ARBITRARY FACTOR IMAGE INTERPOLATION BY CONVOLUTION KERNEL CONSTRAINED 2-D AUTOREGRESSIVE MODELING

Ketan Tang†, *Oscar C. Au*†, *Yuanfang Guo*†, *Jiahao Pang*†, *Jiali Li*†, *Lu Fang*‡

†The Hong Kong University of Science and Technology
{tkt, eeau, eeandyguo, jiali, jpang}@ust.hk
‡University of Science and Technology of China, fanglu@ustc.edu.cn

ABSTRACT

Among existing interpolation methods, convolution-based methods are able to perform arbitrary factor interpolation but the results are usually blurry or jaggy, adaptive interpolation methods usually can reduce the blurry and jaggy artifacts but cannot handle arbitrary factor interpolation. In this paper we propose an arbitrary factor adaptive interpolation algorithm by combining 2-D piecewise autoregressive (PAR) modeling and convolution kernel constraint. PAR model ensures local geometries are well preserved thus the resultant image is not blurry or jaggy. Convolution kernel constraint ensures the recovered high resolution image consistent with the low resolution image, and also provides the flexibility to handle arbitrary interpolation factor. Experiment results show that our algorithm achieves state-of-the-art performance for any interpolation factor.

Index Terms**—** arbitrary factor, interpolation, autoregressive model

1. INTRODUCTION

As the display resolution continues increasing, it is demanded to display Low Resolution (LR) content on High Resolution (HR) displays with high quality. Image interpolation is therefore used to get a HR image/video from a LR image/video. Depending on whether the interpolation is adaptive at different locations, there are usually two categories of interpolation methods. One is convolution based interpolation (CI), in which a fixed convolution kernel is used for all pixels. The other is adaptive interpolation (AI) which changes the interpolation parameters at different local regions. Commonly used convolution kernels include bilinear, bicubic [1], lanczos[2], etc. In [3] Huang et. al. apply consistent resampling theory (CRT) to image resizing and rotation and achieves better results than B-spline interpolation. Essentially their method applies an enhancement filter to the resized image, which means it still belongs to CI. The strength of CI is that their complexities are relatively low, and they can be used for Arbitrary Factor Interpolation (AFI), meaning that the result HR image can be of arbitrary larger size (even non-integer) than the input LR image. However they have common drawback that they cannot adapt to varying pixel structures in an image. As a result, they suffer from some inherent defects such as staircase effect, blurred details, and ringing artifacts.

Adaptive interpolation, on the other hand, can better interpolate images based on local pixel structures. Existing methods include [4, 5, 6, 7] etc. The work that is most related to this paper is SAI [4], in which Zhang and Wu model an image using a 2-D PAR model, and recover the HR image block by block in a soft-decision way. Despite the strength of preserving edge structures, these adaptive methods have a common drawback that they can only handle interpolation with a fixed integer factor. For a different zooming (probably non-integer) factor, these methods have to do multiple times of adaptive interpolation, and then some convolution based interpolation (e.g. bicubic) which may blur the image eventually, and also the artifacts may be accumulated and amplified in the multiple steps. Therefore it is meaningful to design an arbitrary factor adaptive interpolation algorithm. Some works have been done in the literature. In [8] Muresan and Parks cast the interpolation problem into the framework of adaptive optimal recovery and achieve arbitrary factor interpolation by designing the representors according to zooming factor. However the results of their algorithm are still blurry and the PSNR gain compared to cubic is not quite significant.

In this paper we combine convolution based interpolation and adaptive interpolation, so that we can perform arbitrary factor interpolation adaptively at different locations. The idea is to separate the optimization problem into two parts, an adaptive pixel correlation model serving as the cost function and a convolution-based image degradation model serving as the constraint. The proposed algorithm is named Arbitrary Factor Autoregressive Interpolation (AFAI), for that it achieves arbitrary factor interpolation with autoregressive model.

The rest of the paper is organized as follows. In section 2 we review PAR model and formulate our problem into a constrained least square problem. We solve this problem in section 3. The experiment results are given in 5. And we

978-1-4799-2341-0/13/$31.00 ©2013 IEEE

conclude our work in section 6.

2. IMAGE INTERPOLATION AS A CONSTRAINED LEAST SQUARE PROBLEM

Piecewise autoregressive (PAR) model is used for the adaptive pixel correlation modeling, for that with PAR model the local texture of images can be estimated very well, and the interpolation can be easily formulated as a least square problem which has a closed form solution. Similar as in [4], we model an image as

$$X(i,j) = \sum_{(m,n)\in W} \alpha(m,n)X(i+m,j+n) + v_{i,j} \tag{1}$$

where W is a local window, $v_{i,j}$ is a random perturbation independent of spatial location and the image signal. $\alpha(m,n)$ is the autoregressive coefficient for the (m,n)-th neighbor of pixel $X(i,j)$, which is assumed constant in local window W but different between different windows.

The image degradation process is modeled as convolution-based image downsampling which is flexible enough to adapt to any zooming factor and any blurring kernel. Let I_h be the HR image to be estimated, I_l be the observed LR image which is a downsampled version of the HR image by an arbitrary scaling factor r, $x_i \in I_l$ and $y_i \in I_h$ be the pixels of images I_l and I_h, $y_{i\diamond t}$ $(t = 1, 2, \ldots, 8)$ be the neighbors of pixel location i in the HR image, and $y^d_{i\diamond t}$ be the diagonal neighbors, $y^{hv}_{i\diamond t}$ be the horizontal and vertical neighbors (HV neighbors for short). The HR image can be estimated block by block with the following contained least square problem

$$\begin{aligned} \min_{\{y,a,b\}} \quad F(y,a,b) \quad & = \textstyle\sum_{i\in W}\left(\|y_i - \sum_t a_t y^d_{i\diamond t}\|^2 + \lambda\|y_i - \sum_t b_t y^{hv}_{i\diamond t}\|^2\right) \\ & \triangleq \|C^d\tilde{y}\|^2 + \lambda\|C^{hv}\tilde{y}\|^2 \\ \text{s.t.} \quad & S\tilde{y} = x \end{aligned} \tag{2}$$

where λ controls the importance of the HV correlation over diagonal correlation, x is the vector of LR pixels that are inside W, and sampling matrix $S = DH$ computes x, where H is a blurring matrix and D is a downsampling matrix. The matrix C^d and C^{hv} are

$$C^d(i,j) = \begin{cases} 1, & \text{if } \tilde{y}_j = y_i, \\ -a_t, & \text{if } \tilde{y}_j = y^d_{i\diamond t}, \\ 0, & \text{otherwise,} \end{cases}$$

C^{hv} is similar except that $C^{hv}(i,j) = -b_t$ if $\tilde{y}_j = y^{hv}_{i\diamond t}$. $\tilde{y}$ is the vector of all HR pixels related to x, including y and pixels outside W. Let ϵ be the vector of pixels outside W but related to x, then we have $\tilde{y} = \begin{bmatrix} y^T & \epsilon^T \end{bmatrix}^T$. The equality constraint can be rewritten accordingly:

$$\begin{aligned} & \begin{bmatrix} S_y & S_\epsilon \end{bmatrix} \begin{bmatrix} y \\ \epsilon \end{bmatrix} = x \\ \Longleftrightarrow \quad & S_y y = x - S_\epsilon \epsilon \end{aligned} \tag{3}$$

where S_y consists of the columns corresponds to y and S_ϵ consists of the rest columns which correspond to ϵ.

In reality, usually we do not perform blurring and subsampling separately, because this scheme cannot handle non-integer factor subsampling. Instead, we directly apply sampling matrix S to HR images. The pixel value of LR image at position (u,v) has the following form

$$I_l(u,v) = \sum_{m,n} I_h(u_m,v_n)\phi(u_m - u, v_n - v). \tag{4}$$

where $\phi(\cdot)$ is a convolution kernel. Take bilinear sampling for example, we have

$$\phi(u,v) = \begin{cases} (1-\frac{|u|}{h_u})(1-\frac{|v|}{h_v}), & |u| \le h_u, |v| \le h_v, \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where h_u, h_v are the step size of horizontal and vertical direction, which are proportional to the horizontal and vertical scaling factor. Suppose the horizontal and vertical scaling factor are the same, $h_u = h_v = r$, we have

$$I_l(u,v) = \sum_{m,n} I_h(u_m,v_n)(1-\frac{|u-u_m|}{r})(1-\frac{|v-v_n|}{r}). \tag{6}$$

We can rewrite (6) into matrix form by stacking all LR pixels inside window W into a vector $x = \{I_l(u,v)|(u,v) \in W\}$, and all HR pixels related to x into a vector $\tilde{y} = \{I_h(u_m,u_n)|(u,v) \in W,\ |u-u_m| \le r,\ |v-v_n| \le r\}$. The elements of matrix $S(i,j)$ are the corresponding interpolation coefficients of $x(i)$ and $\tilde{y}(j)$, i.e.

$$S(i,j) = (1-\frac{|u(i)-u(j)|}{r})(1-\frac{|v(i)-v(j)|}{r}) \tag{7}$$

where $(u(i),v(i))$ is the 2D location in HR image space of the i-th pixel.

3. JOINT OPTIMIZATION USING GAUSS-SEIDEL METHOD

As shown in [9] that the constrained least square problem 2 can be effectively solved using Gauss-Seidel method. Gauss-Seidel method is to alternatively fix one set of variables (e.g. $\{a,b\}$) and optimize on the other set (e.g. y). Initial value of y can be obtained by bicubic interpolation.

Since a and b are naturally decoupled and do not depend on the constraint, we have two unconstrained sub-problems:

$$\begin{aligned} a^{(n+1)} &= \arg\min_a \sum_{i\in W} \|y_i - \sum_t a_t y^d_{i\diamond t}\|^2 \\ &= (Y_a^T Y_a)^{-1} Y_a^T y^{(n)} \end{aligned} \tag{8}$$

$$\begin{aligned} b^{(n+1)} &= \arg\min_a \sum_{i\in W} \|y_i - \sum_t b_t y^{hv}_{i\diamond t}\|^2 \\ &= (Y_b^T Y_b)^{-1} Y_b^T y^{(n)} \end{aligned} \tag{9}$$

where $Y_a(i,\cdot)=[y^d_{i\diamond1},y^d_{i\diamond2},y^d_{i\diamond3},y^d_{i\diamond4}]$, $Y_b(i,\cdot)=[y^{hv}_{i\diamond1},y^{hv}_{i\diamond2},y^{hv}_{i\diamond3},y^{hv}_{i\diamond4}]$. Note that all elements of y are inside W, however elements of Y_a and Y_b may be outside W. For the ease of representation we denote $a^{(n+1)}$ and $b^{(n+1)}$ by $\hat{a}$ and $\hat{b}$ in the following.

For solving y, by defining $C = \left[\begin{array}{cc} (C^d)^T & \lambda(C^{hv})^T \end{array} \right]^T = \left[\begin{array}{cc} C_y & C_\eta \end{array} \right]$ where C_y consists of the columns of C corresponding to y, C_η consists of the remaining columns of C corresponding to the outermost pixels η which are outside W, we have

$$y^{(n+1)} = \arg\min_y \quad \|C^d\tilde{y}\|^2 + \lambda\|C^{hv}\tilde{y}\|^2 = \|C_y y + C_\eta \eta\|^2$$

$$\text{s.t.} \quad S_y y = x - S_\epsilon \epsilon \qquad (10)$$

This equality constrained quadratic optimization has a closed form solution (§10.1.1 of [10]):

$$\left[\begin{array}{c} y \\ \mu \end{array} \right] = \left[\begin{array}{cc} C_y^T C_y & S_y^T \\ S_y & 0 \end{array} \right]^{-1} \left[\begin{array}{c} -C_y^T C_\eta \eta \\ x - S_\epsilon \epsilon \end{array} \right] \qquad (11)$$

where μ is Lagrange multiplier which can be ignored. The λ value in Eq. (10) is estimated as proportion of the fitting error of a and b so that $\|C^d\tilde{y}\|^2 \approx \lambda\|C^{hv}\tilde{y}\|^2$, i.e.

$$\lambda = \|y - Y_a a\|^2 / \|y - Y_b b\|^2 \qquad (12)$$

4. BLURRING KERNEL ESTIMATION

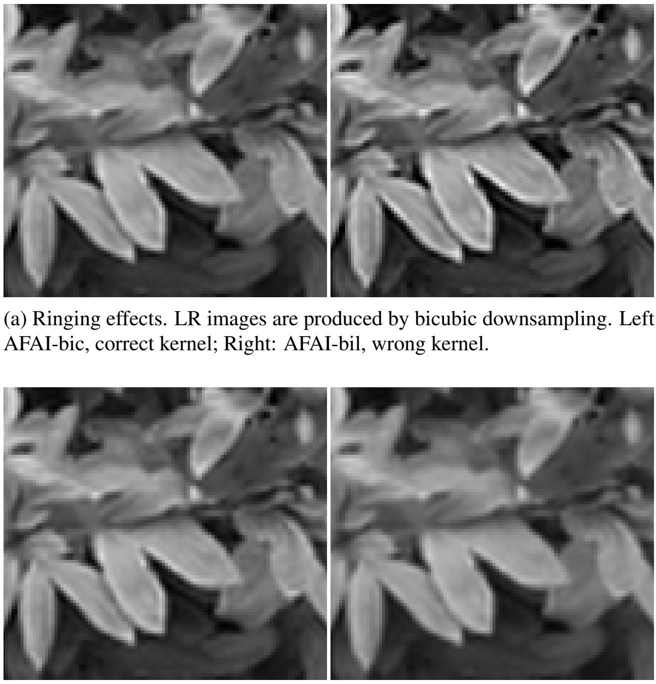
An important procedure in super resolution is to estimate the downsampling kernel of image degradation. Some researchers assume Gaussian kernel [11], while some assume average kernel [8], however it may not match the real degradation process. Experiments have shown that if the downsampling kernel is chosen incorrectly, severe artifacts may occur. Take bilinear and bicubic kernels for example. If the LR images are generated by bicubic downsampling and we assume bilinear kernel in our algorithm, the results will have ringing artifacts (Gibbs phenomenon) at strong edge regions, see Fig. 1a . On the other hand, if the LR images are generated by bilinear downsampling and we assume bicubic kernel in our interpolation algorithm, the result HR images will be very blur, see Fig. 1b. Based on the characteristics of these artifacts, the real downsampling kernel can be estimated. Basically we run trial interpolation process with both bilinear and bicubic kernel (or other possible kernels), then choose the one with fewest artifacts. The blurring kernel estimation algorithm is as follows.

1. Normalize the dynamic range of LR images to $[T_l, T_h]$. Suppose the linear transform function is $I_l' = kI_l + b$. Run trial interpolation using AFAI-bil (AFAI algorithm with bilinear kernel), AFAI-bic (AFAI algorithm with bilinear kernel). We choose $T_l = 20, T_h = 235$.

2. Discard the results with severe ringing artifacts. Ringing pixels are detected as pixels which significantly exceed the original dynamic range, i.e. $N_r = \# I_h(u,v) > T_h + \alpha(T_h - T_l)$ or $I_h(u,v) < T_l - \alpha(T_h - T_l)$. Discard the result image with $N_r > 0.001N_e$ where N_e is number of edge pixels which are detected by canny detector. We choose $\alpha = 0.02$.

3. Choose the sharpest result in the remaining ones. Sharpness is measured by the high frequency energy, $sharpness = \sum_{u,v}[(H * I_h)(u,v)]^2$, where H is a high pass filter and $H * I_h$ is the convolution of H and I_h. We choose Sobel filter as H.

4. Restore the dynamic range of final HR image with $I_h = (I_h' - b)/k$.



(a) Ringing effects. LR images are produced by bicubic downsampling. Left: AFAI-bic, correct kernel; Right: AFAI-bil, wrong kernel.

(b) Blurring effects. LR images are produced by bilinear downsampling. Left: AFAI-bil, correct kernel; Right: AFAI-bic, wrong kernel.

Fig. 1: Ringing and blurring effects because of wrong kernel assumed.

Accuracy of the above algorithm is about 70%. Note that although the accuracy is not high, for those images that our algorithm fails, we observe that choosing a wrong kernels does not cause severe visual artifacts. Here we compare only bilinear and bicubic kernel here because they are two representative convolution kernels. One can also add more kernels to compare, e.g. average kernel.

5. EXPERIMENT RESULTS

Table 1 shows the PSNR results of applying different interpolation methods for zooming factor of 2.5 when LR images are produced by bilinear downsampling and Table 2 shows the average PSNR results when LR images are produced

Table 1: PSNR (dB) results with **bilinear** downsampled images (2.5×). **gain** is the PSNR gain of AFAI-bil over bicubic.

Images	bicubic	SAI+bic	AFAI-bic	AFAI-bil	gain
baboon	23.07	22.99	23.79	**24.07**	1.00
bike	22.94	23.19	24.32	**24.83**	1.89
flower	21.23	20.95	21.96	**22.31**	1.08
lena	31.28	31.45	33.02	**33.78**	2.50
necklace	19.86	19.42	20.89	**21.30**	1.43
parrot	31.01	31.19	32.46	**33.02**	2.01
building	22.75	23.12	24.65	**25.34**	2.59
tree	25.58	25.47	26.94	**27.55**	1.98
average	24.71	24.72	26.00	**26.52**	1.81

Table 2: PSNR (dB) results with **bicubic** downsampled images (2.5×). **gain** is the PSNR gain of AFAI-bic over bicubic.

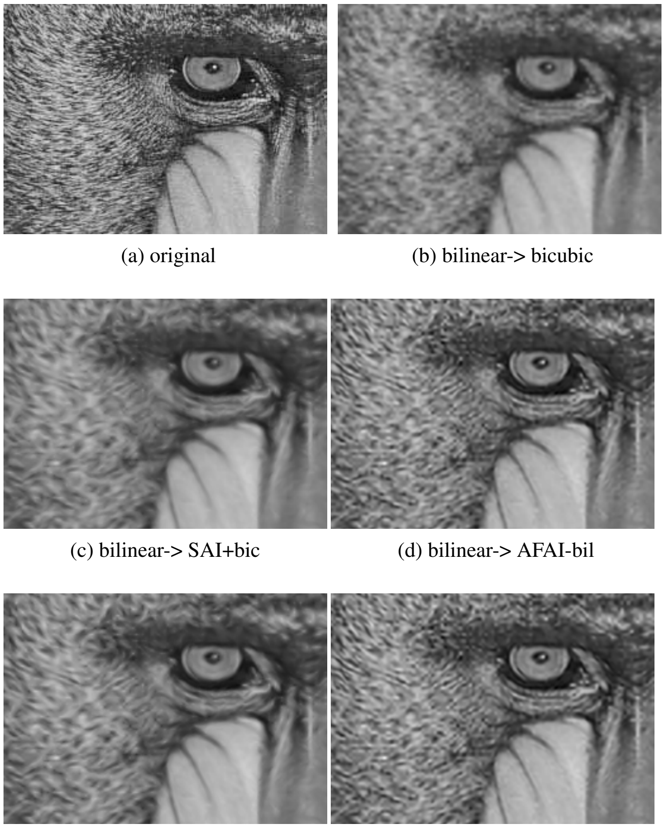
	bicubic	SAI+bic	AFAI-bil	AFAI-bic	gain
average	25.32	25.17	26.19	**26.55**	1.23

by bicubic downsampling. We can see that our AFAI-bil performs best for all images. On average we have around 1.8 dB gain over the bicubic for bilinear downsampled images, and over 1.2 dB gain for bicubic downsampled images. Visually AFAI results are much sharper and more natural, see Fig. 2(d) and (f), where (d) is AFAI-bil result for bilinear downsampled image and (f) is AFAI-bic result for bicubic downsampled image. More results are available on http://ihome.ust.hk/~tkt/research.html.

PSNR of AFAI-bil is 0.5 dB higher than AFAI-bic from Table 1, and AFAI-bic is 0.36 dB higher than AFAI-bil from Table 2. This observation supports our argument that the assumed downsampling kernel should match the real downsampling kernel. Besides, although the visually looks better, the average PSNR results of SAI+bic (SAI interpolation by factor of 2, followed by bicubic interpolation) is not better than bicubic, even worse for bicubic downsampled images. One reason is that SAI does not assume any blurring kernel. Another reason is that the bicubic interpolation in the second step blurs the results eventually. This supports our arguments an integer-factor adaptive interpolation plus a convolution-based interpolation is not sufficient for arbitrary factor interpolation.

6. CONCLUSION

In this paper we propose an effective arbitrary factor interpolation algorithm which combines adaptive 2-D autoregressive modeling and convolution-based kernel constraint. With the proposed AFAI algorithm, the result HR images are smooth along edge direction and sharp across edge direction. Our algorithm is capable of performing arbitrary factor interpolation with arbitrary convolution kernel. Experiment results show that while achieving arbitrary factor interpolation, our proposed algorithm also achieves higher visual quality than convolution-based interpolation only as well as conventional adaptive interpolation method followed by convolution-based interpolation.



(a) original

(b) bilinear-> bicubic

(c) bilinear-> SAI+bic

(d) bilinear-> AFAI-bil

(e) bicubic-> SAI+bic

(f) bicubic-> AFAI-bic

Fig. 2: Results of different interpolation methods applied on **baboon** images with zooming factor 2.5. bilinear-> bicubic means the LR image is produced by bilinear kernel and HR result is produced by bicubic method, and so on.

Acknowledgment

This work has been supported in part by the Research Grants Council (RGC) of the Hong Kong Special Administrative Region, China (GRF 610109 and GRF 610112).

7. REFERENCES

[1] R. Keys, "Cubic convolution interpolation for digital image processing," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 29, no. 6, pp. 1153–1160, 1981.

[2] C. E. Duchon, "Lanczos filtering in one and two di-

mensions," *Journal of Applied Meteorology*, vol. 18, pp. 1016–1022, Aug. 1979.

[3] B. Huang, E. M. K. Lai, and A. P. Vinod, "Image resizing and rotation based on the consistent resampling theory," in *Intelligent Signal Processing and Communications Systems, 2008. ISPACS 2008. International Symposium on*, 2009, pp. 1–4.

[4] X. Zhang and X. Wu, "Image interpolation by adaptive 2-d autoregressive modeling and soft decision estimation," *Image Processing, IEEE Transactions on*, vol. 17, no. 6, pp. 887–896, 2008.

[5] X. Li and M. T. Orchard, "New edge-directed interpolation," *Image Processing, IEEE Transactions on*, vol. 10, no. 10, pp. 1521–1527, 2001.

[6] Z. Lei and W. Xiaolin, "An edge-guided image interpolation algorithm via directional filtering and data fusion," *Image Processing, IEEE Transactions on*, vol. 15, no. 8, pp. 2226–2238, 2006.

[7] D. D. Muresan, "Fast edge directed polynomial interpolation," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 2, 2005, pp. II–990–3.

[8] D. D. Muresan and T. W. Parks, "Adaptively quadratic (aqua) image interpolation," *Image Processing, IEEE Transactions on*, vol. 13, no. 5, pp. 690–698, 2004.

[9] K. Tang, O. C. Au, L. Fang, Z. Yu, and Y. Guo, "Image interpolation using autoregressive model and gauss-seidel optimization," in *Image and Graphics (ICIG), 2011 Sixth International Conference on*, 2011, pp. 66–69.

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[11] W. Dong, L. Zhang, and G. Shi, "Centralized sparse representation for image restoration," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 1259–1266.