EDGE DIRECTION-BASED SIMPLE RESAMPLING ALGORITHM

Gwanggil Jeon, Joohyun Lee, Wonkyun Kim, and Jechang Jeong
Department of Electronics and Computer Engineering, Hanyang University
17 Haengdang-dong, Seongdong-gu, Seoul, Korea
{windcap315, saint81, wonkyun, jjeong}@ece.hanyang.ac.kr

ABSTRACT

We considered the problem of resampling video frames to adjust from 176×72 to 352×288 display formats. Images were divided into six regions according to six different edge directions and different reconstruction techniques were employed in each region. The algorithms developed and implemented on this system include edge pattern-based region classifier and resampling strategies for digital display. The performance of the proposed algorithm is compared to conventional methods such as nearest-neighbor interpolation, bilinear interpolation, and simple cubic curve fitting interpolation.

Index Terms — Resampling, upscaling, edge direction, directional interpolation.

1. INTRODUCTION

Resampling (or interpolation) is the process of sampling one or more input pixels to create one or more output pixels. Image resampling is employed in a wide variety of image manipulation tasks such as image registration, image warping, and image scaling. In general, the quality of the output image and the time required for calculation are highly related to the resampling method selection.

Resampling algorithms for images have been reported to be useful in a wide range of applications [1-3]. Changing image resolution is a common operation. Several simple and efficient approaches for analyzing and reducing the aliasing effect in enlargements, caused by resampling methods, have been proposed. For example, nearest-neighbor interpolation (NNI) [4], bilinear interpolation (BI) [5], simple cubic curve fitting interpolation (SCCF) [6], and other various resampling functions are frequently used in image resampling. We have proposed a resampling technique based on adaptive method selection.

It has been shown that many conventional image-upscaling algorithms often yield blurred results, while more complicated methods are also more time-consuming. For example, neither the NNI nor the BI method is very computationally complexity, but the NNI method provides blocky edges and the BI method provides blurred images. Moreover, these techniques apply the same resampling procedure to the whole image indiscriminately. In other words, conventional methods interpolate missing pixels in the same manner. On the other hand, the adaptive method is designed to avoid those issues by analyzing the local structures of the source image and by using different resampling functions with different areas of support. As mentioned previously, our proposed method is an adaptive method.

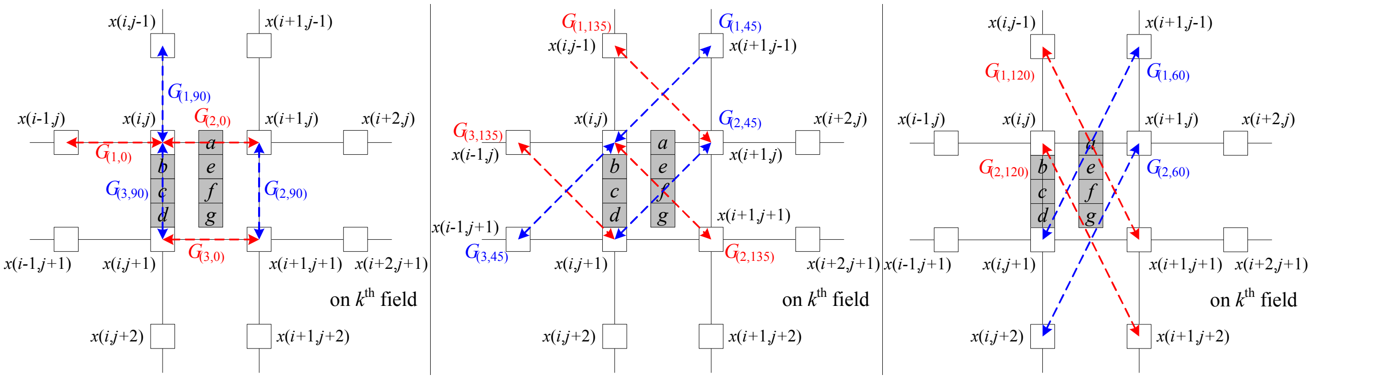
In this paper, we present an adaptive resampling algorithm for transforming images from 176×72 to 352×288. The algorithm is based on segmenting the image dynamically into six regions, and adaptively interpolating missing regions. The remainder of this paper is structured as follows. Section 2 presents a brief review of previous methods and arguments for the importance of the proposed edge direction-based resampling algorithm. In Section 3, the experimental results and performance analysis are provided to show the feasibility of the proposed approach. The paper concludes with a look at possible further investigations in Section 4.

2. PROPOSED ALGORITHM

2.1. Previous Resampling Algorithms

Resampling is an image processing method to increase or decrease the number of pixels in a digital image. The most important task is to smoothly enlarge images without introducing artifacts. Several resampling methods have been proposed such as NNI, BI, and SCCF. The main advantage of the NNI method is its simplicity, which makes it the most efficient of all these implementations. However, strong aliasing and blurring effects are associated with this method. The BI method reproduces at most a first-degree polynomial, but its use results in the attenuation of high frequency and aliasing into low frequencies.

1-4244-1437-7/07/$20.00 ©2007 IEEE



(a) 0° and 90° edge direction (b) 45° and 135° edge direction (c) 60° and 120° edge direction

Fig. 1. Pixel window and illustration of the EDR algorithm.

2.2. Investigation of Edge Pattern

Fig. 1 shows the proposed edge direction-based simple resampling algorithm (EDR), where i refers to the column number, j refers to the line number, and k refers to the field number. $x(i,j,k)$ is the estimate of the intensity of the pixel at location (i,j,k). Pixel sets $\{a, b, c, d, e, f, \text{and } g\}$ are pixels to be resampled using existing neighbor pixels. The parameter $\Gamma_{\Omega,\Psi}$ denotes a directional correlation measurement, which is an intensity change in the direction, classified by $\Omega(\in\{1, 2, 3\})$ and edge direction $\Psi(\in\{0^\circ, 45^\circ, 60^\circ, 90^\circ, 120^\circ, 135^\circ\})$. $\Gamma_{\Omega,\Psi}$ is used to determine the direction of the highest spatial or temporal correlation. If the parameter Ω is 1, the output of the direction-based algorithm is obtained as in (1). In the same manner, other gradients $\{\Gamma_{(1,90)}, \Gamma_{(2,90)}, \Gamma_{(3,90)}\}$, $\{\Gamma_{(1,45)}, \Gamma_{(2,45)}, \Gamma_{(3,45)}\}$, $\{\Gamma_{(1,135)}, \Gamma_{(2,135)}, \Gamma_{(3,135)}\}$, $\{\Gamma_{(1,60)}, \Gamma_{(2,60)}\}$, and $\{\Gamma_{(1,120)}, \Gamma_{(2,120)}\}$ are calculated. The maximum gradients among the $\Gamma_{\Omega,\Psi}$'s are represented as μ_Ψ.

$$\begin{aligned}\Gamma_{(1,0)} &= |x(i-1,j)-x(i,j)|\\ \Gamma_{(2,0)} &= |x(i,j)-x(i+1,j)|\\ \Gamma_{(3,0)} &= |x(i,j+1)-x(i+1,j+1)|\end{aligned} \tag{1}$$

$$\begin{aligned}\mu_0 &= \max(\Gamma_{(1,0)},\Gamma_{(2,0)},\Gamma_{(3,0)})\\ \mu_{45} &= \max(\Gamma_{(1,45)},\Gamma_{(2,45)},\Gamma_{(3,45)})\\ \mu_{60} &= \max(\Gamma_{(1,60)},\Gamma_{(2,60)})\\ \mu_{90} &= \max(\Gamma_{(1,90)},\Gamma_{(2,90)},\Gamma_{(3,90)})\\ \mu_{120} &= \max(\Gamma_{(1,120)},\Gamma_{(2,120)})\\ \mu_{135} &= \max(\Gamma_{(1,135)},\Gamma_{(2,135)},\Gamma_{(3,135)})\end{aligned} \tag{2}$$

The key to the success of the EDR method is an accurate estimation of edge direction. The edge pattern appears along six different directions. The dominant edge with edge direction (Θ_Ψ) across $x(i,j,k)$ is calculated as (3).

$$\Theta_\Psi = \min(\mu_0, \mu_{45}, \mu_{60}, \mu_{90}, \mu_{120}, \mu_{135}) \tag{3}$$

2.3. Cubic Curve Fitting Method

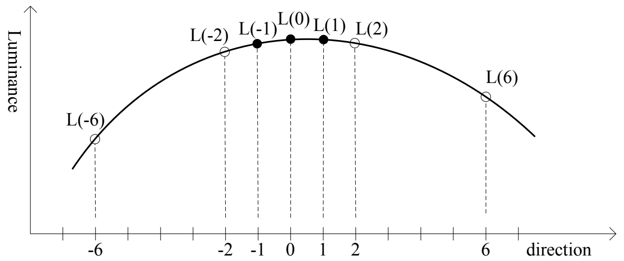


Fig. 2. Luminance transition in a direction of obtaining $L(-1)$, $L(0)$, and $L(1)$. Pixels $L(-6)$, $L(-2)$, $L(2)$, and $L(6)$ are known pixels.

Fan *et al.* introduced a cubic curve fitting algorithm [6]. We adopted and slightly expanded upon the above method, creating the modified cubic curve fitting (MCCF) algorithm. We assume that the luminance transition in the horizontal (i) or vertical (j) direction can be approximated as a third order function.

We regard $L(m)=\alpha+\beta m+\gamma m^2+\delta m^3$ as a third order function of m and assumed that $L(-1)$, $L(0)$, and $L(1)$ are the pixels to be resampled and $L(-6)$, $L(-2)$, $L(2)$, and $L(6)$ correspond to four sample pixels of the original field. With these function values already known, the four equations, $L(-6)=\alpha-6\beta+36\gamma-216\delta$, $L(-2)=\alpha-2\beta+4\gamma-8\delta$, $L(2)=\alpha+2\beta+4\gamma+8\delta$, and $L(6)=\alpha+6\beta+36\gamma+216\delta$, can be calculated by simple substitutions of the value of m. Through the above equations, pixels $L(-1)$, $L(0)$, and $L(1)$ can be written as (4).

$$\begin{bmatrix}L(-1)\\L(0)\\L(1)\end{bmatrix}=\begin{bmatrix}1&-1&1&-1\\1&0&0&0\\1&1&1&1\end{bmatrix}\begin{bmatrix}\alpha\\\beta\\\gamma\\\delta\end{bmatrix}=\begin{bmatrix}1&-1&1&-1\\1&0&0&0\\1&1&1&1\end{bmatrix}\begin{bmatrix}1&-6&36&-216\\1&-2&4&-8\\1&2&4&8\\1&6&36&216\end{bmatrix}^{-1}\begin{bmatrix}L(-6)\\L(-2)\\L(2)\\L(6)\end{bmatrix}=\begin{bmatrix}-0.0547&0.8203&0.2734&-0.0391\\-0.0625&0.5625&0.5625&-0.0625\\-0.0391&0.2734&0.8203&-0.0547\end{bmatrix}\begin{bmatrix}L(-6)\\L(-2)\\L(2)\\L(6)\end{bmatrix} \tag{4}$$

2.4. Resampling Process

According to dominant edge detector, any edge direction can be assigned to the region. The classifier decides where the pixel is located with respect to the edge directions $\{\Theta_1(=0^{o}), \Theta_2(=45^{o}), \Theta_3(=60^{o}), \Theta_4(=90^{o}), \Theta_5(=120^{o})$, or $\Theta_6(=135^{o})\}$.

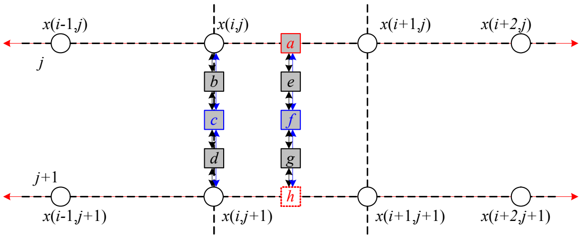


Fig. 3. Directional resampling in the region with 0^{o} direction.

If the region is classified into Θ_1, we assume that there is a horizontal direction edge through the region. Pixel *a* is calculated using the MCCF method in the horizontal direction and pixels *b*, *c*, and *d* are obtained using the weighted BI method in the vertical direction. To interpolate pixels *e*, *f*, and *g*, we first estimate pixel *h* using the MCCF method in the horizontal direction and calculate *e*, *f*, and *g* by using the weighted BI method in the vertical direction.

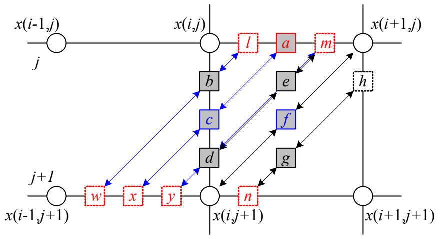


Fig. 4. Directional resampling in the region with 45^{o} direction.

If the region is classified into Θ_2, we assume that there is a 45^{o} diagonal direction edge through the region. Pixels *l*, *a*, and *m* are obtained using the MCCF method in the horizontal direction over the j^{th} row and pixels *w*, *x*, *y*, and *n* are obtained using the MCCF method in the horizontal direction over the $j+1^{th}$ row. Finally, we obtain the pixels *b*, *c*, *d*, *e*, *f*, and *g* using the weighted BI method in the 45^{o} direction edge as shown in Fig. 4. In the same manner, all pixels in the region with Θ_6 are calculated.

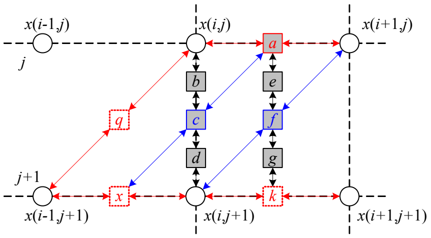


Fig. 5. Directional resampling in the region with 60^{o} direction.

If the region is classified into Θ_3, we assume that there is a 60^{o} diagonal direction edge through the region. Pixels *a*, *q*, *x*, and *k* are calculated using the BI method according to the line in red. Pixels *c* and *f* are calculated using the BI procedure according to the line in blue. The other pixels *b*, *d*, *e*, and *g* are calculated using the BI method in the vertical direction. In the same manner, all pixels in the region with Θ_5 are calculated.

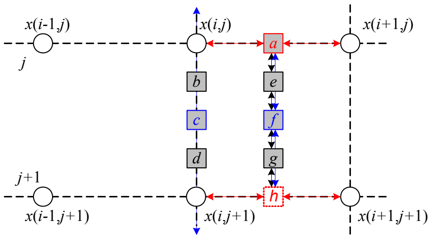


Fig. 6. Directional resampling in the region with 90^{o} direction.

If the region is classified into Θ_4, we assume that there is a vertical direction edge through the region. Pixel *a* is calculated using the BI method in the horizontal direction and pixels *b*, *c*, and *d* are obtained using the MCCF procedure in the vertical direction. To interpolate pixels *e*, *f*, and *g*, we estimate pixel *h* using the BI method in the horizontal direction and calculate *e*, *f*, and *g* using the weighted BI method in the vertical direction.

3. EXPERIMENTS AND LIMITATION

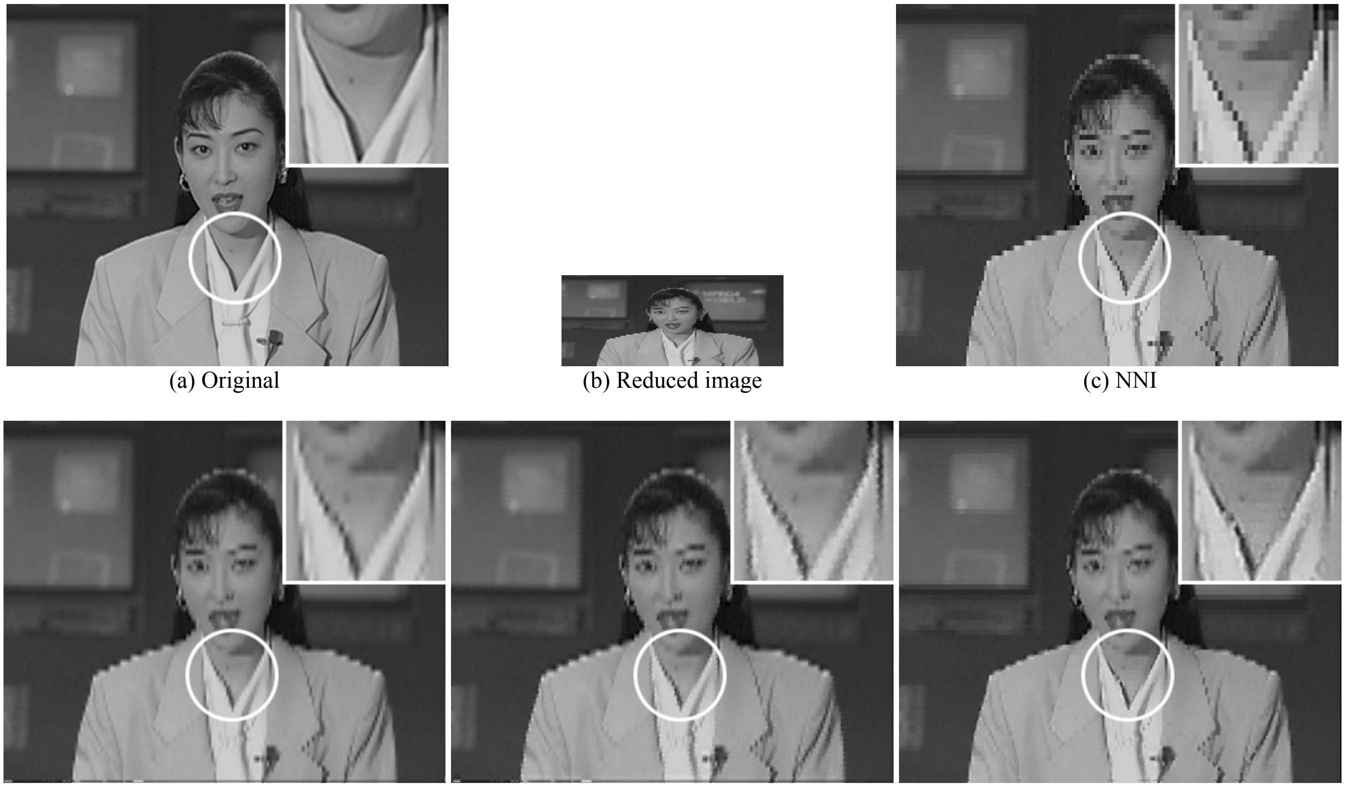
In this section, we apply three previously-devised methods (NNI, BI, and SCCF) and the proposed EDR method. The performance of the EDR method is evaluated on 352×288 CIF images. To compare the proposed algorithm with previous techniques, the traditional peak signal-to-noise ration (PSNR) is used as the objective measure, as shown in Table 1. The PSNR performance improvement over NNI ranges from 2.61 to 4.16 dB, with an average improvement of 3.29 dB, while requiring 3.85 times the complexity.

Table 1. PSNR and Average CPU time (seconds/frame) results of different upsampling methods for four CIF sequences.

Method	NNI	BI	SCCF	EDR
Akiyo	26.34005	29.33518	29.28945	29.70613
	0.01575	0.03525	0.08200	0.09673
Flower	16.77440	19.06718	18.94540	19.38940
	0.03150	0.03075	0.10950	0.11792
Foreman	23.12623	26.88470	26.90478	27.28673
	0.02000	0.03100	0.07825	0.10980
Stefan	17.57695	20.36808	20.07708	20.60720
	0.02350	0.02750	0.10150	0.10050

(units: dB, s)

To compare the subjective qualities, simulation results using the Akiyo image are shown in Fig. 7. They show the comparisons of the recovered Akiyo images from the previous methods and the proposed EDR method. Fig. 7(a) is the original image. The image in Fig. 7(b) was reduced by a factor of 4 by 2. We can see severe artifacts produced by the NNI and BI methods in Figs. 7(c) and 7(d), respectively. We see fewer artifacts in Figs. 7(e) and 7(f). We clearly see



(a) Original (b) Reduced image (c) NNI

(d) BI (e) SCCF (f) Proposed method

Fig. 7. An image of Akiyo sequence reduced by a factor of four (horizontal) and two (vertical) using four different reconstructions.

there is a staircase artifact in Fig. 7(e) that resembles the NNI method with a blurred effect. Observation shows that the proposed EDR method has achieved noticeable improvements. It can be seen that the resampled image quality after using the proposed algorithm is improved in most of the sequences, especially around edges such as the shirt collar. From these results, we conclude that the EDR method is a good and efficient approach for substantially reducing artifact effects in enlargements. However, it is still debatable whether the gained quality advantage justifies the increased computational burden.

4. CONCLUSION

In this paper, we proposed a novel resampling method that is performed by segmenting the image dynamically into regions with six categories, as it is scanned or received. This method combines the advantage of using bilinear interpolation, cubic curve fitting interpolation, and a direction-oriented interpolation algorithm. The proposed resampling algorithm results in the reduced occurrence of staircase artifacts, providing satisfactory image quality. Algorithm performance in terms of objective and subjective qualities and computational complexity aspects was compared to different methods and functions previously reported in the literature. The advantage of the algorithm is quite apparent at the image edges.

ACKNOWLEDGMENT

This work was sponsored by ETRI SoC Industry Promotion Center, Human Resource Development Project for IT SoC Architect.

5. REFERENCES

[1] D. Seidner, “Polyphase Antialiasing in Resampling of Images,” *IEEE Trans. Image Processing*, vol. 14, no. 11, pp. 1876-1889, Nov. 2005.

[2] J. A. Malpica, “Splines Interpolation in High Resolution Satellite Imagery,” *Lecture Note in Computer Science,* vol. 3084, pp. 562-570, 2005.

[3] K. Jack, Video Demystified - *A Handbook for the Digital Engineer*, Elsevier 2005.

[4] C. J. Veenman, M. J. T. Reinder, “The Nearest Subclass Classifier: A Compromise between the Nearest Mean and Nearest Neighbor Classifier,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 9, pp. 1417-1429, Sept. 2005.

[5] E. B. Bellers and G. de Haan, De-interlacing – *A Key Technology for Scan Rate Conversion*, Elsevier, 2000.

[6] Y. -C. Fan, H. -S. Lin, H. -W. Tsao, and C. -C. Kuo, “Intelligent intra-field interpolation for motion compensated deinterlacing,” in *Proc. ITRE 2005*, vol. 3, pp. 200-203, June 2005.