Adaptive General Scale Interpolation Based on Similar Pixels Weighting

Mading Li, Jiaying Liu*, Jie Ren and Zongming Guo
Institute of Computer Science and Technology, Peking University, Beijing, P. R. China, 100871

Abstract— **In this paper, we propose an adaptive general scale interpolation algorithm considering the non-stationarity of natural images in local areas. In image $2\times$ enlargement, there are fixed relative positions between low-resolution (LR) pixels and high-resolution (HR) pixels. Unknown HR pixels can be estimated by their available LR neighbors. However, such relative positions are not fixed in the general-scale enlargement situations. The number and position of available LR pixels are indeterminate, therefore HR pixels can not be estimated by LR pixels. To make our method suitable for general scaling factors, we construct autoregressive (AR) models with pixels' neighbors instead of their available LR neighbors. Simultaneously, we introduce the similarity between pixels within a local window, which improves the method's performance by modeling the non-stationarity of image signals. Experimental results demonstrate the effectiveness of the proposed method on general scaling factors.**

I. Introduction

Image interpolation is a process that generates HR images utilizing the information in LR images. The key task of image interpolation is to estimate the pixels interpolated into the LR image. Conventional interpolation algorithms, such as Bilinear and Bicubic interpolations apply a convolution on every pixel of the HR image. Since these methods apply the same convolution on every pixel, they do not distinguish pixels in plain area and high frequency region. Although these methods have rather low complexity, they produce noticeable reconstruction artifacts near edges and blur the image to some extent.

With technological developments, the computational capabilities are increasing at full speed. Hence, lots of interpolation algorithms with high sophistication are proposed. Since the edge structure is one of the most salient features in natural images, many edge-guided interpolation algorithms are published. Li and Orchard [1] proposed a new edge-directed interpolation (NEDI). They computed the parameters of the AR model in the LR image by a least square problem and estimated HR pixels by their neighbor LR pixels using corresponding parameters. Zhang and Wu [2] further proposed a soft-decision adaptive interpolation (SAI) based on NEDI. They added a cross-direction AR model and more correlations between LR pixels and HR pixels. Thus, SAI gained a better performance upon NEDI. However, these algorithms are based on the assumption that the image is piecewise stationary. To account for the fact that natural images are not always stabilized in local windows, our previous work [3] proposed an implicit piecewise autoregressive model-based image interpolation algorithm (IPAR) based on similarity modulated block estimation. In IPAR [3], a similarity probability model is proposed to model the non-stationarity of image signals.

The adaptive algorithms mentioned above have limitation that they can only deal with enlargement whose magnification is two or a power of two. As the popularity of video devices rises, resolutions of video sequences and images differ greatly among devices. Thus, general-scale enlargement is required. Wu *et al.* [4] proposed an adaptive resolution up-conversion method implemented in H.264/SVC, providing support to arbitrary scaling factors between spatial resolution of the base and refinement layers. This method used two directional AR models that are constructed of pixels' neighbors. However, the method did not consider the instability of natural images in local areas either.

In this paper, we propose a novel image interpolation algorithm that can manage arbitrary general scaling factor while considering the instability of natural images in local areas. At first, we generalize the interpolation by adjusting the composition of AR models and making a constraint by comparing the LR pixels and the down-sampling results of the corresponding region in the HR image. The new AR models are constructed by the pixel's neighbors instead of its neighbor LR pixels. Then we propose a weight calculation method to determine the similarity between pixel pairs in a local window. Based on this method, we get a weight distribution matrix of the whole local window, which can be used to adjust the formulation of the AR models. Finally, since the HR pixels and parameters of two AR models are both unknown, we use structured total least-squares solution (STLS) to linearize the objective function and solve the problem by an iterative process. Experimental results show that our method preserves better details, especially details around edge-structures, among Wu's work [4] and the method implemented in JSVM [5] when the factor is a general rational number. When it comes to $2\times$ enlargement, our method is competitive with IPAR.

The rest of the paper is organized as follows: Section II introduces the two AR models briefly. Section III generalizes the interpolation algorithm to general scaling factor, then describes the new proposed image interpolation algorithm based on weight distribution in the local window. Experimental results and analysis of the proposed method are presented in Section IV. Finally, Section V concludes this paper.

*Corresponding author

This work was supported by National Natural Science Foundation of China under contract No. 61101078, National Basic Research Program(973 Program) of China under contract No.2009CB320907 and Beijing Nova program under contract No. 2010B001.

978-1-4673-5762-3/13/$31.00 ©2013 IEEE

II. Autoregressive model

An autoregressive (AR) model is a type of random process that is often utilized to model and predict various types of natural signals. It is a set of linear formulas that attempts to obtain an estimation of a system based on the given information. In image processing, every pixel in an image can be estimated by its adjacent neighbors with certain weights. The AR model is defined as

$$X(m,n) = \sum_{(i,j)\in\Omega} \varphi(i,j)X(m+i,n+j) + \sigma, \tag{1}$$

where Ω and φ are the adjacent neighbors and their weights to pixel $X(m,n)$, respectively. σ is the estimating error. Based on the assumption that images maintain stability in a local window W, we compute the weights by solving the linear least squares problem showed below,

$$\min_{\varphi} \|\mathbf{X} - \mathbf{X}_n\varphi\|, \tag{2}$$

where X is an $M \times 1$ vector consisting of pixels in W. The i^{th} row of X_n consists of adjacent neighbors of the i^{th} pixel in X. Then unknown pixels in W can be estimated by LR pixels with the same model parameters φ.

For better estimation, two kinds of AR models in different directions are applied. One of them uses a pixel's cross-direction adjacent neighbors to estimate it, the other uses its diagonal direction adjacent neighbors. Two sets of weights can be calculated by performing these two AR models in a local window. Therefore, constraints on pixels are stronger, and the unknown pixels can be estimated more precisely.

These types of AR model are used in [1], [2]. However, there are two drawbacks. First, in general-scale cases there may be insufficient LR pixels in W to estimate unknown pixels. Second, the stationary assumption aforementioned does not always hold in most natural images. The solutions of these two problems will be given in Section III-A and Section III-B.

III. Generalized Image Interpolation Algorithm

In this section, we present a generalized image interpolation. The difference between 2× enlargement methods mentioned before and generalized methods is elaborated at first. Then we introduce a method to define the similarity of two pixels. At last, our generalized image interpolation algorithm is proposed.

A. Generalization of Interpolation at Arbitrary Scaling Factors

When we use interpolation algorithms to obtain an HR image, there are always plenty of pixels that are extracted from the LR image directly. In other words, these pixels are exactly the same as the corresponding pixels in the LR image. We name these pixels *fixed*-pixels and *inter*-pixels for other pixels. In a local region in HR image, the more the *fixed*-pixels there are, the more information we can get to interpolate *inter*-pixels. Fig.1 shows that the relative location between LR pixels and HR pixels varies when the scaling factor changes. When the scaling factor is 2, there is only one pixel between adjacent *fixed*-pixels (Fig.1(a)). Hence, it is a good way to estimate *inter*-pixels from their neighboring *fixed*-pixels. However, 2× enlargement is just one of the special circumstances of image enlargement at general scaling factors. In Fig.1(c), N is the minimal integral multiples of the scaling factor. As N increases, there are less *fixed*-pixels around *inter*-pixels. It should be pointed out that the locality stationary still remains in arbitrary scaling factors. Thus, we apply two AR models on pixels. Unlike methods for 2× enlargement, *inter*-pixels are estimated by its neighbor pixels no matter what types of pixels they are. The formation is modeled as

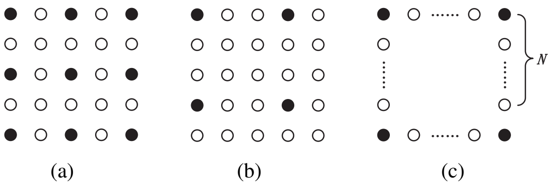

Fig. 1. HR pixel and LR pixel in local region at different scaling factors. (a) Scaling factor = 2.0. (b) Scaling factor = 1.5. (c) More general situation. An general scaling factor, such as 1.57, is likely to produce a really large N.

$$\min_{y,a,b}\{\alpha\|y_c - \mathbf{A}y\|^2 + \beta\|y_c - \mathbf{B}y\|^2\}, \tag{3}$$

where vector y consists of pixels in a local window of HR image. Vector y_c consists of pixels in the local window excluding the pixels on boundaries of the window. α and β are the coefficients that control the weight of two AR models. **A** and **B** are defined as follow,

$$\begin{aligned} \mathbf{A}(i,j) &= \begin{cases} a_k, & \text{if } y_j \text{ is the } k^{th} \text{ pixel of } y_i\text{'s} \\ & \text{diagonal neighbors,} \\ 0, & \text{otherwise.} \end{cases} \\ \mathbf{B}(i,j) &= \begin{cases} b_k, & \text{if } y_j \text{ is the } k^{th} \text{ pixel of } y_i\text{'s} \\ & \text{cross-direction neighbors,} \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{4}$$

where $a = (a_1, a_2, a_3, a_4)$ and $b = (b_1, b_2, b_3, b_4)$ are the parameters of two AR models. The question is, both of the *inter*-pixels (y) and the parameters of AR models (a and b) are unknown. We use the iterative procedure mentioned in Section III-C to solve this problem.

Since *fixed*-pixels are not utilized in AR models, we use these valuable pixels to improve our result in a different way. As we mentioned before, there is a corresponding relation between the pixels in LR image and *fixed*-pixels in HR image. Another way to utilize the *fixed*-pixels is to use them from the LR image. For a local window in HR image, we down-sample it by Bicubic method and compare it to the corresponding window in LR image. The constrain is modeled as

$$\|x - \mathbf{D}y_c\|^2, \tag{5}$$

where vector x consists of pixels in a local window of LR image and matrix **D** represents the Bicubic down-sampling process. Obviously, the smaller this constrain's value is, the better estimation we have. Such constrain is useless in 2× enlargement because the pixels that is down-sampled are exactly the same with the corresponding pixels in LR image.

B. Weight Distribution in a Local Window

AR model works well when the statistics in the local window are stationary because all AR models of a same type share a same weights in the whole window. However, natural

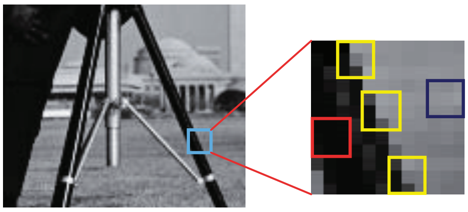

Fig. 2. Similar patches in local window. Patches in yellow frames are similar to each other but different from those in red or navy blue.

pictures do not maintain stability in most local windows. For example, as illustrated in Fig.2, there are significant differences between an edge-crossing area and a plain area in a local window. Thus, estimations of AR models in this window are not robust. In order to solve this problem, we introduce a method to judge the similarity between the pixel to-be-output (usually the center pixel) and other pixels in the local window.

Naturally we prefer to give more weight on the pixel (in other words, its corresponding AR model) that are more similar to the center pixel, and vice versa. It is commonly agreed that pixels are similar if there is a small difference between their local structures. Moreover, two pixels are likely to be similar if they are close to each other. Thus, the weight is a composite of two parts. One of them is the similarity of two center pixels' local structures. The other is the distance between them. The weight $w(m,n)$ between two pixels m and n is defined as

$$w(m,n) = w_{ls}(m,n)w_d(m,n), \tag{6}$$

where $w_{ls}(m,n)$ represents the similarity of two center pixels' local structures. $w_d(m,n)$ represents the degree of two pixels' distance. They are described as

$$w_{ls}(m,n) = e^{-\|L_m - L_n\|^2/\varepsilon_1}, \tag{7}$$

$$w_d(m,n) = e^{-\|P_m - P_n\|^2/\varepsilon_2}, \tag{8}$$

where L_m and L_n represent vectors consisting of the 8-connected neighborhood of m and n, respectively; P_m and P_n are the spatial coordinates of m and n, respectively. ε_1 and ε_2 control the shape of the exponential function.

Our previous work [3] proposed a similarity modulated block estimation, in which pixel's local structure consists of its four nearest diagonal LR pixel neighbors, since they are accurate and reliable to form a local structure. Such characteristic does not exist in general situations. So we choose the 8-connected neighborhood to compose the local structure.

After obtaining all pixels' weight to the center pixel, we can form a diagonal weight matrix $\mathbf{W}$ which represents the weight distribution in current local window.

C. The Generalized Interpolation Algorithm

By adding $\mathbf{W}$ to (3), combining with (5) we can get the objective function described below,

$$\min_{y,a,b}\{\alpha\|\mathbf{W}(y_c - \mathbf{A}y)\|^2 + \beta\|\mathbf{W}(y_c - \mathbf{B}y)\|^2 + \lambda\|x - \mathbf{D}y\|^2\}. \tag{9}$$

The objective function (9) can be represented by a least square problem as

$$\min_{y,a,b}\|R(y,a,b)\|^2. \tag{10}$$

where $R(y,a,b)$ is the residue vector, representing the estimating residue. It is described as

$$R(y,a,b) = \begin{bmatrix} \sqrt{\alpha}\mathbf{W}(\mathbf{I}-\mathbf{A})y \\ \sqrt{\beta}\mathbf{W}(\mathbf{I}-\mathbf{B})y \\ \sqrt{\lambda}(x-\mathbf{D}y_c) \end{bmatrix}. \tag{11}$$

The least-squares problem in (10) is nonlinear. In order to make it easier to be solved, we use the structured total least-squares solution to linearize the problem. Let Δy, Δa and Δb be the small changes in y, a and b respectively. To better constrain the pixels inside the window, we keep pixels on the boundaries of the window unchanged. Thus, $\mathbf{A}$ and $\mathbf{B}$ can be decomposed to $[\mathbf{A}_c, \mathbf{A}_b]$ and $[\mathbf{B}_c, \mathbf{B}_b]$ respectively. Let T be the length of the square window, then $\mathbf{A}_c$ and $\mathbf{B}_c$ are composed by the first $(T-2)^2$ columns of $\mathbf{A}$ and $\mathbf{B}$ respectively, $\mathbf{A}_b$ and $\mathbf{B}_b$ are composed by the remaining columns.

The residue vector $R(y,a,b)$ can be linearized as

$$R(y+\Delta y, a+\Delta a, b+\Delta b) =$$
$$R(y,a,b) - \begin{bmatrix} \sqrt{\alpha}\mathbf{W}(-\mathbf{I}+\mathbf{A}_c)\Delta y_c + \mathbf{E}_1\Delta a \\ \sqrt{\beta}\mathbf{W}(-\mathbf{I}+\mathbf{B}_c)\Delta y_c + \mathbf{E}_2\Delta b \\ \sqrt{\lambda}(x - \mathbf{D}\Delta y_c) \end{bmatrix}. \tag{12}$$

where $\mathbf{E}_1$ and $\mathbf{E}_2$ are constructed as follows: the k^{th} row of $\mathbf{E}_1$ is a vector constructed by four diagonal neighbors of pixel y_k. The k^{th} row of $\mathbf{E}_2$ is a vector constructed by four cross-direction neighbors of pixel y_k.

Let

$$\mathbf{C} = \begin{bmatrix} \sqrt{\alpha}\mathbf{W}(-\mathbf{I}+\mathbf{A}_c) & \mathbf{E}_1 & \mathbf{0} \\ \sqrt{\beta}\mathbf{W}(-\mathbf{I}+\mathbf{B}_c) & \mathbf{0} & \mathbf{E}_2 \\ \sqrt{\lambda}\mathbf{D} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

$$\Delta R = \begin{bmatrix} \Delta y & \Delta a & \Delta b \end{bmatrix}^T.$$

For convenient representation, we rewrite (12) as

$$\min_{\Delta y, \Delta a, \Delta b}\|R(y,a,b) - \mathbf{C}\cdot\Delta R\|^2. \tag{13}$$

Therefore, given the initial values of y, a and b, we can obtain ΔR and use it to update y, a and b for the next iteration. In our implementation, we use Bicubic's result as the initial value of y. a and b are initialized as $(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$.

The iterative process is computationally expensive. In order to alleviate the complexity of the proposed method, we only apply the proposed algorithm on high-frequency areas. Furthermore, we output the center 3×3 pixels at once. It may

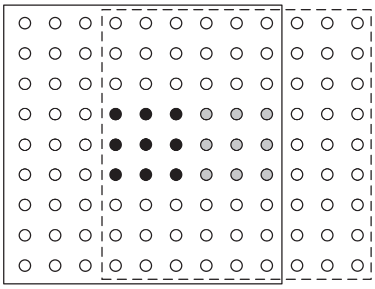

Fig. 3. Possible configuration used in proposed algorithm. The size of windows is set to be 9×9. The square in full line represents the current window while the square in dotted line represents its next window. Black pixels and gray pixels are the output of the current window and the next window, respectively.

slightly reduce the performance, but can lead to a 9 times speed-up. Meanwhile, in order to avoid blocking artifacts, we produce an overlapping region between adjacent windows and the offset is set to be 3 pixels. In Fig.3, the output pixels of two adjacent windows are shown. It can be seen that every pixel except for the pixels in boundary areas of the whole image can be processed.

IV. Experimental Results

The proposed interpolation is implemented on MATLAB 7.10 platform. We implemented Wu's work [4] by ourselves, using our data fidelity constraint and an 11×11 window. The proposed algorithm is compared with Bicubic and the method implemented in JSVM [5] on the factor of 1.5. It is also compared with Bicubic, SAI [2], IPAR [3] and Wu's work [4] on the factors of 1.7 and 2. We tested our interpolation algorithm on a large number of images. The testing images are selected from the Kodak database and other standard testing images. More results are presented in our web site [6].

For a scaling factor s, we firstly generate an LR image by down-sampling an original HR image by a factor of $1/s$. Then we use different methods to obtain the HR images from the LR image and compare them with the original HR image. Peak Signal-to-Noise Ratio (PSNR) is selected as the objective evaluation criterion.

TABLE I
Average PSNR of results in different methods, scaling factor $s = 1.5$

Sequences	JSVM	Bicubic	Proposed
Akiyo	39.50	41.79	**43.81**
Foreman	39.60	42.17	**43.94**
Highway	40.85	43.07	**45.70**
Average	39.99	42.34	**44.48**

TABLE II
Average PSNR of results in different methods, scaling factors $s = 1.7$ & 2

s	Images	Bicubic	SAI	IPAR	Wu's	Proposed
1.7	Monarch	34.27	-	-	31.72	**35.20**
	Lena	36.26	-	-	34.64	**36.78**
	Bike	28.33	-	-	26.72	**29.45**
2	Pepper	32.06	31.84	**32.69**	32.43	32.56
	Airplane	29.40	29.62	**30.05**	29.73	29.87
	Lighthouse	26.97	26.70	26.76	26.91	**27.13**

Weight coefficients α, β and λ are set by 0.2, 0.3 and 0.5. ε_1 and ε_2 are set to be 17 and 33, respectively. The window size is particularly important to our algorithm (a large size window may impact the locality stationary, and a small one cannot provide enough structural information), we set it to be 11×11. Results of $1.5\times$, $1.7\times$ and $2\times$ magnification are shown in Table I and Table II.

It is interesting to notice that, in Table II, our method's performance is better than SAI [2] and competitive with IPAR [3] in $2\times$ enlargement. It should be pointed out again that, NEDI [1], SAI [2] and IPAR [3] are all designed only for $2\times$ enlargement. The proposed method is suitable for general scaling. The gaps between the performances of Wu's work [4] and the proposed work in $2\times$ enlargement are quite small, but the gaps in $1.7\times$ and $1.5\times$ magnification are rather big.

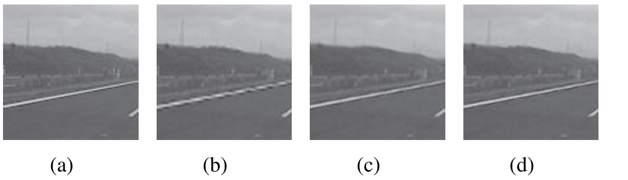

Fig. 4. Subjective image quality comparison of $1.5\times$ enlargement. (a) a part of first frame of original sequence *Highway*. (b) result of JSVM. (c) result of Wu's work. (d) result of proposed work.

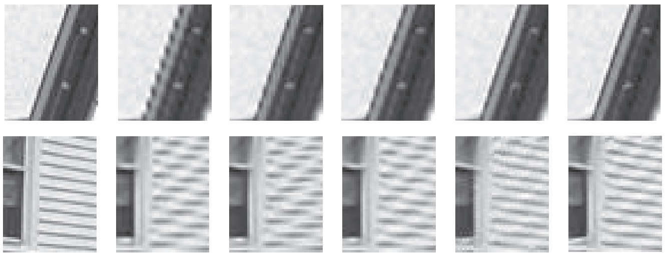

Fig. 5. Subjective image quality comparison of $2\times$ enlargement. The upper image is a part of *House* and the nether image is a part of *Lighthouse*. From left to right: original image, Bicubic, SAI, IPAR, Wu's work, proposed method.

Subjective image quality is also demonstrated in Fig.4 and Fig.5. In general-scale cases, the proposed method presents more clean edges than JSVM [5]. As shown in Fig.4, our method presents better edges. In $2\times$ enlargement situation (Fig.5), Bicubic interpolation presents fuzzy areas around edges. SAI [2] and IPAR [3] present rather clean edges, but there are some jags on the sharp edges and affect the subjective image quality. The proposed method presents clean sharp edges and does not produce aliasing effects.

V. Conclusions

In this paper, we present an interpolation algorithm suitable for general scaling factors considering the non-stationarity of natural images in local areas. The AR models are constructed by the pixel's neighbors instead of its available LR neighbors in order to solve the general-scale interpolation problem. Similarity of pixels in the local window is utilized to alleviate the inaccurate estimation caused by the local instability of natural images. As the experimental results show, our method performs the best in general-scale cases and it preserves better details around edges among other competitive interpolation algorithms.

References

[1] Xin Li, Michael Orchard, "New edge-directed interpolation", *IEEE Transactions on Image Processing*, vol.10, no.10, pp.1521-1527, 2001

[2] Xiangjun Zhang, Xiaolin Wu, "Image Interpolation by Adaptive 2-D Autoregressive Modeling and Soft-Decision Estimation", *IEEE Transactions on Image Processing*, vol.17, no.6, pp.887-896, 2008

[3] Jie Ren, Jiaying Liu, Wei Bai, Zongming Guo, "Similarity modulated block estimation for image interpolation", *IEEE International Conference on Image Processing (ICIP)*, 2011

[4] Xiaolin Wu, Mingkai Shao, Xiangjun Zhang, "Improvement of H.264 SVC by model-based adaptive resolution upconversion", *IEEE International Conference on Image Processing (ICIP)*, 2010

[5] Andrew Segall and Gary Sullivan, "Spatial Scalability Within the H.264/AVC Scalable Video Coding Extension", *IEEE Transactions on Circuits and Systems for Video Technology*, vol.17, no.9, pp.1121-1135, 2007

[6] http://www.icst.pku.edu.cn/course/icb/GenInterp.html