2010 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS 2010) December 6-8, 2010

IMAGE INTERPOLATION BASED ON DECOMPOSITION

Nana Guo, Li Song, Xiaokang Yang and Wenjun Zhang

Institute of Image Communication and Information Processing, Shanghai Jiao Tong University
Shanghai Key Lab of Digital Media Processing and Transmission
Shanghai, China, 200240

ABSTRACT

This paper proposes a new approach for interpolating natural images. Unlike other conventional interpolation methods, we exploit the characteristics of an image and decompose it into a texture part and a non-texture (cartoon) part. In the non-texture part, we interpolate it with bicubic interpolation which performs well for smooth areas. We deal the texture part with peak transform based interpolation which resets the image pixels to capture the characters of the texture through peak transform. The experiment results demonstrate our new interpolation method has higher PSNR and better subjective quality of the interpolated image, especially in the texture part.

1. INTRODUCTION

The process of image interpolation is to acquire a high-resolution image from its low-resolution one. So how to describe the relationship between a high-resolution image and a low-resolution image is important. Many interpolation methods have been proposed. Linear interpolation (e.g. bilinear and bicubic) approximates the value of the interpolated point within the local known ones linearly which is suitable for the low-frequency images. The edge-directed based interpolation in [1] or [2] is to estimate local correlations from a low-resolution image and then use these correlations' estimates to adapt the interpolation at a higher resolution one. It has good quality for image edges. Adaptive interpolation in [3] or [4] also exploits local spatial structures around the edges and then adapts the interpolation coefficients to improve performance. Transform domain interpolation in [9] or [10] transforms images into another domain to efficiently capture the features from the low-resolution images and use these extracted information to get high-resolution ones. Most of these previous interpolation algorithms operate on the images with a single approach. To the authors' knowledge, there are few works considering the differences of image features on interpolation.

In this paper we describe a new method to image interpolation. Our motivation comes from the image decomposition [5]. Image decomposition is to decompose an image into a non-texture part and a texture part. The non-texture part corresponds to the low-frequency areas of an image while the texture part contains the image's fine-grained details. We interpolate non-texture part with bicubic since it has good performance for smooth areas. For texture component, we propose a peak transform (PT) based interpolation method which can well adapt to the characteristics of texture signals. The experiment results demonstrate that our method improves both the PSNR and the visual image quality.

The rest of the paper is organized as follows: In Section 2, we will present our proposed interpolation scheme. Experimental results will be reported in section 3 and section 4 will conclude the paper

2. PROPOSED INTERPOLATION SCHEME

The challenge of image interpolation is how to obtain a high-resolution image from its low-resolution one with high PSNR and good vision quality. Among previous methods, there are some with good performance in smooth areas, and some are suitable for edges and so on. But most of them don't have these characteristics with simultaneity. Therefore, it is highly desirable to find an approach which is able to interpolate an image with good performance in many aspects such as smooth areas and the textures. In other word, it could provide good performance both in the non-texture part and the texture part. Our problem in this work becomes: can we find a method which can interpolate the image's texture part and non-texture part respectively? In this section we will introduce our novel interpolation approach.

The key in our research is to interpolate the image's texture part and non-texture part respectively. So we can decompose an image into the texture part and the non-texture part. Now we need to choose the appropriate interpolation for the two decomposed parts respectively.

This work was supported in part by NSFC(60702044, 60632040, 60625103), MIIT of China(2010ZX03004-003) and 973(2010CB731401, 2010CB731406).

978-1-4244-7371-7/10/$26.00 ©2010 IEEE

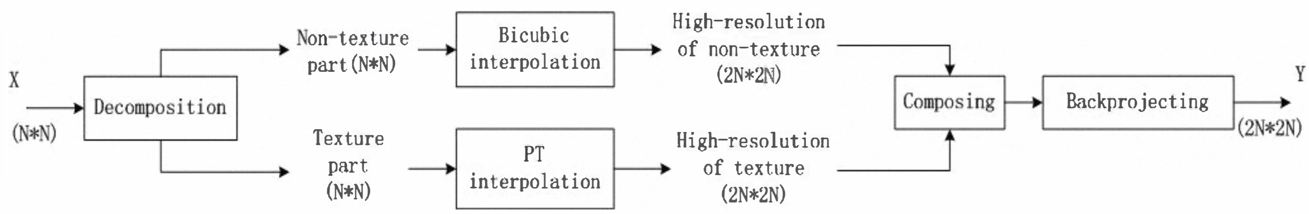

Fig.1. Interpolation based on decomposition

As we know, bicubic interpolation has higher PSNR and better vision quality in image with low-frequency area than others. So we interpolate the non-texture part with bicubic interpolation. In regard of the texture part, we propose a new method based on peak transform which can capture the details of the texture. After combining these two interpolated parts, an iterative algorithm is used to improve the image quality. The whole process of the interpolation is shown in Figure 1. Later we will discuss the image decomposition and the interpolation based on peak transform respectively.

2.1. Image decomposition

Decomposing an image into a texture part and a non-texture (cartoon) part is paid great attentions in many image applications. In general, the non-texture part is the low-frequency objects extracting from an image and the texture part contains the high-frequency details. Many decomposition methods have been proposed. They are different in formulating the texture part. Usually the texture part is defined according to its application. In our interpolation, we choose the decomposition proposed by [5] which aims to fully extract self-similar structure from an image. We will introduce it in details.

A two-dimensional image ($M \times N$) is reordered into an one-dimensional signal in the raster scanning order. So the image decomposition problem becomes to find the optimum solution by solving

$$\begin{aligned} &\min_{x_l x_h x_n} \|\nabla x_l\|_1 + \mu_h \|\mathrm{H} x_h\|_1 + \mu_n \|x_n\|_1 \\ &s.t. x = x_l + x_h + x_n \end{aligned} \tag{1}$$

Where x_l is the non-texture part, x_h is the self-similar structure and x_n is the noisy part. $\|\nabla x_l\|_1$ is the total variation of x_l.

Because it needs too many bases to represent the noisy part, it combines the self-similar structure and the noisy part x_n together as the texture part in order to get sparse representation. Furthermore, during the kernel H learning process, it is unsupervised by principal component analysis (PCA), so it is more suitable to get the clusters with similar sub-texture.

The decomposing process is as follows: assign each pixel a cluster sign to get the pixels clustering by mixture of probabilistic principal component analysis (MPPCA), and sign the map φ by maximizing the conditional probability. Then make a coarse estimation of x_h by $TV - l^1$ decomposition at certain μ . For each φ_i, collect the neighborhood samplers signed by φ_i from x_h , and compute the sub-kernel H_i by PCA. Separate x_l and x_h from x by $TV - l^1$ decomposition at λ_h , and separate x from x_h by basis pursuit using H_i . Set $x = x - x'$ then separate x_l and x_h from x by $TV - l^1$ decomposition at λ_n . Set $x_n = x_h, x_h = x', x_l = x_l$, sign x_l as non-texture and $x_h + x_n$ as the texture part. Segment each sub-texture according to the map φ . So we can obtain two images of its texture part and non-texture part.

2.2 Peak transform-based interpolation

Peak Transform is proposed firstly by He in [6] for image compression. Given two functions $g_1(t), g_2(t)$ defined over intervals $[t_1, t_2]$ and $[t_3, t_4]$ with $t_2 < t_3$, making special connection of these two functions yields a new function $g(t)$.

$$g(t) = \begin{cases} g_1(t) & t \in [t_1, t_2] \\ g_2(t) - g_2(t_3) + g_1(t_2) & t \in [t_2, t_2 + t_4 - t_3] \end{cases} \tag{2}$$

We denote this connecting operation by

$$g(t) = g_1(t) \vee g_2(t) \tag{3}$$

Physically, $g(t)$ is obtained by connecting two functions with proper shifting operations.

So we can get the N-Point peak transform (PT) as follows:

$$h_o(t) = g_1(t) \vee g_3(t) \vee \ldots\ldots \vee g_{2*\lfloor (N-1)/2+1 \rfloor}(t) \tag{4}$$

$$h_e(t) = g_2(t) \vee g_4(t) \vee \ldots\ldots \vee g_{2*\lfloor N/2 \rfloor}(t) \tag{5}$$

$$h(t) = h_o(t) \vee h_e(t) \tag{6}$$

$h(t)$ is the connections of all odd and even numbered functions, respectively. It can be seen that the PT only changes the order of the functions and is reversible as Figure 2 shows. The inverse PT transform can be done by re-connecting the functions according to their original order. Let $T_i = t_i - t_{i-1}$. In inverse PT, the original function $g(t)$ is constructed as follows:

$$g(t) = g_1(t) \vee g_2(t) \vee \ldots\ldots \vee g_N(t) \tag{7}$$

If i is odd, the curve segment is given by

$$g_i(t) = h(t), \sum_{k=0}^{\lfloor i/2 \rfloor - 1} T_{2k+1} \le t \le \sum_{k=0}^{\lfloor i/2 \rfloor} T_{2k+1} \quad (8)$$

If i is even

$$g_i(t) = h(t), \sum_{k=0}^{\lfloor N/2 \rfloor} T_{2k+1} + \sum_{k=1}^{\lfloor i/2 \rfloor - 1} T_{2k} \le t \le \sum_{k=0}^{\lfloor N/2 \rfloor} T_{2k+1} + \sum_{k=1}^{\lfloor i/2 \rfloor} T_{2k} \quad (9)$$

Although the PT and inverse PT are defined for continuous-functions, they can be also applied for images. We can simply treat each row or column of pixels as a function segment with applying PT to reorder the positions of the pixels and get a new image with our intention.

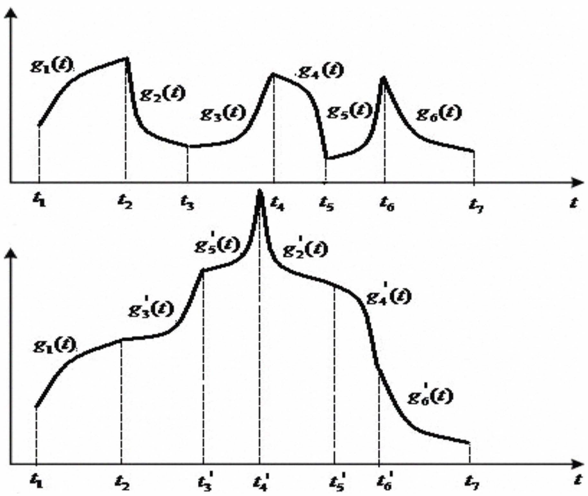

Fig.2. Example of PT: top is the original function, bottom is the function through peak transform. The process is reversible.

As the above describes, the nature of PT is joining two function segments with proper shifting operations. Through the shifting, it moves the segments which have the same trend (ascending or descending) together. So the transformed function is much smoother than the original one. With the discrete-time signals, through the PT, we want to reset the point data at the high-frequency pixels and make the high-frequency pixels as neighborhoods. When interpolating with the neighborhood pixels, the interpolated pixel's value is close to the neighborhood pixels (high frequency) so it can preserve the details in the texture part.

To successfully design an interpolation based on PT, the following problems should be addressed: where are the re- joined points and how to find them. So we should first find the original signal breakpoints correctly. In this application, we want to preserve the image's texture, so we need to get the points at the high-frequency location, namely peak. To do this, we can apply a high-pass filter to the original data. If the signal's response absolute value is greater than the threshold σ (generally σ =8) [9], the location Λ is chosen as peak. So we can get a peak map for an image. With the map, we can obtain the signal segments. We re-connect the odd segments and the even segments respectively. After joining the two parts, we implement the peak transform for discrete signals.

To realize the interpolation (assuming the original image's size is $N \times N$), we first apply the peak transform with the peak map to an image for each row, and interpolate the unknown pixel with its two neighbors, namely horizontal interpolation. Then we perform the inverse PT to it, we get the interpolated image with size $N \times 2N$. Now we are ready for each column which is the same operation as the rows, we call it vertical interpolation. Finally we get the interpolated image ($2N \times 2N$). It should be noted that when the inverse PT is performed with the peak map, due to the up-sampling its peak locations T become $T = 2\Lambda$.

2.3 Backprojecting for post-processing

The PT-based interpolation just contains horizontal and vertical operation. It isn't sufficient enough to estimate the high-frequency details especially with the directional information. So we explore an iterative algorithm [8] to increase the image resolution. It works as follows: starting with an initial high-resolution image. Its process is simulated to obtain a set of low-resolution image corresponding to the observed input image. The differences between these two ones are calculated, and used to improve the initial image by back-projecting each value. This process is repeated iteratively to minimize the differences.

3. RESULTS

TABLE 1
PSNR COMPARISON OF IMAGE INTERPOLATION ALGORITHMS

Test Image	Image Interpolation Algorithm		
	Bicubic	**NEDI**	**Proposed**
Barbara	24.6512	22.0474	**25.0857**
Lena	**33.9487**	33.7100	33.8326
Baboon	22.4666	22.5553	**23.0461**
Boat	**29.2491**	29.0808	29.2089
Cameraman	25.4550	25.4435	**25.7708**
Hair	40.5526	39.4527	**40.6336**

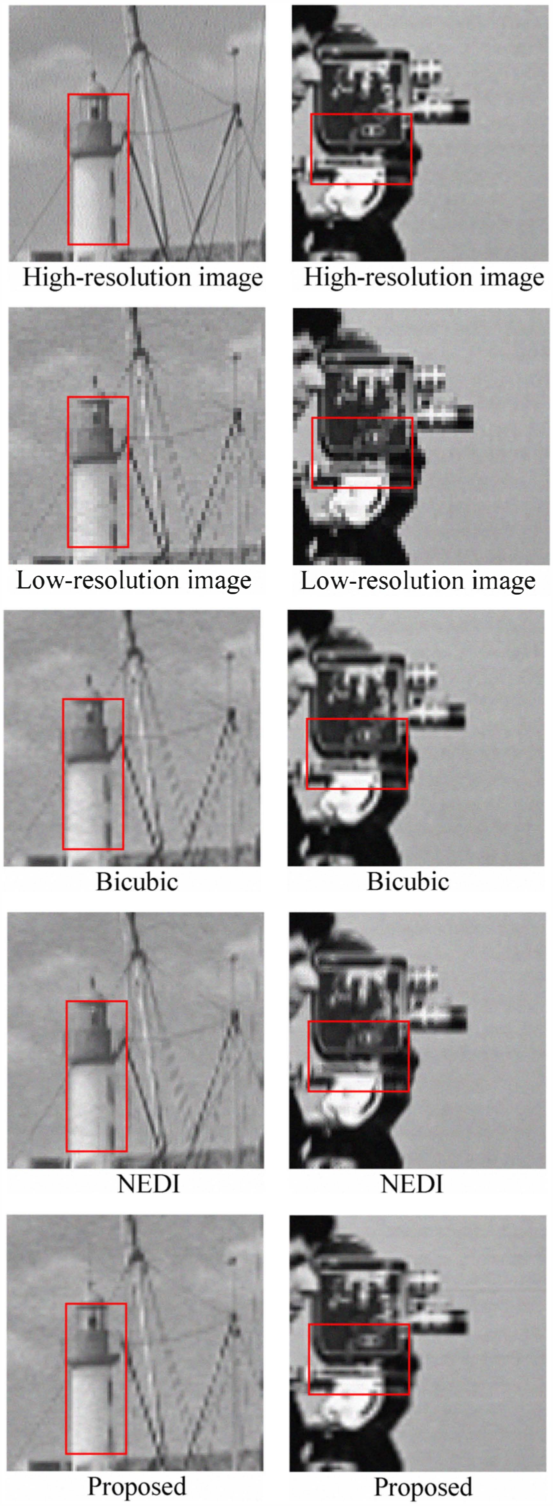

Fig.3. Results of several interpolation methods used in the cropped image from Boat and Cameraman. From the top to bottom, left to right: high-resolution image, low-resolution image (shown at the same scale by enlarging the pixel size), bicubic interpolation, NEDI and the proposed method

In the decomposition process, we set $\mu = 1.5$, $\mu_h = 0.9$, $\mu_n = 1.8$ .The sub-kernel's size is set as 17 and the cluster number is set as 6. The principal component number is set as 16. We make a comparison among the proposed method, bicubic and NEDI. Table 1 gives the PSNRs (db) generated by these algorithms for the test images. Through the result, we can get that the proposed method improves the PSNR of image with many fine-grained details and the visual image quality. There is a very good reconstruction of the texture especially. In Figure 3, we can get that the images interpolated by the proposed method preserve the details(the red mark) in Boat and Cameraman better.

4. CONLUSION

This paper introduces a novel image interpolation scheme. The interpolation is based on the image decomposition into a non-texture part and a texture part. We interpolate the non-texture part with bicubic, and interpolate the other part with PT interpolation. Compose these two parts, and then an iterative algorithm is used to improve its resolution. The new approach demonstrates that it improves both the PSNR and the visual image quality.

5. REFERENCES

[1] X. Li and M.T. Orchard, "New edge-directed interpolation," IEEE Transaction on Image Processing, vol.10. no.10, pp.1521-1527, October 2001.

[2] J. Allebach and P.W. Wong, "Edge-directed interpolation," IEEE 1st Conference Image Processing, vol.3, pp.707-710, 1996.

[3] S.W. Lee and J.K. Paik, "Image interpolation using adaptive fast B-spline filtering," IEEE International Conference Acoustics, Speech and Signal Processing, vol.5, pp.177-180, 1993.

[4] V.R. Algazi, G.E. Ford and R. Potharlanka, "Directional interpolation of image based on visual properties and rank order filtering," IEEE International Conference Acoustics, Speech and Signal Processing, vol.4, pp.3005-3008, 1991.

[5] F. Zhang, X. Ye, and W. Liu, "Image decomposition and texture segmentation via sparse representation," IEEE Signal Processing Letters, vol.15, pp.641-644, 2008.

[6] Z. He, "Peak transform for efficient image representation and coding," IEEE Transactions on Image Processing, 2007.

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

[8] M. Irani and S. Peleg, "Improving resolution by image registration," Graphical Models and Image Processing, vol.53, no.3, pp.231-239, May 1991.

[9] J.I. Agbinga, "Interpolation using the discrete cosine transform," Electronics Letters, vol.28, no.20, pp.1927-1928, September 1992.

[10] V. Velisaljevic, "Image interpolation using oriented wavelet transform," European Signal Processing Conference 2008, August 2008.