Article

Image Interpolation via Scanning Line Algorithm and Discontinuous B-Spline

Cheng-ming Liu [1], Ze-kun Wang [2], Hai-bo Pang [1] and Jun-xiao Xue [1,*]

[1] School of Software and Applied Technology, Zhengzhou University, Zhengzhou 450002, China; cmliu@zzu.edu.cn (C.L.); phb@zzu.edu.cn (H.P.)
[2] Department of Automation, Tsinghua University, Beijing 100084, China; zk-wa14@mails.tsinghua.edu.cn
[*] Correspondence: xuejx@zzu.edu.cn; Tel.: +86-371-6388-9129

Abstract: Image interpolation is a basic operation in image processing. Lots of methods have been proposed, including convolution-based methods, edge modeling methods, point spread function (PSF)-based methods or learning-based methods. Most of them, however, present a high computational complexity and are not suitable for real time applications. However, fast methods are not able to provide artifacts-free images. In this paper we describe a new image interpolation method by using scanning line algorithm which can generate C^{-1} curves or surfaces. The C^{-1} interpolation can truncate the interpolation curve at big skipping; hence, the image edge can be kept. Numerical experiments illustrate the efficiency of the novel method.

Keywords: image interpolation; C^{-1}; scanning line; B-spline

1. Introduction

Image interpolation, also known as image super resolution, has recently become a hot topic in the field of computer vision and computer graphics due to the increasing number of practical applications of the algorithms proposed. The aim is to improve the resolution of a low resolution (LR) image/video to obtain a high-resolution (HR) one which is able to preserve the characteristics of natural images/videos. The interpolation methods are implemented in a variety of computer tools like printers, digital TV, media players, image processing packages, graphics renderers, and so on. In this paper, we propose a method to generate high-resolution image from a single low-resolution image by a simple C^{-1} interpolation, which runs very fast.

In recent years, many image interpolation or super resolution algorithms have been proposed [1,2], and they can be classified into five categories: based on the point spread function (PSF) method, learning-based method, edge-directed method, geometry-based method or polynomial-based method.

1.1. PSF-Based Interpolation

These methods are constructed based on the PSF, and it is assumed that the LR image is generated by down-sampling from the HR image according to a convolution model as described in Equation (1) where **Y** denotes the LR image and **X** the unknown HR image, **D** is the down-sampling matrix, and **N** is the noise. The image interpolation is regarded as an inverse problem to solve the unknown HR image data [3,4].

$$\mathbf{Y} = \mathbf{DX} + \mathbf{N}. \tag{1}$$

But usually this inverse problem is ill-posed, so we can only find a optimized solution. An extra regular term is usually required to ensure the smoothness of the HR image and it makes the question more complex. Due to the large amount of image data, the inverse problem solution is usually

time-consuming. Qi Shan et al. [5] proposed a fast image/video upsampling method by a feedback control framework based on graphics processing unit (GPU).

1.2. Learning-Based Interpolation

By this method, the HR image patches are recovered from the counterpart of LR patches, and the down-sampling matrix **D** in Equation (1) is not pre-defined, but is learned from a training set [6]. The training set is a database of LR and HR image pairs, usually with a relative scale factor of 2. Then we can recover a most likely HR image according to the training set. Under this framework, Markov random field [7] and sparse representation [8] are also introduced. Recently, deep learning based image super resolution has become a hot topic in image processing [9].

1.3. Edge-Directed Interpolation

Since edges are visually attractive to the human perceptual system, many algorithms focus on edge-directed interpolation methods [10,11]. The main idea of these methods is to preserve the edge sharpness during the image interpolation process. The intuitive way is to explicitly estimate the edge orientation and then interpolate along the edge orientation. Some adaptive interpolation methods focus on preserving sharp edges of images, these adaptive and edge-oriented algorithms [12] are also classified into this category.

1.4. Geometry-Based Method

In recent years, geometry-based image processing methods [13–15] become more and more popular, some geometric tools or computer graphics tools, such as gradient meshes [16], Delauney triangulations [17] and subdivision are introduced to image processing. In our previous work, we proposed a subdivision based image interpolation method [18]. The subdivision method possessed a tangent vector, by adjusting the tangent vector, we can control the sharpness at image edges. This method is also based on a continuous model, and its tangent vector may be affected by image noises.

1.5. Polynomial-Based Interpolation

Polynomial-based interpolation methods are simple and most often used. The nearest neighbor interpolation, bi-linear interpolation, bi-cubic interpolation and cubic B-spline interpolation belong to this category. They are usually implemented by convolving an image with a convolution kernel as the weighting function [19]. These methods run very fast, but they usually cause blurring on image edges because the kernels are continuous functions. Blending kernels [20] and two-dimensional cubic convolution kernels [21] have been proposed to improve it.

The PSF-based method, learning-based method, edge-directed method and geometry-based method mentioned above are very efficient in keeping the edge sharping of HR images, but most of them are time-consuming, especially the learning based method. The polynomial-based methods are almost real-time, but they do not handle the image edges specially. In this paper, we propose a novel image interpolation algorithm based on discontinuous B-spline, which we called C^{-1} interpolation. It can generate interpolation curves with discontinuous points at image edges, hence it can preserve the sharpness of images from the theoretical foundation. While in flat area, our interpolation becomes the classical B-spline interpolation. In general, our method runs as the B-spline interpolation in most of the image area except image edges which are processed specially by C^{-1} interpolation.

2. C^{-1} Interpolation

Interpolation is a general tool in geometric design, it aims to generate fairing (piecewise) polynomial curves/surfaces while keeping smoothness as high as possible. While in some practical applications such as computer aided design (CAD), simulating deformations of objects, animation,

etc., only interpolation and smoothness are not enough, and some geometrical or physical constraints are necessary, such as arc length constrained curve design [22,23], subdivision surfaces with boundary curves and cross boundary derivatives [24], modified Catmull-Clark or Loop subdivision, which can interpolate given normals [25], or constrained subdivision surfaces interpolating a given curve [26,27]. These methods may not be smooth near some extraordinary points, but they are all continuous everywhere. In image interpolation, the interpolation should be interrupted at big data skipping to maintain image edge sharping. In this paper, we describe a new interpolation method with some discontinuous points, which we called C^{-1} interpolation, because its smoothness is less than C^0 continuous.

Discontinuous interpolation may be implemented by combining several piecewise interpolation curves/surfaces, but it is not convenient for real application. We adopt a kind of B-spline interpolation with replicative knots, and the C^{-1} interpolation is a kind of modified B-spline that is discontinuous at some knots. A B-Spline curve is constituted of some piecewise polynomials defined on some knots, and it is formulated as

$$p(t) = \sum_{i=0}^{n} p_i N_{i,k}(t). \tag{2}$$

where $\{p_i\}$ are control vertices, and $N_{i,k}(t)$ is the $i-$th basis function of a k order B-Spline [28] defined on knots vector T which is defined as

$$T = [\overbrace{0, \cdots, 0}, \cdots, \overbrace{b, \cdots, b}, \cdots, \overbrace{1, \cdots, 1}].$$

where b $(0 < b < 1)$ is the inner knot with replicative number k. For a inner knot with replicative number r, the smoothness of the B-Spline curve at this knot is $k - 1 - r$. Hence, the curve will be discontinuous. In this paper, we just modify the quasi-uniform B-spline knot by setting some inner knots with replicative number k. Figure 1 illustrates a C^{-1} interpolation with two discontinuous points, Figure 1a is the traditional B-spline curve, and Figure 1b is discontinuous B-spline interpolation.
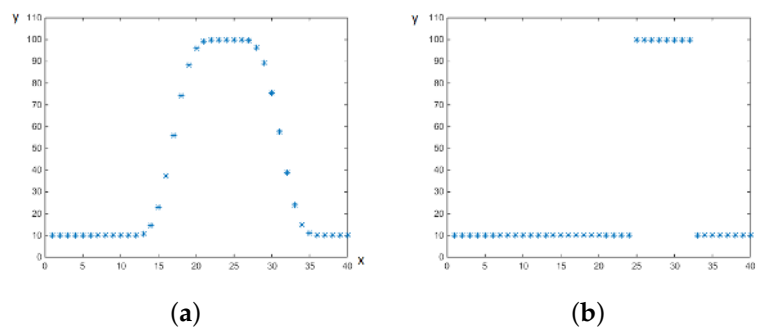
Figure 1. B-Spline interpolation with two discontinuous points. (**a**) is the traditional B-spline; (**b**) is the discontinuous B-spline.

3. Algorithm and Experiment Results

The key point of image interpolation method base on C^{-1} is to construct knot vectors with replicative inner knots according to the image big skip. Hence, the first step is to determine the image edge points, we adopt the scanning line algorithm here to detect image edges.

For a scanning line $y = i$, which denotes the $i-$th line of an image, we set a threshold $thres = (max(I_i) - min(I_i))/4$. If $\|\Delta I_i(j)\| > thres$, then a node with j is added to a linked list, where $\Delta I_i(j) = I_i(j+1) - I_i(j)$.

According to the analysis above, our algorithm is described as follows.

For each line of the LR image,

Step1. For $i-$th line of an image, determine the edge point with the scanning line algorithm. The big skip points are stored in a linked list.

Step2. Construct a knot vector T according to the linked list. For node j, let $t_j = t_{j+1} = \cdots = t_{j+k-1}$. Other knots are determined uniformly.

Step3. Interpolate the $i-$th line with a C^{-1} B-Spline determined on knot vector T, where k is set to 3.

For each column of the LR image, replicate step1–step3, and we will get the HR image.

In the next, we will illustrate the efficiency of our new image interpolation algorithm by comparing some experiments. For convenience, we assume that the scale factor stays the same both horizontally and vertically. We will compare our method with the bilinear interpolation, the new edge directed interpolation (NEDI) algorithm [10] , the fast image upsampling [5]. The bilinear algorithm and the C^{-1} interpolation is implemented on Matlab2014 (MathWorks, Natick, MS, United States), programs of the NEDI algorithm and fast upsampling algorithm are obtained from the homepage of the authors [29,30]. For a color image, it is usually interpolated through three channels separately and is finally combined to a color image. In this paper, we convert the RGB model to YUV model, and then interpolate only on the Y-channel. We test these methods on a variety of images. All the experiments were done on a Thinkpad laptop (Lenovo, Beijing, China) with Core i3 CPU @1.70 GHz and 4 GB main memory. 2×2 times enlargement of a color image with resolution 100×100 typically takes 1 s by our method, while the NEDI and upsampling method take over 8 seconds CPU time.

In the following, we show the ability of our C^{-1} interpolation method numerically by calculating the peak signal-to-noise ratio (PSNR) and the mean structural similarity (MSSIM) [31]. In our experiments, the test data are obtained as follows. For an original image, we down-sample it to $\frac{1}{2} \times \frac{1}{2}$ of its original size by bilinear interpolation, and then up-sample it by the mentioned image interpolation algorithms and finally get the HR images with the same size as the original ones. Here we choose the Kodak lossless true color image suite [32] with resolution 512×768 as the test images and the scale factor is 2×2. The PSNR values and the MSSIM values of the mentioned methods on the test images are shown in Tables 1 and 2, respectively. The test images are the baboon (Figure 2) , kodim05 (Figure 3a) and kodim23 (Figure 3b). Here, we can see that the performance of our methods are better than the others both in PSNR and MSSIM.

| (a) | (b) | (c) | (d) | (e) |
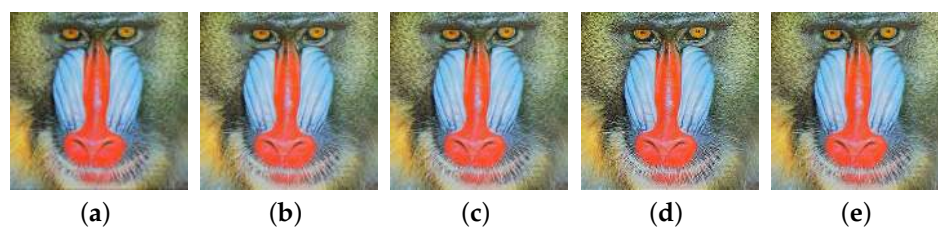
Figure 2. Results of image interpolation. From left to right are: (**a**) original image; (**b**) bi-linear interpolation; (**c**) new edge directed interpolation (NEDI) interpolation; (**d**) fast upsampling interpolation; (**e**) C^{-1} interpolation.

| (a) | (b) |

Figure 3. 4×4 enlargement results of our image interpolation. (**a**) 4×4 enlargement result of Kodim05; (**b**) 4×4 enlargement result of Kodim23.

Table 1. Peak signal-to-noise ratio (PSNR) values.

	Bilinear	**NEDI**	**Upsampling**	C^{-1} **Interpolation**
baboon	24.684	24.072	24.576	25.168
kodim05	23.817	23.545	23.878	24.269
kodim23	31.334	30.828	31.039	32.234

Table 2. Mean structural similarity (MSSIM) values.

	Bilinear	**NEDI**	**Upsampling**	C^{-1} **Interpolation**
baboon	0.7245	0.7041	0.7448	0.7670
kodim05	0.7370	0.7281	0.7534	0.7689
kodim23	0.9249	0.9207	0.9185	0.9335

4. Conclusions

In this paper, we propose a simple but efficient image interpolation algorithm. The algorithm can automatically generate discontinuous interpolation at image edges, and it can maintain sharpness at the image edges from the theoretical basis. Some numerical experiments illustrate the efficiency of the novel algorithm. Compared with other methods, our algorithm is simple. Hence, it is suitable for hardware implementation and runs very fast.

Author Contributions: Cheng-ming Liu and Jun-xiao Xue deduced the C^{-1} B-spline interpolation formulae; Ze-kun Wang implemented the image intepolation algorithm; Cheng-ming Liu and Hai-bo Pang wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lehmann, T.M.; Gönner, C.; Spitzer, K. Survey: Interpolation Methods in Medical Image Processing. *IEEE Trans. Med. Imaging* **1999**, *18*, 1049–1075.
2. Acharya, T.; Tsai, P.S. Computational Foundations of Image Interpolation Algorithms. *ACM Ubiquity* **2007**, *8*, 1–17.
3. Chan, R.H.; Riemenschneider, S.D.; Shen, L.X.; Shen, Z.W. Tight frame: An Efficient Way for High-resolution Image Reconstruction. *Appl. Comput. Harmonic Anal.* **2004**, *17*, 91–115.
4. El-Khamy, S.; Hadhoud, M.; Dessouky, M.; Salam, B.; El-Samie, F.A. Efficient Implementation of Image Interpolation as an Inverse Problem. *Digit. Signal Process.* **2005**, *15*, 137–152.
5. Shan, Q.; Li, Z.R.; Jia, J.Y.; Tang, C.K. Fast Image/Video Upsampling. *ACM Trans. Graph.* **2008**, *27*, 1–7.
6. Chang, H.; Yeung, D.Y.; Xiong, Y. Super-Resolution through Neighbor Embedding. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2004; Volume 1, pp. 275–282.
7. Freeman, W.T.; Pasztor, E.C.; Carmichael, O.T. Learning Low-Level Vision. *Int. J. Comput. Vis.* **2000**, *40*, 25–47.
8. Kim, K.I.; Kwon, Y. Single-Image Super-Resolution Using Sparse Regression and Natural Image Prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1127–1133.
9. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 295–307.
10. Li, X.; Orchard, M.T. New Edge-Directed Interpolation. *IEEE Trans. Image Process.* **2001**, *10*, 1521–1527.
11. Akhtar, P.; Azhar, F. A Single Image Interpolation Scheme for Enhanced Super Resolution in Bio-Medical Imaging. In Proceedings of the 2010 4th International Conference on Bioinformatics and Biomedical Engineering (iCBBE), Chengdu, China, 18–20 June 2010; pp. 1–5.

12. Chen, M.J.; Huang, C.H.; Lee, W.L. A Fast Edge-oriented Algorithm for Image Interpolation. *Image Vis. Comput.* **2005**, *23*, 791–798.

13. Su, D.; Willis, P. Image Interpolation by Pixel-Level Data-Dependent Triangulation. *Comput. Graph. Forum* **2004**, *23*, 189–201.

14. Liang, L.M.; Zou, B.J.; Zhao, H.X.; Qiu, X. Image Interpolation via Tangent-Control Subdivision. In Proceedings of the 2008 Congress on Image and Signal Processing, Sanya, China, 27–30 May 2008; Volume 4, pp. 102–106.

15. Pan, Q.; Luo, X.N.; Zhu, J.W. A Medical Image Magnification Based on Local Geometric Similarity. *J. Softw.* **2009**, *20*, 1146–1155.

16. Lai, Y.K.; Hu, S.M.; Martin, R.R. Automatic and Topology-Preserving Gradient Mesh Generation for Image Vectorization. *ACM Trans. Graph. (SIGGRAPH)* **2009**, *28*, 1–8.

17. Demaret, L.; Dyn, N.; Iske, A. Image Compression by Linear Splines Over Adaptive Triangulations. *Signal Process.* **2006**, *86*, 1604–1616.

18. Liu, C.M.; Luo, X.N. Image enlargement via interpolatory subdivision. *IET Image Process.* **2011**, *5*, 567–571.

19. Keys, R.G. Cubic Convolution Interpolation for Digital Image Processing. *IEEE Trans. Acoust. Speech Signal Process.* **1981**, *29*, 1153–1160.

20. Liang, L.M. Image Interpolation by Blending Kernels. *IEEE Signal Process. Lett.* **2008**, *15*, 805–808.

21. Shi, J.Z.; Reichenbach, S.E. Image Interpolation by Two-Dimensional Parametric Cubic Convolution. *IEEE Trans. Image Process.* **2006**, *15*, 1857–1870.

22. Van Damme, R.; Wang, R.H. Curve Interpolation with Constrained Length. *Computing* **1995**, *54*, 69–81.

23. Roulier, J.A.; Piper, B. Prescribing the Length of Parametric Curves. *Comput. Aided Geom. Des.* **1996**, *13*, 3–22.

24. Levin, A. Combined subdivision schemes for the design of surfaces satisfying boundary conditions. *Comput. Aided Geom. Des.* **1999**, *16*, 345–354.

25. Biermann, H.; Levin, A.; Zorin, D. Piecewise smooth subdivision surfaces with normal control. In Proceedings of the 27th annual conference on computer graphics and interactive techniques, New Orleans, LA, USA, 23–28 July 2000; pp. 113–120.

26. Nasri, A.H. Recursive subdivision of polygonal complexes and its applications in computer-aided geometric design. *Comput. Aided Geom. Des.* **2000**, *17*, 595–619.

27. Nasri, A.H.; Abbas, A. Designing Catmull-Clark subdivision surfaces with curve interpolation constraints. *Comput. Graph.* **2002**, *26*, 393–400.

28. Piegl, L.; Tiller, W. *The NURBS Book*, 2nd ed.; Springer: New York, NY, USA, 1997.

29. Li, X. New Edge Direction Interpolation. 2001. Available online: http://www.csee.wvu.edu/~xinl/publications.html (accessed on 19 July 2013).

30. Shan, Q.; Li, Z.; Jia, J.; Tang, C.K. Fast Image/Video Upsampling. 2008. Available online: http://www.cse.cuhk.edu.hk/~leojia/projects/upsampling/index.html (accessed on 11 August 2011).

31. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612.

32. Kodak. Kodak Lossless True Color Image Suite. Available online: http://r0k.us/graphics/kodak/ (accessed on 1 June 2011).