DOI: 10.2352/ISSN.2470-1173.2016.15.IPAS-026

Edge-directional interpolation algorithm using structure tensor

Andrey V. Nasonov[1], Andrey S. Krylov[1], Xenya Yu. Petrova[2], Michael N. Rychagov[2];
[1] ***Lomonosov Moscow State University, Faculty of Computational Mathematics and Cybernetics, Laboratory of Mathematical Methods of Image Processing***
[2] ***Samsung R&D Institute Russia, Algorithm Lab, 12, Dvintsev str., Moscow, Russia 127018***

Abstract

The paper presents a new low complexity edge-directed image interpolation algorithm. The algorithm uses structure tensor analysis to distinguish edges from textured areas and to find local structure direction vectors. The vectors are quantized into 6 directions. Individual adaptive interpolation kernels are used for each direction. Experimental results show high performance of the proposed method without introducing artifacts.

Introduction

Development of a high-quality and detail-preserving image interpolation algorithm is a challenging problem. Linear methods [1] like bilinear and bicubic interpolation show great performance but they suffer from blur, ringing and staircase artifacts in the edge areas. Additional knowledge about image contents is used by more advanced image interpolation algorithms. For example, NEDI algorithm [2] uses the assumption of self-similarity between high- and low-resolution images.

Regularization-based image interpolation algorithms pose the image interpolation as a functional minimization problem [3, 4]. The functional contains the data-fitting term and the stabilizer term. The data-fitting term restricts the high-resolution image to match the low-resolution image. The stabilizer term makes the high-resolution image fit an a priori information. The similar approaches are MAP, POCS and PDE-based algorithms [5, 6, 7, 8, 9]. Learning-based algorithms construct the high-resolution image using a pre-built dictionary containing pairs of corresponding high- and low-resolution patches [10, 11].

Regularization-based algorithms are time consuming as they perform the minimization of the regularization functional by iterative methods. Non-iterative edge-directional image interpolation algorithms are developed for the performance critical applications.

Great effectiveness has been shown by single-frame super-resolution algorithms which map low-resolution image patches into high-resolution ones. Deep convolutional neural networks [12, 13] and regression [14, 15] are used for high quality image interpolation.

High effectiveness has been also shown by low complexity edge-directed image interpolation algorithms that consider the image resampling procedure as two consecuent problems [16, 17, 18, 19]. The first problem is finding the direction for each pixel corresponding to local image structure. The second problem is directional interpolation according to previously found directions. Existing algorithms use directional filtering [16], directional variation [17], second order derivatives [18] to find local structure direction. Existing algorithms also limit the number of possible directions to 2 or 4 to improve computational efficiency and to reduce the influence of discretization. For example, the algorithm [19] combines the results of applying directional cubic interpolation in two directions.

Textured areas are a problem for edge-directed interpolation methods. Artifacts usually appear when edge-directed algorithm is applied to corners. Corners contain multiple directions and usually appear in textures areas. Using the structure tensor is an effective way to distinguish between edges, corners and flat areas.

In this work, we propose fast and effective edge-directional algorithm based on structure tensor and individual interpolation kernels for each direction. The main difference of the proposed algorithm with state-of-the-art algorithms is quantization of the direction vector into 6 directions and using optimal 4x4 kernels for each direction. The kernels are optimized by PSNR minimization over 29 reference images from LIVE database [20].

Algorithm detail

The algorithm consists of the following steps:

1. Initial approximation of the high-resolution image.
2. Finding direction for every pixel of a high-resolution image.
3. First interpolation step: the interpolation of a central pixel inside every 4x4 block of low-resolution pixels.
4. Second interpolation step: the interpolation for the rest of pixels using the approach from the first interpolation step.

Initial approximation

Initial approximation of the high-resolution image is used to find directions for every interpolated pixel.

Experiments have shown that directions obtained at the next step practically do not depend on the used interpolation method. For simplicity, we use standard bicubic interpolation method to construct the initial approximation of the high-resolution image. Let u be the input low-resolution image, v be the interpolated image. The

DOI: 10.2352/ISSN.2470-1173.2016.15.IPAS-026

approximation looks as:

$$v_{2i,2j} = u_{i,j};$$
$$v_{2i+1,2j} = \frac{9}{16}\left(u_{i,j} + u_{i+1,j}\right) - \frac{1}{16}\left(u_{i-1,j} + u_{i+2,j}\right);$$
$$v_{i,2j+1} = \frac{9}{16}\left(v_{i,2j} + v_{i,2j+2}\right) - \frac{1}{16}\left(v_{i,2j-2} + v_{i,2j+4}\right).$$

Finding directions

Construction of the structure tensor

The structure tensor has the following matrix:

$$T_{i,j} = \begin{bmatrix} \langle v_x \rangle_{i,j}^2 & \langle v_x v_y \rangle_{i,j} \\ \langle v_x v_y \rangle_{i,j} & \langle v_y \rangle_{i,j}^2, \end{bmatrix} \tag{1}$$

where v_x and v_y are partial derivatives of v, $\langle \ldots \rangle_{i,j}$ is averaging over small neighborhood of the pixel (i, j).

We calculate the partial derivatives as follows:

$$v_x = v * h_1(x) * h_2(y),$$
$$v_x = v * h_2(x) * h_1(y),$$

where h_1 and h_2 are Gaussian filter and shifted derivatives of Gaussian filter respectively:

$$h_1(t) = \exp\left(-\frac{(t+0.5)^2}{2\sigma_1^2}\right),$$
$$h_2(t) = -(t+0.5)\exp\left(-\frac{(t+0.5)^2}{2\sigma_2^2}\right),$$

where $\sigma_1 = \sigma_2 = 0.5$.

The averaging $\langle \ldots \rangle_{i,j}$ is the convolution with shifted Gaussian filter with kernel

$$\exp\left(-\frac{(t-0.5)^2}{2\sigma_3^2}\right)$$

with $\sigma_3 = 1.5$.

Normalization of kernels h_1, h_2, h_3 is not necessary here. Half-pixel shift is used to improve the accuracy of the method when applied to discrete images.

Structure tensor analysis

Local structure directions are obtained using the analysis of eigenvectors and eigenvalues of the matrix (1).

Let λ_1, λ_2 be the eigenvalues of T such that $|\lambda_1| \geq |\lambda_2|$ and $\vec{p}$ be the eigenvector corresponding to λ_1. The ratio between λ_1 and λ_2 defines the type of structural element in the analyzed pixel. If $|\lambda_1|$ is significantly greater than $|\lambda_2|$, then the pixel is a part of linear structure like edge or ridge with the direction $\vec{p}$ and edge-directional interpolation will be effective. Otherwise, if $|\lambda_1| \sim |\lambda_2|$ or $\lambda_1 \sim 0$, then there is no dominant direction in the analyzed pixel.

The direction of $\vec{p}$ is quantized into one of the 6 directions (0, 30, 60, 90, 120 and 150 degrees).

The output of the finding directions step is the following (see Fig. 1): 1. If $|\lambda_1| \leq 2|\lambda_2|$ or $|\vec{p}| \sim 0$, there is no distinct direction in the analyzed pixel, the output is zero. 2. Otherwise, the output is the direction index of $\vec{p}$ (1 to 6 range).

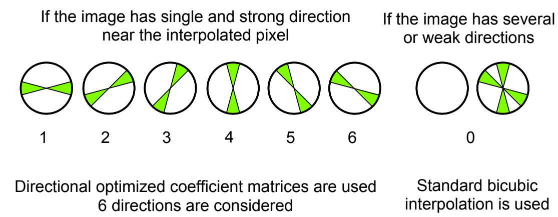

***Figure 1.** Finding edge directions using the structure tensor*

First interpolation step

At the first interpolation step, pixels with coordinates $(2i, 2j)$ are copied directly from the low-resolution image:

$$v_{2i,2j} = u_{i,j},$$

while pixels with coordinates $(2i+1, 2j+1)$ are updated using 4x4 block of surrounding pixels from the low-resolution image (see Fig. 2).

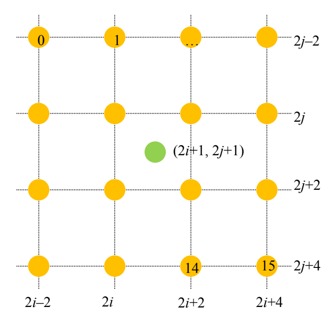

***Figure 2.** The first interpolation step. The interpolated pixel is in the center of 4x4 grid of low-resolution Pixels of low-resolution image are orange; the interpolated pixel is green.*

Let a be the vector constructed from 16 pixels from the 4x4 block, $q^{(d)}$ be the interpolation kernel corresponding to the direction d. The value of the interpolated pixel is computed as

$$v_{2i+1,2j+1} = \sum_{k=0}^{15} a_{n,k} q_k^{(d)}. \tag{2}$$

The interpolation kernels $q^{(d)}$ have been calculated experimentally using the reference images from LIVE database [20]. The reference images were downsampled by 2 times then a set of correspondences between vectors a and values v was constructed for each direction d for all pixels. The interpolation kernels $q^{(d)}$ were obtained by minimizing the squared error sum

$$\sum_n \left(v_n - \sum_{k=0}^{15} a_{n,k} q_k^{(d)}\right)^2.$$

DOI: 10.2352/ISSN.2470-1173.2016.15.IPAS-026

Restrictions based on kernel symmetry and rotation are added to minimize the number of coefficients and to reduce the condition number of the least squares minimization problem. The restrictions include:

1. Central symmetry of all the kernels: $q_k^{(d)} = q_{15-k}^{(d)}$, horizontal and vertical symmetry for $q^{(1)}$ and $q^{(4)}$, 4-directional symmetry for $q^{(0)}$.

2. The kernel $q^{(4)}$ is equal to the kernel $q^{(1)}$ with 90 degree rotation.

3. The kernels $q^{(3)}$, $q^{(5)}$, $q^{(6)}$ are equal to the kernel $q^{(2)}$ with 90 degree rotation and transposition.

The coefficient values of the kernels calculated for the images from LIVE database are presented at the website: `http://imaging.cs.msu.ru/en/publication?id=319`

Second interpolation step

The second interpolation step is similar to the first step but instead of 4x4 block of pixels of the low-resolution image, the rotated by 45 degrees 4x4 pixel block is used. It contains both pixels from low-resolution image and pixels interpolated at the previous step.

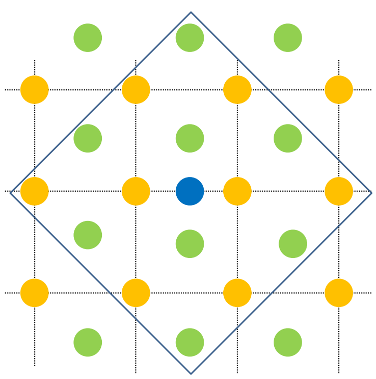

Figure 3. *The second interpolation step. Pixels of low-resolution image are orange; pixels interpolated at the previous step are green.*

Results

The performance of the proposed algorithm is shown in Fig. 4. Objective quality comparison using PSNR and SSIM metrics is presented in Fig. 5. The algorithm DCCI [19] is used in the comparison as it has been shown great performance among low complexity state-of-the-art algorithms [21]. It can been seen that the proposed algorithm shows quality improvement even for highly textured images like Goldhill and Baboon. For low-textured images like Cameraman, the proposed algorithm shows worse PSNR but better SSIM than DCCI.

Although modern image interpolation algorithms based on LR-to-HR mapping [12, 13, 14, 15] show better quality than the proposed algorithm, they have significantly higher computational complexity. Also they may produce flicking and vibration of fine details when applied for video containing noise. The proposed algorithm does not produce such effects. It is suitable for video resampling and can be used as part of multi-frame super-resolution algorithms.

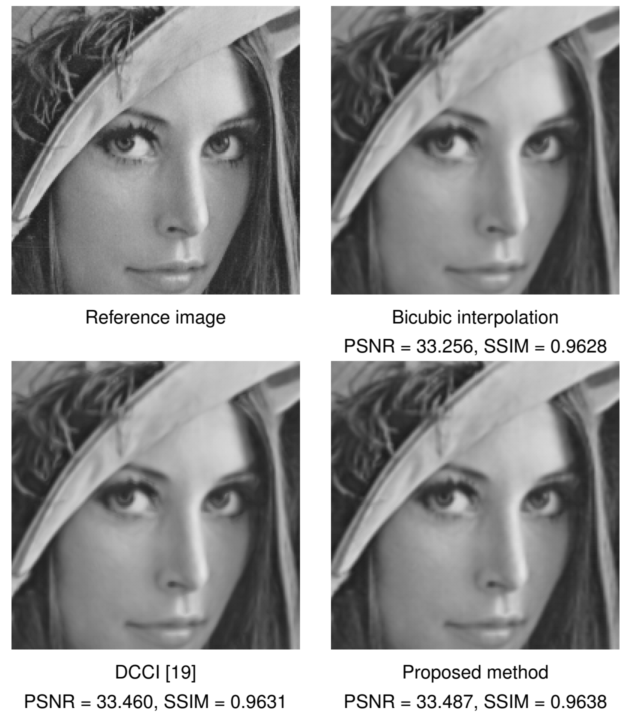

Figure 4. *The results of the proposed image interpolation algorithm*

Image	Bicubic		DCCI [19]		Proposed	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Lena	33.256	0.9629	33.460	0.9632	**33.487**	**0.9638**
Peppers	32.101	0.9520	32.226	0.9518	**32.232**	**0.9523**
Goldhill	30.949	0.9243	30.870	0.9224	**30.950**	**0.9244**
Baboon	23.581	0.7972	23.585	0.7965	**23.612**	**0.7994**
Cameraman	25.499	0.9152	**25.678**	0.9163	25.619	**0.9169**

Figure 5. *Objective quality comparison*

Conclusion

A new low complexity edge-directed image interpolation algorithm has been developed. The algorithm has shown great performance and quality in comparison to state-of-the-art image interpolation algorithms. The proposed algorithm does not introduce artifacts in textured areas. The algorithm has a great potential to be used for video interpolation and multi-frame super-resolution due to robustness of the directions obtained from the structure tensor to noise.

References

[1] Blu, T., Thevenaz, P., Unser, M.: Linear interpolation revitalized. IEEE Transactions on Image Processing **13**(5) (2004) 710–719

[2] Li, X., Orchrard, M.: New edge-directed interpolation. IEEE Transactions on Image Processing **10**(10) (2001) 1521–1527

DOI: 10.2352/ISSN.2470-1173.2016.15.IPAS-026

[3] Liu, Z.: Adaptive regularized image interpolation using a probabilistic gradient measure. Optics Communications **285**(3) (2012) 245–248

[4] Farsiu, S., Robinson, M., Elad, M., Milanfar, P.: Fast and robust multiframe super resolution. IEEE Transactions on Image Processing **13**(10) (2004) 1327–1344

[5] Pan, M., Lettington, A.: Efficient method for improving poisson MAP super-resolution. Electronic Letters **35**(10) (1999) 803–805

[6] Belekos, S., Galatsanos, N., Katsaggelos, A.: Maximum a posteriori video super-resolution using a new multichannel image prior. IEEE Transactions on Image Processing **19**(6) (2010) 1451–1464

[7] Morse, B., Schwartzwald, D.: Isophote-based interpolation. Proceedings of ICIP **3** (1998) 227–231

[8] Chan, T., Shen, J.: Image Processing And Analysis: Variational, PDE, Wavelet, And Stochastic Methods. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2005)

[9] Ratakondo, K., Ahuja, N.: Pocs based adaptive image magnification. Proceedings of ICIP (1998) 203–207

[10] Peleg, T., Elad, M.: A statistical prediction model based on sparse representations for single image super-resolution. IEEE Transactions on Image Processing **23**(6) (2014) 2569–2582

[11] Nasrollahi, K., Moeslund, T.B.: Super-resolution: a comprehensive survey. Machine Vision and Applications **25**(6) (2014) 1423–1468

[12] Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. European Conference on Computer Vision (2014) 184–199

[13] Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE Transactions on Pattern Analysis and Machine Intelligence (2015) preprint

[14] Timofte, R., De, V., Gool, L.V.: A+: adjusted anchored neighborhood regression for fast super-resolution. Asian Conference of Computer Vision (2014) 111–126

[15] Dai, D., Timofte, R., Gool, L.V.: Jointly optimized regressors for image super-resolution. Eurographics **34** (2015)

[16] Zhang, L., Wu, X.: An edge-guided image interpolation algorithm via directional filtering and data fusion. IEEE Transactions on Image Processing **15** (2006) 2226–2235

[17] Chen, M.J., Huang, C.H., Lee, W.L.: A fast edge-oriented algorithm for image interpolation. Image and Vision Computing **23**(9) (2005) 791–798

[18] Giachetti, A., Asuni, N.: Real time artifact-free image upscaling. IEEE Transactions on Image Processing **20**(10) (2011) 2760–2768

[19] Zhou, D., Shen, X., Dong, W.: Image zooming using directional cubic convolution interpolation. IET Image Processing **6**(6) (2012) 627–634

[20] Sheikh, H., Sabir, M., Bovik, A.: A statistical evaluation of recent full reference image quality assessment algorithms. IEEE Transactions on Image Processing **15**(11) (2006) 3440–3451

[21] Yu, S., Zhu, Q., Wu, S., Xie, Y.: Performance evaluation of edge-directed interpolation methods for images. Computer Vision and Pattern Recognition (2013)

Author Biography

Andrey Nasonov received his BS (2007) and PhD (2011) in applied mathematics from Lomonosov Moscow State University. Since then he has worked in the Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University. Now he is a senior researcher at the Laboratory of mathematical methods of image processing.

Andrey Krylov received his BS (1978) and PhD (1983) in applied mathematics from Lomonosov Moscow State University. Since then he has worked in the Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University. Now he is a full professor, head of the Laboratory of mathematical methods of image processing.

Xenya Petrova received her BS (1998) in computer science and PhD (2002) in automatic control in St-Petersburg Institute of Aero-space Instrumentation. Since 2005 she works in Samsung R&D Institute Rus, Moscow, Russia, where she is currently a senior engineer in Image Processing Group.

Michael N. Rychagov received his MS degree in physics from Department of Physics at Lomonosov Moscow State University, Russia in 1986, PhD degree and Dr. Sc. degree from the same university in 1989 and 2000 correspondingly. Since 1989 he has been involved in R&D and teaching on Faculty of Electronic and Computer Engineering at Moscow Institute of Electronic Technology (Technical University). Since 2004, he joined Samsung R&D Institute Rus, Moscow, Russia, where he is currently Director of Algorithm Laboratory. He is member of IS&T and IEEE Societies.