assessed using a different technique. To this aim we computed the percentage of zoomed pixels whose difference, in absolute value between the computed and the original intensity is greater than a given threshold.

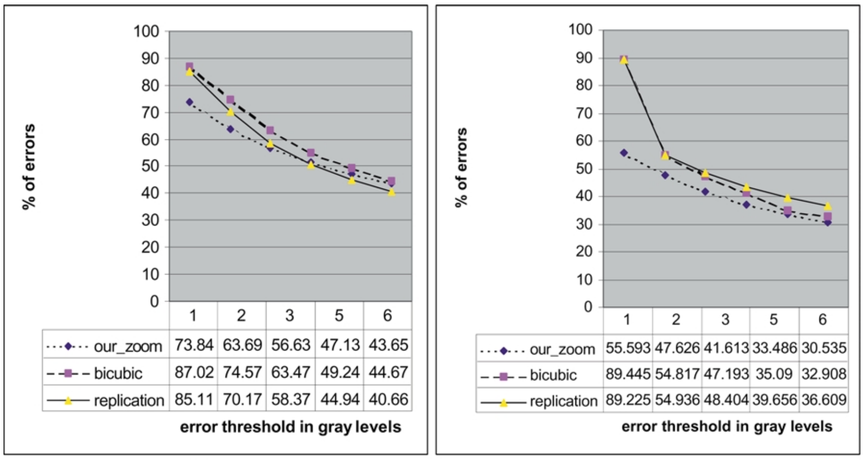
Table 2
PSNR values in Db measured over a test pool of digital images between original and zoomed images obtained with classical and proposed technique starting from image reduced by decimation and averaging

	Our	Replication	Bicubic
Decimation	23.66	19.95	24.77
Average	21.84	23.05	22.19

Fig. 4a shows the average percentage of errors observed over the test pool as different tolerance thresholds are considered in the case when the sub-sampled image is obtained by decimation. The proposed technique clearly outperforms the other two. Fig. 4b shows the average percentage of errors observed over the test pool as different tolerance thresholds are considered in the case when the sub-sampled image is obtained by local averaging. In this case as well the proposed technique gives a better performance.

Another classic quality measure we used in our validation experiments is the classical PSNR. In Table 2 are reported the average PSNR values in Db obtained from zoomed images reduced by averaging or decimation. In both cases our approach has PSNR values very close to the bicubic ones. Also in this case the pixel replication technique in terms of measured error gives good results.

The numerical results showed above does not give us a clear and objective comparative judgment with respect to the final quality of a zoomed images. It is also important to note that the perceived quality is not necessarily equivalent to fidelity (i.e. the accurate reproduction of the original). For example, sharp images with high contrast are usually more appealing to the average viewer. Indeed, the best way to valuate the visual quality of the zooming proposed algorithm is the subjective observation. In Figs. 5 and 6 are shown zooming results for three different images.



(a) (b)

Fig. 4. The average percentage of errors observed over the test pool for different tolerance thresholds when the sub-sampled image is obtained: (a) by local averaging; (b) by decimation.

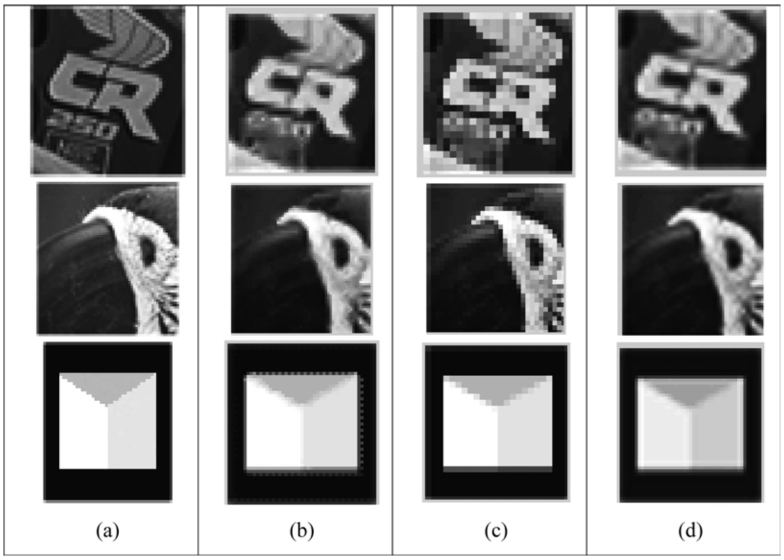


Fig. 5. (a) Ideal image; (b) our zooming; (c) replication zooming; (d) bicubic interpolation from averaging sub-sampling.

5. Working on Bayer data

The method has been generalized to work with Bayer data images [4], acquired by CCD/CMOS sensor in almost digital camcorder. Each pixel, using suitable CFA (Color Filtering Array) array, preserves the intensity of just one of the many-color separations. Notice that in a typical Bayer pattern, only one quarter of the array's pixels is red or blue and one half green. Green is chosen to have twice the number of pixels as red or blue because our eyes are most sensitive to green light. This filtering scheme allows us to capture color images, but since four pixels must be combined to form one color dot, the resolution of the image is less than a monochrome image. The reconstruction process must guarantee the rendering of a high quality images avoiding typical artifacts that, due to the acquisition process, could be present. For this reason powerful and smart algorithms are applied to enhance quality in a sort of chain known as Digital Still Camera pipeline (Fig. 7). To be able to work in the Bayer pattern domain allows one to manage 'noise-free' data and the complexity will be lower than working on RGB domain. Working in the Bayer domain requires a little effort to readapt ideas and techniques to the new particular environment but allows to improve significantly the quality of the final image reducing at the same time the associated computational complexity.

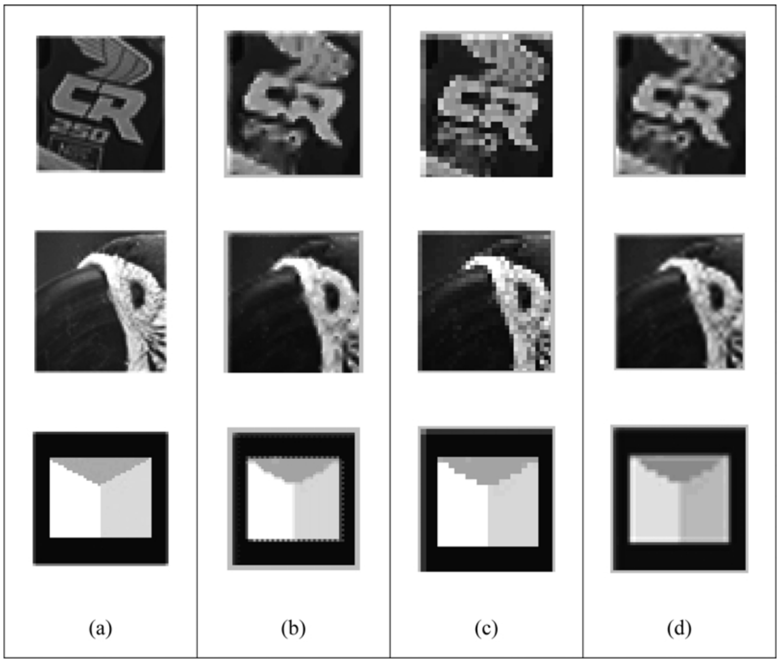


Fig. 6. (a) Ideal image; (b) our zooming; (c) replication zooming; (d) bicubic interpolation from decimation sub-sampling.

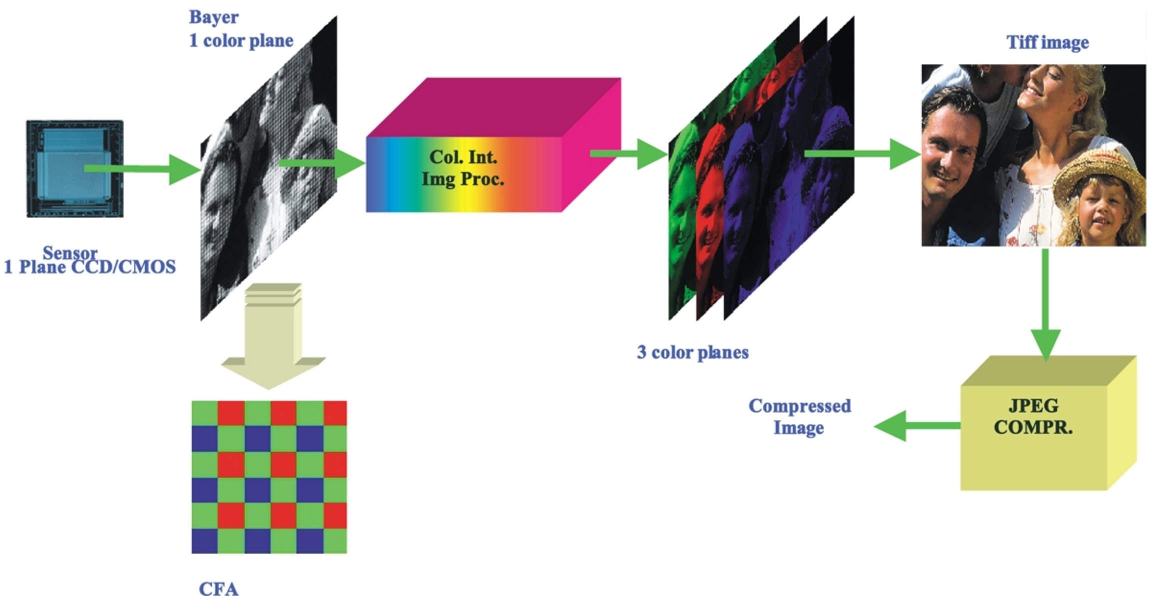


Fig. 7. Typical Digital Still Camera Pipeline.

In order to preserve the details of the original images, without introducing visible artifacts, the input $n \times n$ Bayer image is split into three sub-planes, R, G, B obtained retaining the corresponding chromatic component having, respectively, $n/2 \times n/2$, $n \times n/2$ and $n/2 \times n/2$ pixels. The proposed zooming algorithm is then applied, independently, for each one of these color-planes. Combining together these intermediate results a new zoomed Bayer data images is obtained. The image obtained is not an RGB image because the proposed algorithm is not applied as a color reconstruction algorithm. It simply enlarges an input Bayer pattern, obtaining a new zoomed Bayer data in a locally adaptive way preserving details without smoothing artifacts.

Fig. 8 shows two related examples: starting from two Bayer images we have obtained the corresponding zoomed Bayer images, successively interpolated with a simple color interpolation algorithm [18]. We claim that working directly in the Bayer domain, before color interpolation algorithm, is possible to improve further the quality of the final zoomed image. Further experiments must be done in order to be able to manipulate such images in a more proper way. Our preliminary results seem to suggest that it is necessary to retain information strictly related to the 'local' color distribution.

6. Conclusions

In this paper we have proposed a new technique for zooming a digital picture, both in gray scale, in RGB colors and in Bayer data domain. The proposed technique has been compared according to different performance indicators to bicubic interpolation and pixel replication algorithm. The experimental results show that the proposed method, while of not greater complexity than bicubic interpolation provides qualitatively better results.

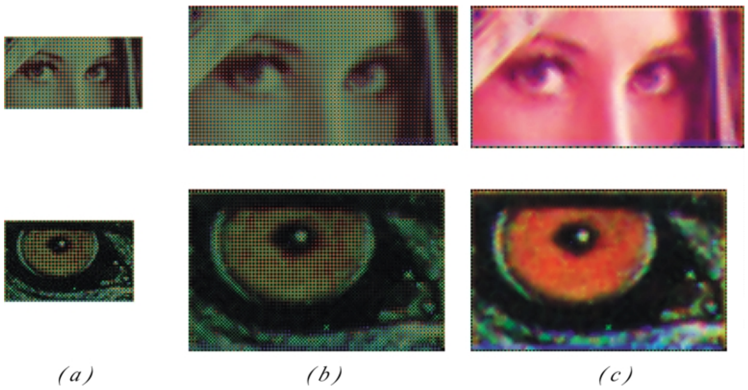


Fig. 8. (a) Original Bayer pattern; (b) zoomed Bayer pattern; (c) RGB zoomed image.

Acknowledgments

The authors are deeply grateful to Dr Massimo Mancuso of STMicroelectronics for providing motivation to this research and constructive critics to an early version of this algorithm. Thanks go also to the Digital Still Camera group of STMicroelectronics for providing some of the test pictures.

References

[1] S. Battiato, G. Gallo, F. Stanco, A new edge-adaptive zooming algorithm for digital images, Proceedings of Signal Processing and Communications SPC, Marbella, Spain (2000) 144–149.

[2] S. Battiato, M. Mancuso, An introduction to the digital still camera technology, ST Journal of System Research, Special Issue on Image Processing for Digital Still Camera 2 (2) (2001).

[3] S.D. Bayarakeri, R.M. Mersereau, A new method for directional image interpolation, Proceedings of International Conference on Acoustics, Speech and Signal Processing, Detroit, MI 24 (1995).

[4] B.E. Bayer, Color imaging array, US Patent 3,971,065-1976.

[5] D.F. Florencio, R.W. Schafer, Post-sampling aliasing control for images, Proceedings of International Conference on Acoustics, Speech and Signal Processing, Detroit, MI 2 (1995) 893–896.

[6] K.P. Hong, J.K. Paik, H. Ju Kim, C. Ho Lee, An edge-preserving image interpolation system for a digital camcorder, IEEE Transactions on Consumer Electronics 42 (3) (1996).

[7] H.S. Hou, H.C. Andrews, Cubic splines for image interpolation and digital filtering, IEEE Transactions on Acoustics, Speech, Signal Processing ASSP-26 (6) (1978) 508–517.

[8] A.K. Jain, Fundamentals of Digital Image Processing, Prenctice-Hall, Englewood Cliffs, NJ, 1989.

[9] R.G. Keys, Cubic convolution interpolation for digital image processing, IEEE Transactions. on Acoustics, Speech, Signal Processing 29 (6) (1981) 1153–1160.

[10] S.W. Lee, J.K. Paik, Image interpolation using adaptive fast B-spline filtering, Proceedings of International Conference on Acoustics, Speech, and Signal Processing 5 (1993) 177–179.

[11] T.M. Lehmann, C. Gonner, K. Spitzer, Survey: interpolation methods in medical image processing, IEEE Transactions on Medical Imaging 18 (11) (1999).

[12] E. Maeland, On the comparison of interpolation methods, IEEE Transactions on Medical Imaging MI-7 (1988) 213–217.

[13] M.W. Marcellin, M.J. Gormish, A. Bilgin, M.P. Boliek, An overview of JPEG-2000, Proceedings of IEEE DCC (2000).

[14] D.M. Monro, P.D. Wakefield, Zooming with implicit fractals, Proceedings of International Conference on Image Processing ICIP97 1 (1997) 913–916.

[15] J.A. Parker, R.V. Kenyon, D.E. Troxel, Comparison of interpolating methods for image resampling, IEEE Transactions on Medical Imaging MI-2 (1983) 31–39.

[16] W.B. Pennebaker, J. Mitchell, JPEG Still Image Data Compression Standard, Van Nostrand Reinhold, New York, NY, 1993.

[17] E. Polidori, J.L. Dugelay, Zooming using iterated function systems, NATO ASI Conference on Fractal Image Encoding and Analysis, Trondheim, Norway July (1995).

[18] T. Sakamoto, C. Nakanishi, T. Hase, Software pixel interpolation for digital still cameras suitable for a 32-bit MCU, IEEE Transactions on Consumer Electronics 44 (4) (1998) 1342–1352.

[19] P.V. Sankar, L.A. Ferrari, Simple algorithms and architecture for B-spline interpolation, IEEE Transactions on Pattern Analysis Machine Intelligence PAMI-10 (1988) 271–276.

[20] S. Thurnhofer, S. Mitra, Edge-enhanced image zooming, Optical Engineering 35 (7) (1996) 1862–1870.

[21] G.K. Wallace, The JPEG still picture compression standard, Communications of the ACM 34 (4) (1991).