ADAPTIVE LOW COMPLEXITY ALGORITHM FOR IMAGE ZOOMING AT FRACTIONAL SCALING RATIO

Nira Shezaf[†], Hagit Abramov-Segal[†], Ilan Sutskover[†] and Ran Bar-Sella[‡]

† SIPL – Signal and Image Processing Lab.
Department of Electrical Engineering - Technion
Technion City, Haifa, 32000, ISRAEL
E-mail: ilan@siglab.technion.ac.il

‡ Net2Wireless Research
Malat Bldg. , Technion
Technion City , Haifa 32000, ISRAEL
E-mail: barsella@net2wireless.co.il

Abstract

Interpolation of images to higher resolution is an important feature in many digital imaging applications ranging from image zooming for digital camera to image zooming in digital video. Practical digital imaging applications require the ability to scale an image in any fractional ratio, under low complexity/memory restrictions. Many traditional zooming algorithms are restricted to integer zooming ratios and yield poor results in edgy areas of the image; some are even restricted to powers-of-two ratios. Algorithms that achieve high quality are usually too complex to be implemented in low complexity/memory environments like VLSI. In this work we present a new adaptive algorithm for image zooming, suitable for fractional zooming ratios, motivated by the work of Ting. *et al.*[3]. The new algorithm performs better than known algorithms on a variety of images at both integer and fractional zooming ratios and is better suited for VLSI implementation.

Introduction

Interpolation of images from one resolution to another is usually application dependent. A video decoder is a good example. While the decoder works at a predefined resolution, its output is displayed at a resolution that depends on the application's specific TV standard. In many cases the required up-scaling factor is rational – and has the form of p/q.

Image interpolation is commonly performed by very simple methods such as pixel replication, bilinear or spline interpolation [1,2]. The visual results of these interpolation methods all suffer from unacceptable effects (e.g. blurring, aliasing, blockiness) to some extent, especially in edge areas of the image. Adaptive interpolation algorithms were developed to yield better quality results. However, most existing adaptive interpolation algorithms are designed for integer scaling ratios, or even powers-of-two scaling ratios. Another typical problem of most advanced algorithms is their high computational complexity and high storage requirements. In [4,5] interpolation techniques that are directed by a high-resolution edge-map are used. This edge-map is generated from the original low-resolution image. Other methods first find the local edge orientation and then interpolate along that direction [3,6,7].

Interpolation methods may be divided to the following categories:

1. Conventional interpolation methods that use constant convolution kernels for the entire image.

[*] This work was conducted at the Signal and Image Processing Lab, EE department, Technion. It was supported by Zoran Microelectronics Ltd. and is patent pending.

[‡] This work was conducted while the author was a research assistant at SIPL.

2. Adaptive methods that use edge information for the interpolation.
3. Interpolation methods that use Median filters.
4. Interpolation methods in a transformed domain, i.e., DCT.
5. Interpolation methods that exploit video motion compensation.

Only the first two approaches are suitable under restrictions like low complexity and/or low storage requirements.

The rest of this paper is divided as follows. In the next section the algorithm of Ting *et al.* [3] is presented together with its performance at fractional zoom ratios. Next, a new adaptive algorithm for fractional zooming of images is presented and its performance is demonstrated. A discussion concludes this paper.

The fuzzy based interpolation of Ting. *et al*

The algorithm suggested by Ting. *et al.*[3] is based on the notion of fuzzy sets. It tries to avoid the effects of both blockiness and smoothing by using locally adaptive interpolation. It uses a fuzzy inference approach in which the contribution of each of the neighbors of an interpolated pixel is determined using a Gaussian membership function. The membership function takes into account both the local gradients of the original image and the distance between the interpolated pixel and the pixels of the original image. Neighboring pixels effect the interpolated value via an exponential functional of the squared distances and the local gradients.

The interpolation phase described above does not yield good results on edges, hence a second stage is applied in [3]. This stage is performed on sharply curved edges located in the original image. First, edges are located in the original image and their direction is determined. Then, the new interpolated values are updated using linear interpolation along these edges.

When tested at fractional zoom factors (like 3:2), this algorithm gives very poor results. In most areas of the image the algorithm reduces to a simple pixel replication and blockiness is evident. In addition, the interpolated image suffers from an unpleasant ‘warping’ effect due to an unsymmetrical pixel replication caused by zooming at a fractional factor.

The complexity of the second stage of [3] is not suitable for low complexity/memory applications since it requires a stage of edge detection followed by interpolation along the detected edges.

While the algorithm in [3] was found to be unsuitable for our problem, the basic notion of using the local gradients still holds. In the next section a new algorithm is developed using this concept.

A new gradient based adaptive interpolation algorithm

Adaptive interpolation means that the way the neighboring pixels influence the value of the interpolated pixel depends on local properties. The interpolation problem is formulated in figure 1. Let x be a pixel to be interpolated. Let $\{a_1,..,a_4\}$ be the pixels of the original image which are closest to the new pixel. Denote by $p(a_i)$ the gray-level value of a pixel a_i in the original image. The horizontal distance from the new pixel x to pixel a_i is Δx_i and the vertical distance is Δy_i.

One basic notion, used in many interpolation methods, is that a pixel a_i contributes to the interpolate pixel x in a way which is proportional to its distance from x. This concept does not produce an adaptive interpolation rule

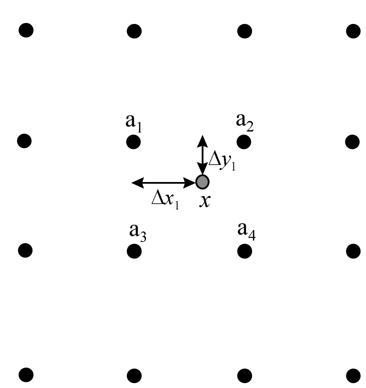


Figure 1: Image Interpolation scenario

since the way the original pixels are used is only determined by the grids of the original and the interpolate image.

Adaptive capability can be achieved in many ways. One way of utilizing gradients, mentioned in [3], relies on the following observation. Assume that an interpolated pixel lies exactly between two pixels, one in a flat region and one on an edge. It is reasonable to expect that the interpolated pixel should resemble the pixel in the flat area more than the other pixel. In essences the smaller the local gradient of a pixel a_i is, the more influence it should have on the interpolated pixel. Figure 2 demonstrates a one-dimensional case. Although the interpolated pixel x is closer to the original pixel a_2 it is clear that it should be closer in value to a_1 since a has a higher local gradient and is evidently on an edge.

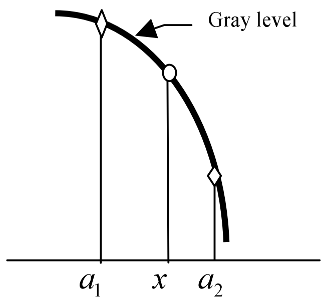


Figure 2: Local gradient and Interpolation

The new proposed algorithm uses a weighted average of the four neighboring pixels of x, $\{a_1,..,a_4\}$. The weights depend on both the distance of each pixel from the interpolated one and on the local gradients.

For each of the pixels a_i the local gradient is evaluated by averaging two (normalized) Sobel masks of size 3 (one horizontal and one vertical) to produce:

$$G(a_i) = \frac{|f'_x(a_i)| + |f'_y(a_i)|}{2}. \quad (1)$$

The local gradient G is between zero and one. Next the local gradient is used to calculate the gradient weight function:

$$W(a_i) = (-\mu G(a_i) + 1)^n. \quad (2)$$

Where n is a positive value, and μ is a positive value close to and lower than 1.

The gradient weight function is such that for lower gradients the weight is closer to 1 while for higher gradients it approaches zero.

For each of the pixels a_i the distance function $D(a_i)$ is calculated using the vertical and horizontal distances:

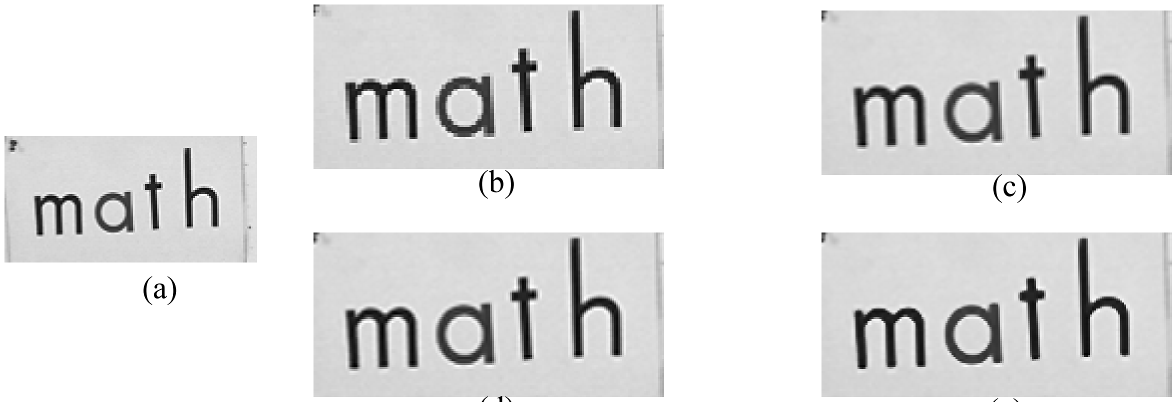
$$D(a_i) = (1 - \Delta x_i)(1 - \Delta y_i). \quad (3)$$

Note that this distance function is different from the one used in [3] and is better suited to cases where there are different Δx_i for different a_i's, as is the case at fractional zoom ratios.

The gradient weight function (2) and distance function (3) are then combined to form the interpolated pixel value in the following manner:

$$p(x) = \frac{\sum_i D(a_i)W(a_i)p(a_i)}{\sum_i D(a_i)W(a_i)}. \quad (4)$$

As an example, consider figure 3 where an image that contains text is interpolated. Text in an image presents a great challenge to interpolation algorithms since it has many edges and unlike textures, bad interpolation results are immediately evident.



(a) (b) (c) (d) (e)

Figure 3: Interpolation of an image at 3/2 up-scaling ratio using several methods:
(a) original image (b) pixel replication; (c) linear interpolation; (d) Cubic-Spline interpolation; (e) The new Gradient Based Adaptive Interpolation algorithm

Note that the edges in the magnified image are sharp yet lack jaggedness, as in the original image. Simulations at arbitrary integer interpolation factors demonstrate that the algorithm outperform both spline based algorithms and adaptive ones.

Conclusion

A new adaptive interpolation algorithm based on local gradients was presented. The new algorithm was tested at various interpolation factors, integer and fractional, and was found to yield better results than previously suggested algorithms.

The new algorithm suits VLSI implementations since it is local in nature and has modest requirements in both memory and complexity.

Future work in this area should address the extension of the algorithm to color images and further extend the adaptive capability by changing some of the parameters locally.

References

[1] J. A. Parker, R. V. Kenyon and D. E. Troxel, "Comparison of Interpolating Methods for Image Resampling", *IEEE Trans. on Med. Imaging*, 1983, Vol. 2, No. 1, pp. 31-9, Mar. 1983.

[2] N. A. Dodgson, "Quadratic Interpolation for Image Resampling", *IEEE Trans. on Image Processing*, Vol. 6, No. 9, p. 1322-6, Sept. 1997.

[3] H. C. Ting and H. M. Hang, "Spatially adaptive interpolation of digital images using fuzzy inference", *Proceedings of the SPIE*, Vol. 2727, pt. 3, pp. 1206-17, March 1996.

[4] J. Allebach and P.W. Wong, "Edge-Directed Interpolation", *Proceedings of ICIP-96*, IEEE Press, Lausanne CH, Vol. III, pp. 707-10, 1996.

[5] Biancardi, L. Lombardi and V. Pacaccio, "Improvements to image magnification", *Proceedings of ICIAP '97*, Vol. 2, pp. 142-9, 1997.

[6] S. Thurnhofer and S. K. Mitka, "Edge-enhanced image zooming", *Optical Engineering*, Vol. 35, No. 7, pp. 1862-70, SPIE Proceedings of the European Signal Processing Conference, Grenoble, France, pp. 1445-8, July 1996.

[7] F. Michaud, C. T. Le Dinh and G. Lachiver, "Fuzzy Detection of Edge-Direction for Video Line Doubling", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, No. 3, pp. 539-42, June 1997.