Efficient Algorithm for Enhancement Images Quality across Distributed Systems

Ameer A. M. Al-Shammaa
Information Technology Research
and Development Center
University of Kufa, An Najaf, Iraq

Ebtesam Najim Abdullah AlShemmary, Ph. D
Information Technology Research
and Development Center
University of Kufa, An Najaf, Iraq

ABSTRACT

Due to the large development in the use of distributed systems applications many problems emerged. We have discussed in this paper one of the most important problems that are faced when using distributed systems to send data, the distortions that occurs while image is being sent where these distortions caused reduced color accuracy and clarity of images. This paper introduced a very efficient algorithm to process the problem of distortions that occur with images and this algorithm is very fast to images interpolation. The basic idea of this algorithm is digital image segmentation into homogeneous areas and edges, depending on the structural analysis of the images. In addition, in order to get the best performance of images interpolations, algorithms used are customized to interpolate area classified images, respectively. The experimental results showed a significant improvement in the quality of interpolation of images compared with the interpolation of images by using the proposed algorithm. The comparison results of complexity calculations of the proposed algorithm are shown in this work.

Keywords

Image processing, Distributed System, Image Enhancement, Image Quality

1. INTRODUCTION

In the recent years the need has grown for the use of distributed systems and increased this need with the development of technology in all fields, and applications of distributed systems began used for different fields.

In reality, the Web represents a huge distributed system that appears as a single resource to the user available at the click of a button [1]. There are several definitions and view points on what distributed systems are. Coulouris defines a distributed system as “a system in which hardware or software components located at networked computers communicate and coordinate their actions only by message passing” [2]; and Tanenbaum defines it as “A collection of independent computers that appear to the users of the system as a single computer” [3].

Various types of distributed systems and applications have been developed and are being used extensively in the real world. In this article, we present one of the main Applications of distributed systems that is Multimedia communication field, Image transfer is one of the main applications in multimedia communication. Efficient manipulation of large amount of image data in a system with limited bandwidth is considered as a key issue in this process. Where in this paper we propose a web-based architecture to use to send and receive the images across the internet, by using the most common type of the distributed computing architecture that is a client / server model, which includes a high performance algorithm for Enhancement Images Quality that will send or receive across the internet in the above model, because when we request the images from the server side by using the internet may be the images it will arrive to the client side contain some distortions or unclear in terms of color accuracy, So we will use efficient algorithm to improve the images quality, and we will put this algorithm in the client side.

Digital images and video sequences essentially result in huge volume of image data. With huge volume of image data and limitation of network bandwidth, the quality of the transferred images is often unacceptable and poor performance, such as jitters and flickers can occur. The bandwidth for image transfer can be saved much more and image quality can be improved if the low-resolution video sequences is coded in encoder end and the low-resolution video sequences is afterward enlarged to high-resolution ones using interpolation techniques in decoder end. Image interpolation can be used in image enlargement and local image zooming. Several commonly used interpolation algorithms have been suggested. The basic concept of the interpolation algorithms is to interpolate images using the feature of pixels.

2. RELATED WORK

In the previous section, we explain what is the errors that are accrue on the images when it sent over the distributed systems, and how we can solved by using the interpolation algorithms, We will discuss some of the previous work on the same area of our research, what are the weaknesses in the previous work, what is the difference between them and the solution provided by us.

As we said in the previous section, Image interpolation can be used in image enlargement and local image zooming. Several commonly used interpolation algorithms have been suggested, such as zero-order interpolation [4], linear interpolation [4] and cubic convolution interpolation [5]. However, image artifacts like blurring or zigzag on edge may occur when these interpolation schemes are used. In order to reduce the effect of image artifacts, other algorithms have been proposed [6–13]. The method proposed in [6] is based on directions in spatial domain. Interpolated values along different directions are calculated using directional weights. The weights are dependent on the variation in the directions. The algorithms in [7–8] are convolution-based interpolation methods. Pixels to be interpolated are classified into two decimations in which pixels are interpolated by different specified filters. A hybrid of convolution and median-based scheme splits the interpolation into two directional stages [9]. The soul of the algorithm in [10] is based on variation models with smoothing and orientation constraints. The algorithm in [11] is applied to an image after it was expanded using either bilinear or bicubic interpolation. Edges in the expanded image are obtained by use of canny edge detector.

The values of pixels around the edges are modified to yield a crisper and less zigzagged picture. Another algorithm of adaptive re-sampling analyses the local structure of an image and applies a near optimal and least time-consuming re-sampling function that can preserve edge locations and their contrast [9]. Algorithms described above [6–12] have better performance than common interpolation algorithms introduced in [4–5]. Another robust algorithm, New Edge-Directed Interpolation (NEDI), performs good subjective quality for image enlargement. The method initially estimates local covariance coefficients from a low-resolution image and then the estimated covariance are used to adapt the interpolation at a higher resolution based on the geometric duality between the low-resolution covariance and the high-resolution covariance. By comparing to other algorithms, the algorithm of covariance-based interpolation has higher computational complexity [13].

The basic concept of the algorithms mentioned above is to interpolate images using the feature of pixels. Determination of pixel feature by these methods needs higher computational complexity. Consequently, these algorithms are unable to achieve real-time (30 frames/sec) image enlargement in video sequences. To solve the problem, we propose a new interpolation algorithm, which provides better subjective quality and lower computational complexity for image enlargement.

3. PROPOSED WEB-BASED ARCHITECTURE

In this section we explain the proposed architecture for our system, that system contains two subsections: the Client side and the Server side.

In the client: We will request the image from the Server that is represent the data center and use to store the data by send Http request, after this operation the server will send reply to this request to the client, When the client receive the reply from the server, the client will apply the enhancement algorithm on the images by the two step as that shown in figure (1).

In the server side: The server will receive the request from the client by Http request and search in the server database on the request image, then the server will reply the client by send the Http reply that contain the requested image.

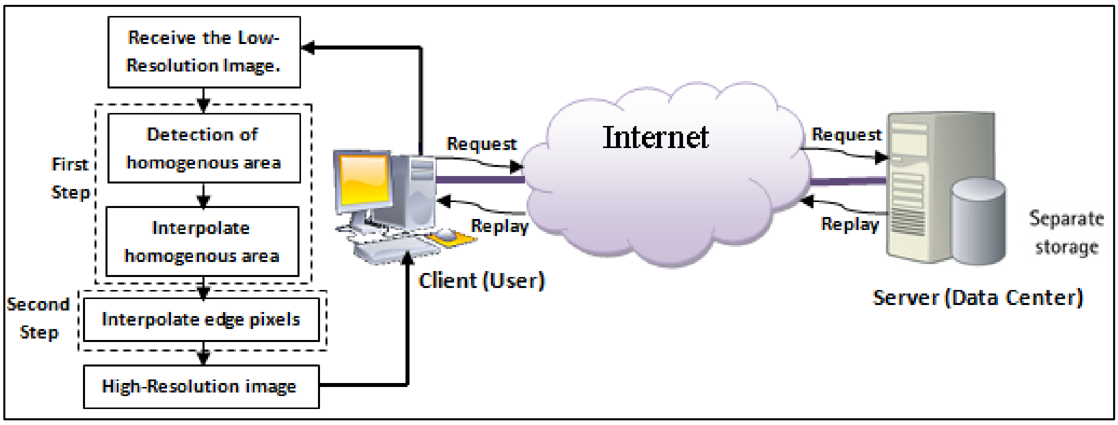


Figure 1: The proposed Web-Based Architecture

4. THE PROPOSED ENHANCEMENT ALGORITHM

The proposed algorithm is expected to achieve two major goals: *lower computational complexity* and *better subjective quality*. Hence, the new method can be practically applied to video sequences and videoconference [14].

Our proposed method interpolates images based on analyzing the local structure of the images. The original images are classified into two categories [14]: *homogenous areas* and *edge areas*. The interpolation of pixels in the different classified areas is accomplished by using individual interpolation algorithm, respectively. The conceptual procedure of our proposed algorithm is illustrated as Figure 2.

Determination of pixels to be either homogenous pixels or edge pixels is based on a preset threshold value.

First, the differences of pixels values along horizontal, vertical and diagonal directions are determined in a 3×3 window, respectively. Choice of the 3×3 window can significantly reduce computational loading and structural complexity of our proposed algorithm. We determine the differences of pixel values of the four directions one by one.

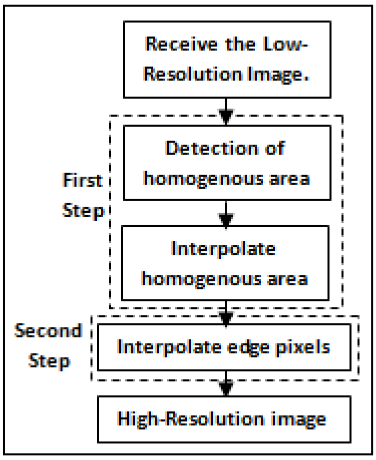
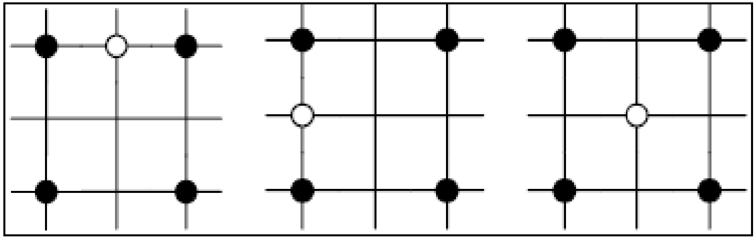

Receive the Low-Resolution Image.
First Step
Detection of homogenous area
Interpolate homogenous area
Second Step
Interpolate edge pixels
High-Resolution image

Figure 2: Proposed enhancement algorithm.

Figure (3) gives a schematic illustration to the four directions, one for horizontal, one for vertical and two for diagonal directions. If the difference is less than the preset threshold value, a non-interpolated pixel (white dot) will be classified as a homogenous pixel. Non-interpolated points in the homogenous areas are simply filled using bilinear interpolation [14].

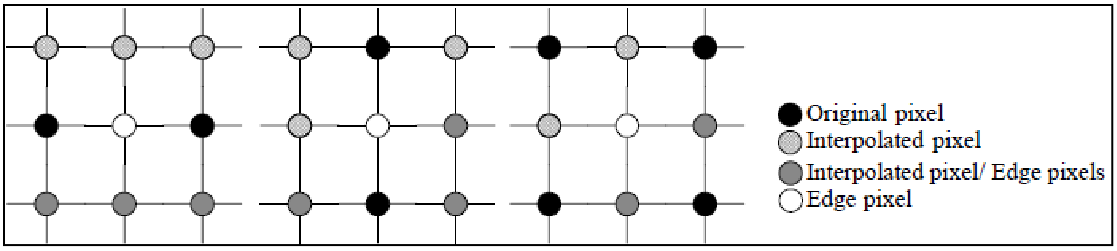
If the difference of pixel values is larger than the threshold value, the non-interpolated point is assigned as an edge pixel. After the first step, some non-interpolated pixels are remained as points in the edge areas. These pixels on edges will be left for further processing in the second step.



(a) Horizontal (b) Vertical (c) Diagonal

Figure 3: The differences of four directions (one for horizontal, one for vertical and two for diagonal).

The second step of the proposed algorithm is to interpolate the edge pixels using all neighboring pixels, which contain original pixels and interpolated pixels at the first step. Various types of neighboring pixels are shown in Figure (4), including black points, grey-points and spot points. For each pair of pixels, the smallest difference of pixel values implies the highest correlation between them. The edge pixel is interpolated along the direction of having minimum difference. If all of the spot points are interpolated pixels at the first step, the minimum difference is found on four directions across the white point (edge pixel). If a spot point is classified as an edge pixel, the minimum difference is obtained on the other directions excluding the edge pixel.



(a) Case 1 (b) Case 2 (c) Case 3

Figure 4: All neighboring pixels around a non-interpolated pixel.

A. Bilinear interpolation algorithm for homogenous pixels

We assume the X of low-resolution image with size H×W is enlarged to the Y of high-resolution image with size 2H×2W. The $Y_{2i,2j}$ is zoomed from $X_{i,j}$. We determine the homogenous pixels at $Y_{2i+1,2j}$, $Y_{2i,2j+1}$ and $Y_{2i+1,2j+1}$ using pixel difference criterion (Seen from Figure 5). When these pixels are homogenous, we interpolate these pixels using bilinear interpolation algorithm. The procedure for determining pixel difference is described as follows [14]:

```
$\Delta Y_1 = |Y_{2i,2j} - Y_{2i+2p,2j+2q}|$
$\Delta Y_2 = |Y_{2i+2,2j} - Y_{2i,2j+2}|$
$\Delta Y_3 = |Y_{2i,2j} - Y_{2i+2,2j+2}|$

if $\Delta Y_1 <$ threshold then
$Y_{2i+p,2j+q} = (Y_{2i,2j} + Y_{2i+2p,2j+2q})/2$
else
$Y_{2i+p,2j+q}$ is edge pixel
where $p, q \in \{(0,1),(1,0)\}$

if $\Delta Y_2 <$ threshold and $\Delta Y_3 <$threshold then
$\Delta Y_{min} = \min\{\Delta Y_2, \Delta Y_3\}$
if $\Delta Y_{min} = \Delta Y2$
$Y_{2i+1,2j+1} = (Y_{2i+2,2j} + Y_{2i,2j+2})/2$
else
$Y_{2i+1,2j+1} = (Y_{2i,2j} + Y_{2i+2,2j+2})/2$
else if $\Delta Y_2 <$ threshold then
$Y_{2i+1,2j+1} = (Y_{2i+2,2j} + Y_{2i,2j+2})/2$
else if $\Delta Y_3 <$ threshold then
$Y_{2i+1,2j+1} = (Y_{2i,2j} + Y_{2i+2,2j+2})/2$
else
$Y_{2i+1,2j+1}$ is edge pixel
```

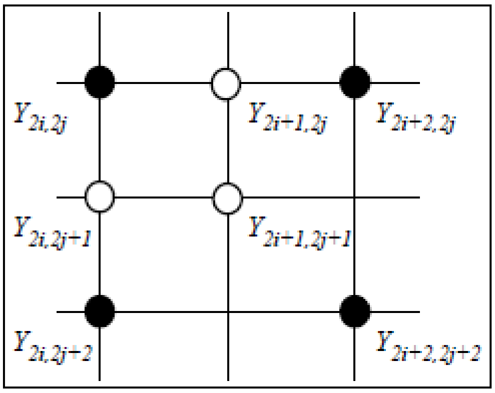


Figure (5): Interpolation of homogenous pixels.

After determining the pixel differences, the non-interpolated pixels are classified as the edge pixels and then processed by the following procedures.

B. Edge-oriented adaptive interpolation for edge pixels

Minimum difference of various directions can be found in either of the following cases. In case 1, all of the neighboring pixels are already interpolated at the first step.

The edge pixels are interpolated by the neighboring pixels of four directions. The minimum difference among these four directions is evaluated first. The direction of the minimum difference indicates that the edge pixel is oriented to this direction as shown in Figure (6). The procedure of the minimum difference algorithm is described as follows:

$diff_1 = |Y_{2i-1,2j} - Y_{2i+1,2j}|$

$diff_2 = |Y_{2i-1,2j-1} - Y_{2i+1,2j+1}|$
$diff_3 = |Y_{2i,2j-1} - Y_{2i,2j+1}|$
$diff_4 = |Y_{2i+1,2j-1} - Y_{2i-1,2j+1}|$
$diff_{min} = \min\{diff_k\}, for\ 1 \leq k \leq 4$

In case 2, some of the neighboring pixels which are not interpolated at the first step are classified as the edge pixels. The minimum difference is found by using the remaining neighboring pixels. In Figure (6), we assume that two pixels $Y_{2i-1,2j+1}$ and $Y_{2i,2j+1}$ are not interpolated at the first step, and the minimum difference is found by skipping the two directions along with $diff_3$ and $diff_4$. The procedure is expressed by the following equations.

$diff_1 = |Y_{2i-1,2j} - Y_{2i+1,2j}|$
$diff_2 = |Y_{2i-1,2j-1} - Y_{2i+1,2j+1}|$
$diff_{min} = \min\{diff_k\}$, for $1 \leq k \leq 2$

From the cases discussed above, the orientation of the $diff_{min}$ can be found. For example, if the $diff_{min}$ is found as $diff_1$, $Y_{2i-1,2j}$ correlating closely with $Y_{2i+1,2j}$,$Y_{2i,2j}$ is therefore interpolated by using $Y_{2i,2j}=(Y_{2i-1,2j}+Y_{2i+1;2j})/2$

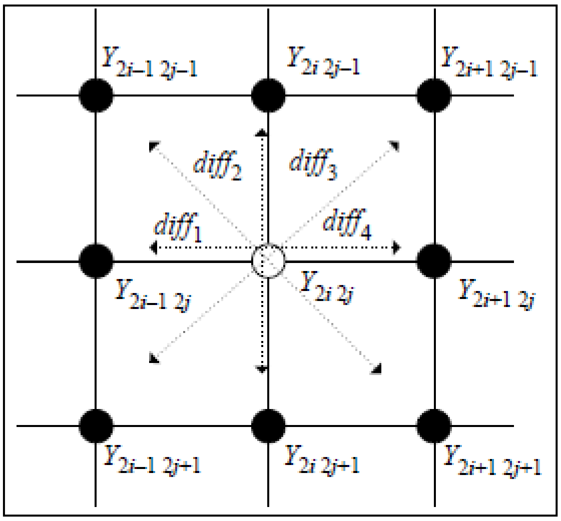


Figure (6): Interpolation of edge pixels.

5. EXPERIMENTAL RESULTS

In this section, we first compare subjective and objective qualities of Zero-order, Bilinear, Bicubic, NEDI and our proposed algorithms applied to interpolate still images. One grey-level image (Pepper) and one colour image (Pepper) are tested. Zero-order, bilinear and bicubic interpolation algorithms are well-known linear interpolation methods.

The performance of using bilinear interpolation is similar with that of using bicubic interpolation. Moreover, a comparison of the computational complexity for the five methods shown in table (1), which indicates that the bicubic method has much greater computational complexity than the bilinear method. Therefore, the bilinear interpolation algorithm is heuristically employed in the first step of our proposed algorithm in order to reduce computational loading.

Table 1:Comparison of computational complexity (Assume n pixels are interpolated)

Algorithm		Add	Sub.	Mul.	Div.	Shift	Inverse
Zero-order		--	--	--	--	--	--
Bilinear		3n	3n	6n	--	--	--
Bicubic		27n	45n	135n	9n	--	--
NEDI	Homogenous	n	--	--	--	n	--
	Edge	4n	--	1288n	--	--	4n
Proposed	Homogenous	n	--	--	--	n	--
	Edge	n	4n	--	--	n	--

Our proposed method has similar subjective quality compared to that of NEDI and performance superior to the rest of the three methods. Subjective qualities of colour images for these algorithms are demonstrated by portion of Pepper images in Figure (8). The interpolated image of our proposed algorithm has smoother edges than the others. Table (2) shows the objective quality of grey-level images up-sampling from 256×256 to 512×512 and from 128×128 to 512×512. Table (3) shows the objective quality of up-sampled colour images from 256×256 to 512×512 and 128×128 to 512×512. In general, our proposed method has better objective quality seconded to NEDI.

Table 2: PSNR (dB) comparison for grey level image

Algorithm	Pepper
256×256-512×512	
Zero-order	27.53
Bilinear	32.19
Bicubic	32.27
NEDI	33.54
Proposed	33.35
128×128-512×512	
Zero-order	24.39
Bilinear	27.52
Bicubic	27.50
NEDI	29.33
Proposed	28.96

Table 3: PSNR (dB) Comparison for colour image

Algorithm	Pepper
256×256-512×512	
Zero-order	26.14
Bilinear	30.01
Bicubic	29.95
NEDI	30.97
Proposed	30.84
128×128-512×512	
Zero-order	23.34
Bilinear	26.25
Bicubic	26.14
NEDI	27.66
Proposed	27.47

Figure (8): Subjective quality of a portion of Pepper image (Down-sampled from 512×512 to 128×128, then interpolated from 128×128 to 512×512).

Our proposed algorithm has relatively lower computational complexity and average CPU time, which is capable of being used for real-time interpolation of images.

Preset threshold value for pixels grouping is one of factors that may influence quality performance and processing speed. The total number of edge pixels in an image is increased when the threshold is set to a value closing to zero. In the circumstance, the PSNR of the interpolated image is accordingly increased, but computational complexity is much higher due to the increased number of edge pixels. Table (1) depicts the fact that the computation loading for interpolating an edge pixel is heavier than a homogenous pixel. The threshold value is set to 25 for our experiments with concerning quality performance and processing speed. The experimental results of our tests show that the amount of PSNR dropping is less than 0.2 dB compared to that threshold values are set to near zero. It is worthy to note that no decreasing of the subjective quality can be found in images interpolated by using threshold value set to 25.

The major goals of the proposed algorithm are to achieve real-time interpolation and to have better subjective quality.

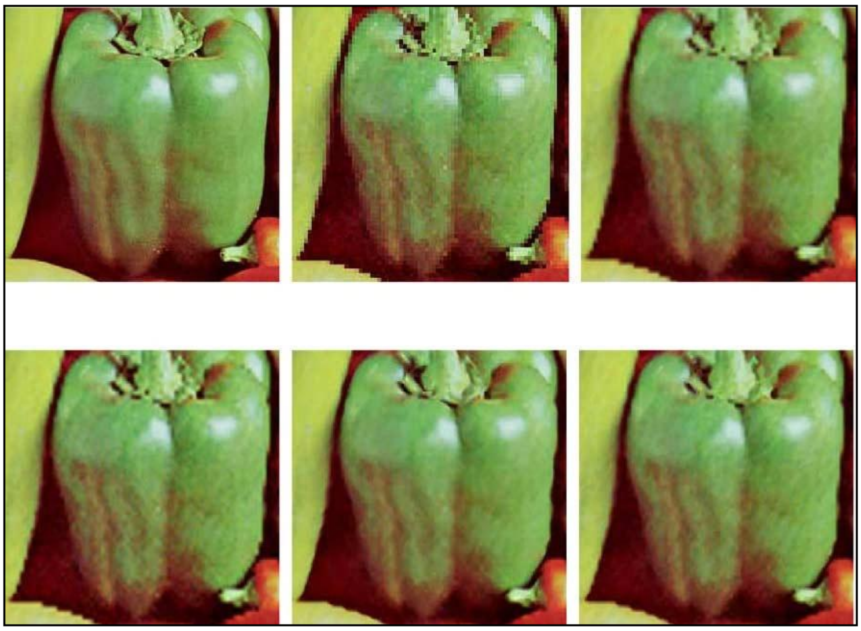


Figure (8): Subjective quality of a portion of Pepper image

6. CONCLUSIONS

The proposed web-based Architecture system is efficient to transport the image data from the server to the client by using the proposed algorithm that is allow as to achieve real-time interpolation and to have better subjective quality.

The proposed algorithm will improved the images quality after received in the client side, where the bandwidth for image transfer can be saved much more and image quality can be improved if the low-resolution video sequences is coded in encoder end and the low-resolution video sequences is afterward enlarged to high-resolution ones using the proposed algorithm.

We suggest that the algorithm can be applied to videoconference or video images based on the nature of low computational complexity and good image quality.

REFERENCES

[1] Ameer A. Mohammed Baqer, Suhas H. Patil, "Efficient Iris Biometrics Technique for Secure Distributed Systems", International Journal of Digital Society (IJDS), Infonomics Society, Volume 3, Issue 1, March 2012, p.p. 617 – 623.

[2] G. Couloris, J. Dollimore, T. Kinberg,2001, "Distributed Systems-Concepts and Design", 4th Edition, Addison-Wesley, Pearson Education, UK.

[3] A. Tanenbaum and M. Van Steen,2002, "Distributed Systems: Principles and Paradigms", Prentice Hall, Pearson Education, USA.

[4] W.K. Pratt, Digital Image Processing, Wiley, Toronto, Ont., Canada, 1978.

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

[6] S.D. Bayraker, R.M. Mersereau, A new method for directional image interpolation, IEEE International

Conference on Acoustics, Speech, and Signal Processing 4 (1995) 2383–2386.

[7] L. Khriji, M. Gabbouj, Directional-vector rational filters for color image interpolation Proceedings of the Tenth International Conference on Microelectronics (1998), pp. 236–240.

[8] C. Lee, B. Zeng, A novel interpolation scheme for rectangularly subsampled images, International Conference on Image Processing 3 (1999) 787–791.

[9] B. Zeng, M.S. Fu, C.C. Chuang, New interleaved hierarchical interpolation with median-based interpolators for progressive image transmission, Signal Processing 81 (2001) 431–438.

[10] H. Jiang, C. Moloney, A new direction adaptive scheme for image interpolation, International Conference on Image Processing 3 (2002) 369–372.

[11] H. Shi, R. Ward, Canny edge based image expansion, International Symposium on Circuits and Systems 1 (2002) 785–788.

[12] A.M. Darwish, M.S. Bedair, S.I. Shaheen, Adaptive resampling algorithm for image zooming, IEE Proceedings-Vision Image and Signal Processing 144 (4) (1997) 207–212.

[13] X. Li, M.T. Orchard, New edge-directed interpolation, IEEE Transactions on Image Processing 10 (10) (2001) 1521–1527.

[14] M. J. Chen, Chin-Hui Huang, Wen-Li Lee, A fast edge-oriented algorithm for image interpolation, Image and Vision Computing 23 (2005) 791–798.