Noise-reduced Example-Based Image/Video Enlarging

李昆霖、李根逸、陳炳宇

國立臺灣大學

E-mail: {holyo,kez}@cmlab.csie.ntu.edu.tw、roibn@ntu.edu.tw

Abstract

To enlarge an image, bi-linear interpolation and bi-cubic interpolation are two common and simple methods, but they always generate unacceptable zigzag or blurry results. Some image processing methods are provided to reduce the zigzag or blurry effects by enhancing the edges and textureness or deblurring the images, and some example-based methods used some training high resolution images to provide the missing high frequency part for the enlarged input low resolution image. Although it is possible to record the relationship between the high resolution and low resolution samples of some natural training images and transfer the relationship to enhance the enlarged input low resolution image by using the example-based methods, the noise embedded in the input image is also amplified and makes the enhanced enlarged image looks noisy. Due to the noise, the example-based methods cannot be used to produce a satisfying result when applying to video frames directly. In this paper, we present an improved example-based approach that records the relationship of the middle and high frequency data of some training images, since the low frequency data is not needed for reconstructing the high frequency one. When enlarging an image, we only use the middle frequency data to estimate the missing high frequency one and transfer it to enhance the textureness of the enlarged input image. Since our approach can reduce the noise while enhancing the enlarged image, it can also be used for enlarging the video frames directly.

1 Introduction

Digital camera devices are becoming more and more popular today. To capture photos or videos is much more easily than before. However, to share these digital media with other people is still a difficult problem. For example, most public photo sharing websites on Internet today still only provides limited storage space, such that most upload tools for these websites would automatically decrease the image resolution of the uploaded photos in order to reduce the file sizes. Similarly, even if we can capture high resolution videos with camcorders today, to downscale videos is also a common preprocess before sharing or uploading them. In the meantime, display devices are becoming able to show much higher resolution images and videos. Therefore, the quality gap between the distributed digital media and the display device is becoming much larger.

To fill this gap, low resolution images and videos should be properly enlarged before shown on a higher resolution display. In this case, bi-linear or bi-cubic interpolation is usually chosen to be used. However, to enlarge images or videos by such a common but simple method always produces unacceptable zigzag or blurry results. Therefore, some image processing approaches are introduced to reduce zigzag or blurry effects by enhancing edges and textureness [1], or deblurring [2]. There are also some example-based approaches using some high resolution training images to deduce the missing part of high frequency information by enlarging low resolution input image [3]. In example-based approaches, they estimate the relationship between high resolution and low resolution sample pairs from training images, and then enhance enlarged lower resolution input image based on the learned relationship. Unfortunately, after enhancing the enlarged low resolution input image, the noise embedded in the image is also amplified and makes the image look noisy which is shown in Fig. 2(d).

In most image processing approaches, an image is usually taken or divided as two parts in the frequency domain, i.e., the low frequency part and the high frequency one. Usually the low frequency part contains the "general shapes" of the image, while the high frequency one contains the "details" of it. If we decrease the resolution of a given image, the high frequency information will be lost. Hence, when we enlarge the reduced image, the enlarged image will look blurry since its details are lost. In order to produce a less blurry image, to recover the high frequency information is the main task of the image enlarging, upsampling, or super-resolution work. However, only using the low frequency part to recover the high frequency information may make some low frequency noises embedded in the low frequency part be amplified. Hence, how to recover the high frequency part while filtering out the low frequency noises is the most concerned in the paper.

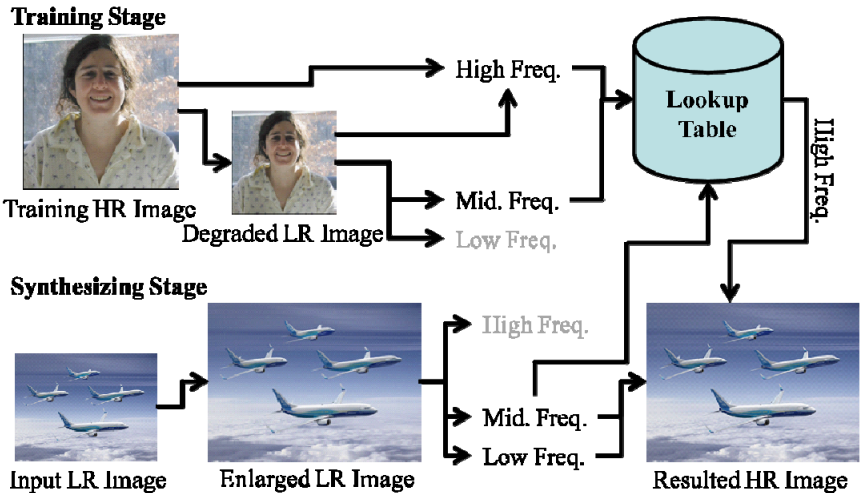


Fig. 1 The framework of our algorithm.

In this paper, we present an improved example-based approach inspired from [3] to enlarge the image resolution based on a key assumption that the low frequency data of an image is locally related to the higher frequency one of the same image. That means, if we know the low frequency band of an image, we can recover its high frequency band by searching a lookup table as shown in Fig. 1. However, to prevent the low frequency noise, we further divide the low frequency band into real low

frequency band and medium frequency one. Hence, the low frequency noise will be confined in the real low frequency band and won't affect the retrieved high frequency part. Furthermore, we can directly apply our method on video frames and obtain a less flickered enlarged video.

The rest of this paper is organized as follows. In Section 3, we describe our noise-reduced example-based approach for enlarging images, and extend this approach for videos in Section 4. Section 5 shows our results and some comparisons with other methods. Finally, we summarize our work in Section 6 and discuss our future work.

2 Related Work

The fastest and easiest way to enlarge an image or video is using the nearest neighbor or standard interpolation (bi-linear, bi-cubic, etc.) methods. These methods are very fast and widely used in many applications today. The downside is, these methods however do not perform well on edges, and will produce blocking artifacts in diagonal lines [4]. Furthermore, these methods cannot recover the details of the image or video since the details are not included in the original low resolution input. Another popular approach is to convolute the input image with a deblurring kernel or an image sharpening one. This approach is also an easy and fast one, but this approach will, in many cases, amplify the noises from the input image or video [2].

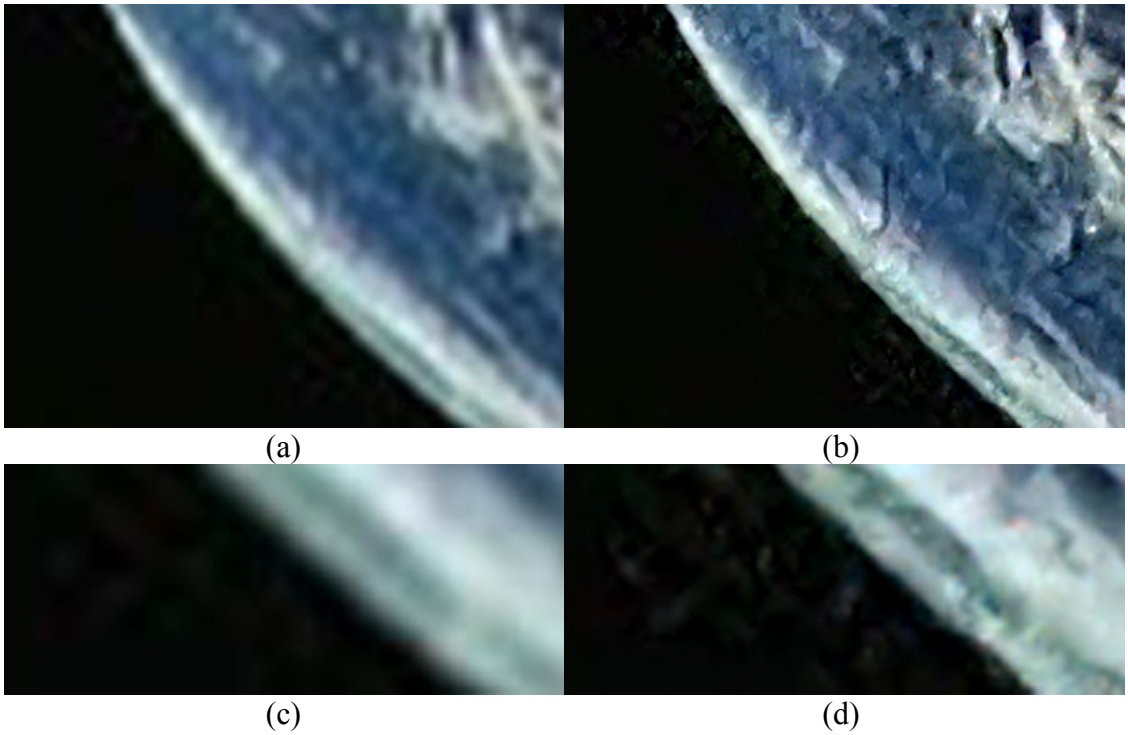


Fig. 2 The enlarged image results of (a) bi-cubic interpolation and (b) Freeman *et al.*'s method [3], where (c) and (d) are the close-up view of (a) and (b). Although Freeman *et al.*'s method produces clearer results; some artifacts are appeared in the black (low frequency) part as shown in (d).

The problem of super-resolution has been studied for more than two decades. One of the most popular methods today is the reconstruction-based algorithm, which relies on sampling theorems [5]. A detailed review of reconstruction-based algorithm can be found in [6]. However, several papers, like [7], have proved that there are many limitations related to the reconstruction-based algorithms. Furthermore, most algorithms proposed so far have problems dealing with dynamic videos, due to the constraints on motion models [8].

Freeman *et al.* proposed an example-based algorithm [3][9]. It assumes that we can guess the details of a high resolution image by analyzing its low resolution version. Based on this assumption, they developed a learning-based method that can enhance the resolution of an image. This method first stores the relationship between the low frequency part and high frequency one of an image. When enhancing the resolution of an image, this relationship can help us to fix the damaged high frequency data of the low resolution image. Hence, a better high resolution result can be achieved. The example-based method produces nice results in many cases and does not need too much extra input. However, this algorithm has serious noise/artifact problems as shown in Fig. 2, and that is what we want to address in this paper.

3 Noise-reduced Example-Based Image Enlarging

The framework of our method is illustrated in Fig. 1. It contains two stages, which are training and synthesizing.

3.1 Training Stage

In the training stage as shown in the upper part of Fig. 1, we first obtain some high resolution images as the training images. For each training data, we downscale the image to simulate the interpolation result of a low resolution input. In this paper, we simply convolute the image with a Gaussian kernel. The downscaled image still preserves the low and medium frequency data of the original training image, but the high frequency part (details) are removed, so the difference between the downscaled image and the original input one is just the lost high frequency data that we want to add to our low resolution input. The downscaled input image is then further divided into low frequency part and medium frequency one by using a band-pass filter. As the result, the original training image is divided into three frequency bands: high, medium, and low frequency parts.

Then, we simply discard the low frequency band, since it is not very informative and may contain some low frequency noise which may not be noticed in the original input image. The medium frequency band and high frequency one of the training images are supposed to have a certain kind of relationship. According to this initial assumption, they are used to form a relationship record, that means the medium frequency data is related to a certain type of high frequency one as the follows:

$$\begin{cases} I_{high} = I - F_D(I) \\ I_{mid} = F_H\left(F_D(I)\right) \\ I_{low} = F_D(I) - F_H\left(F_D(I)\right) \end{cases}, \quad (1)$$

where I_{high}, I_{mid} and I_{low} denote the high, medium, and low frequency parts of the input image $I = I_{high} + I_{mid} + I_{low}$, F_D is our downscale kernel, and F_H is a high pass filter that filters out the lower frequency part.

Since what we want to extract here is the structure of the image, rather than the actual pixel values, we perform a contrast normalization operation, as suggested in [3], on those images to remove the local contrast effects. Then, for each pixel in the training image, except the ones on the edge of it, we generate a corresponding 5x5 patch from the medium frequency image and a 3x3 patch from the high frequency one. These two patches are paired to form the training dictionary. The result of the training stage finally forms a lookup table, which contains the information about the relationship between the medium frequency data and high frequency one. The medium frequency patch acts as the index, which can be matched by using another medium frequency patch, and the high frequency one is the content we need to improve the quality of the enlarged input images.

For a single 400x400 training image, the lookup table includes close to 160,000 patches, each patch has an index of 5x5 = 25 pixels and 3 color channels. To search such a huge lookup table is very time consuming, so in our implementation, we use the ANN library (http://www.cs.umd.edu/~mount/ANN/), to build a kd-tree for the lookup table and thus increase the searching speed. Since the training images are split into several small patches during the training stage, the actual content of the training images is not very important to us. Any image can be used as the training image as long as they are clear and include a rich collection of objects. According to our experiments, it is not necessary to use a training image similar to the low resolution input one in the synthesizing stage, since it does not produce better results. In this paper, the lookup table is trained by using two of the six training images used in [3].

3.2 Synthesizing Stage

After the lookup table is built, as shown in the lower part of Fig. 1, we can take any image as input, enlarge it using conventional interpolation method, and enhance the result using the high frequency data queried from the lookup table. For each input low resolution image we want to enlarge, we first enlarge it to the target resolution by using a simple interpolation method (bi-cubic interpolation in our case), which serves our initial guess I'. Then, we decompose the enlarged image I' into three channels of frequency I'_{high}, I'_{mid}, and I'_{low} by using the same method in the training stage, where $I' = I'_{high} + I'_{mid} + I'_{low}$. Since the low resolution input image includes less details compared to its high resolution version, we can say that the high frequency data I'_{high} embedded in the input image is damaged. Hence, after local contrast normalization, I'_{mid} is used to query the lookup table we built in the training stage to recover the high

frequency part $I'_{high'}$ that corresponds to this medium frequency image I'_{mid}. The retrieved high frequency data $I'_{high'}$ is then used to replace the damaged high frequency data in our initial guess I' to obtain the result image $I'' = I'_{high'} + I'_{mid} + I'_{low}$.

The only problem now is how to find a corresponding high frequency patch in the lookup table for each medium frequency patch in I'_{mid} stripped from the enlarged input image I'. For each medium frequency patch $m^i \in I'_{mid}$ with a spatial index *i*, we want to find a corresponding high frequency patch h^i that optimizes the following two conditions:

- For each retrieved high frequency path $h^i = h_k$ at the *k*-th entry in the lookup table, its corresponding medium frequency patch m_k constructed in the training stage should be similar with the given medium frequency patch m^i. - the **similarity term** (*S*)
- For each retrieved high frequency patch h^i for the spatial index *i* on the given medium frequency image I'_{mid}, the border area of h^i must be coherent with that of its neighboring patches h^j on the same image I'_{mid}. - the **coherence term** (*C*)

Hence, our goal is to find an entry *k* in the lookup table for a given medium frequency patch m^i with a spatial index *i* on a given medium frequency image I'_{mid} that minimizes the following cost function:

$$h_k = \arg\min_k \varepsilon(k, m^i) = S(m_k, m^i) + \alpha \sum_{j \in N_i} C(h_k, h^j), \tag{2}$$

where $S(m_k, m^i)$ is the L_2-norm distance between the medium frequency patch m^i from the input image and the *k*-th medium frequency entry m_k from the lookup table, $C(h_k, h^j)$ is the L_2-norm distance between the boundary area of the *k*-th high frequency data h_k from the lookup table and the already recovered high frequency data h^j from the neighboring patches *j* of *i* on the input image (i.e., N_i denotes the neighboring patches of *i*). Since our result is synthesized in raster scan order, i.e., from top left to bottom right, we only have to compare the current patch with the patches above and left in the coherence term *C* (i.e., $|N_i| = 2$).

In order to speedup the searching process, in our implementation, we first find the 100 closest patches from the lookup table by considering the similarity term *S* only. L_2-norm is the only calculation required with a kd-tree, so the computation cost is not very high. After the 100 closest patches are found, we then put the coherence term *C* into account to perform the minimization of Eq. (2). After the closest high frequency patch is found, in the post processing stage, we undo the local contrast normalization performed earlier during the training stage. Furthermore, since the high frequency patches overlap with each other in the result image, it is possible that one pixel is

"covered" by multiple patches. In this case, the high frequency data added to this pixel is determined by averaging the pixel values in these patches corresponding to this pixel's location. By adding the high frequency information, our initial guess I', which is assumed to be lack of high frequency data, becomes enhanced enlarged image I''.

Fig. 3 The comparison of (a) bi-cubic interpolation and (b) our result. (c) is the source image, and (d) and (e) are the close-up view of (a) and (b).

4 Noise-reduced Example-Based Video Enlarging

When enlarging a video sequence, we tried to directly apply Freeman *et al.*'s method [3] on each frame of the input video. However, we found that the synthesized video has a very serious flicker problem and is completely unsatisfiable to the users. One can easily guess that a possible reason for this flicker problem is the lack of temporal coherency. That means, the same location in two neighboring frames may be covered by two different high frequency patches in the resulted high resolution video even though they are quite similar in the original low resolution input video. Bishop *et al.* [10] tried to address this issue by enforcing the temporal coherence constraints. They modified the cost function used in [3], so the high frequency patch recovered at a particular location of one frame is more favorable in the next frame when finding the high frequency patches for the area or nearby areas. Their approach did reduce some of the flicker problems, but many patches that can cause flicker problems are still left behind.

Fig. 4 The comparison of (a) bi-cubic interpolation, (b) Freeman *et al.*'s method [3], and (c) our result. (d) is the source image, and (e), (f), and (g) are the close-up view of (a), (b), and (c).

On the other side, example-based algorithms are known to produce ringing and other types of artifacts on the image. This is true even for the derived algorithms which are not very similar to the original one [11]. These artifacts are highly unpredictable and differ from image to image. Many of these artifacts are not very obvious on the result images, especially when they are not used to compare with the original input image directly. However, in the case of video enlarging, these noises will cause flicker ones since they differ in every single frame. Since our method produces fewer noises than the previous methods and thus reduces the flicker problems when applying our method to video sequences. Furthermore, to take the temporal coherence into consideration, both of the spatial and temporal neighboring patches $j \in N_i$ in Eq. (2) are taken into account (i.e., $|N_i| = 3$). Moreover, about the coherence term C for the temporal neighboring patches, instead of comparing the boundary area, all of the pixels in the two patches are compared.

5 Experimental Results

In Fig. 3, our result is compared with that of bi-cubic interpolation. The result image is enlarged four times. When highlighting the face of the woman in the picture as shown in Fig. 3 (d) and (e), it is obvious that our result includes more details

compared to bi-cubic interpolation. Fig. 4 is another similar comparison between bi-cubic interpolation, Freeman *et al.*'s method [3], and our result. In this comparison, our result also includes more details compared to bi-cubic interpolation. The close-up view as shown in Fig. 4 (e), (f), and (g) shows that the difference among the three results, and our result looks smoother and includes fewer noises/artifacts than Freeman *et al.*'s method [3]. Fig. 6 shows another comparison result.

Table 1: The calculation result of BlurExtent [12].

Test image	Bi-cubic	Freeman *et al.*[3]	Our method
Woman	0.1848	0.1411	0.1208
Kids	0.4428	0.3617	0.3604
Koala	0.2548	0.2292	0.2183
Boeing 737	0.6921	0.6605	0.6485

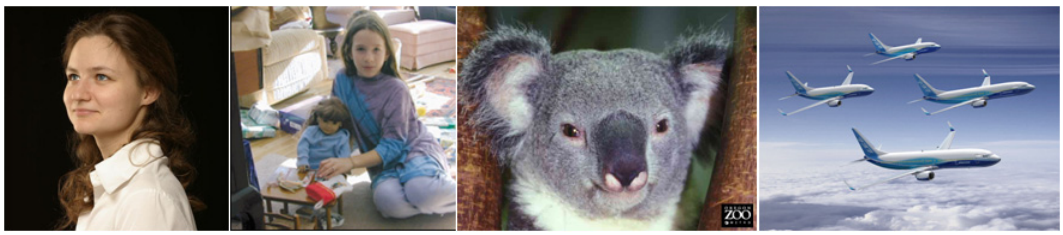


Fig. 5 The photos tested in Table 1.

To show a quantitative test for our method, we performed a blur detection for the results of bi-cubic interpolation, Freeman *et al.*'s method [3], and our method. In this quantitative blurry test, we used four random chosen images as shown in Fig. 5 and calculated the BlurExtent [12] value for the result images and listed them in Table 1. BlurExtent is a method used to measure the blurriness of an image. Since one of the major motivation behind the image/video enlarging algorithm is to produce a less blurry enlarged image/video, we can assume that a good image/video enlarging algorithm must produce a less blurry result. In Table 1, our result is less blurry than both of bi-cubic interpolation and Freeman *et al.*'s method [3].

For the video enlarging result, our result is compared with that of Bishop *et al.*'s method [10]. The test video is enlarged for four times and shown in Fig. 7. In order to measure the quality of the result video, we simply use the difference between the results synthesized by the two methods and the ground truth, respectively. We calculate the error pixel-wisely for each frame in the result video, and average them over the timeline to get the average difference map. Assume that the ground truth does not have flicker problems, the error between the ground truth and result video can be used as the measurement of flickers. Of course it is possible that an area in the result video is constantly different to the corresponding spatial location in the ground truth. In this case, there is no flicker can be observed even through there are errors. However, the error means that the result is not look like the ground truth, and of course that is not what we want in a video enlarging algorithm. The difference maps of Bishop *et al.*'s method [10] and our method are shown in Fig. 8. The darker areas mean that the error is higher, while the white areas mean that there is no error. Hence, our method produces fewer errors than Bishop *et al.*'s method [10]. Fig. 9 is another test video and the experience result is shown in Fig. 10. Our method still produces fewer errors.

Fig. 6 The comparison of (a) ground truth image, (b) bi-cubic interpolation, (c) Freeman *et al.*'s method [3], and (d) our result.

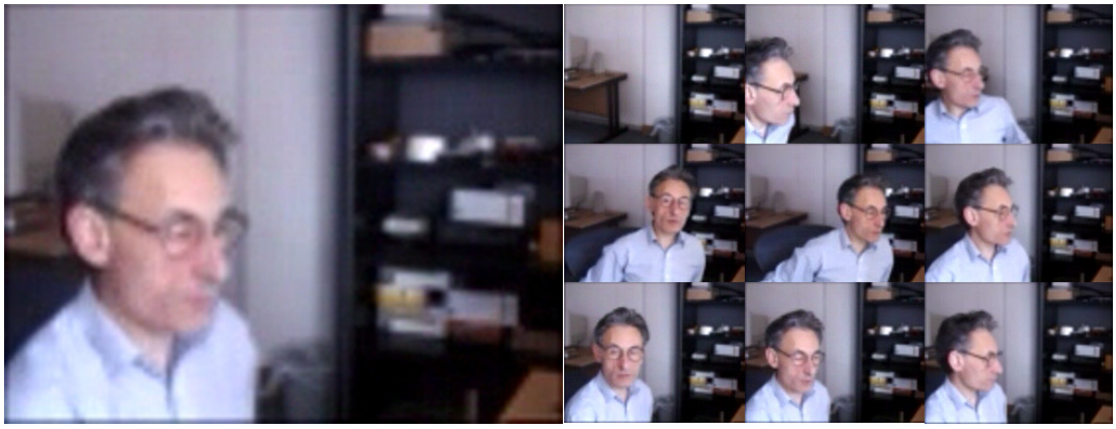
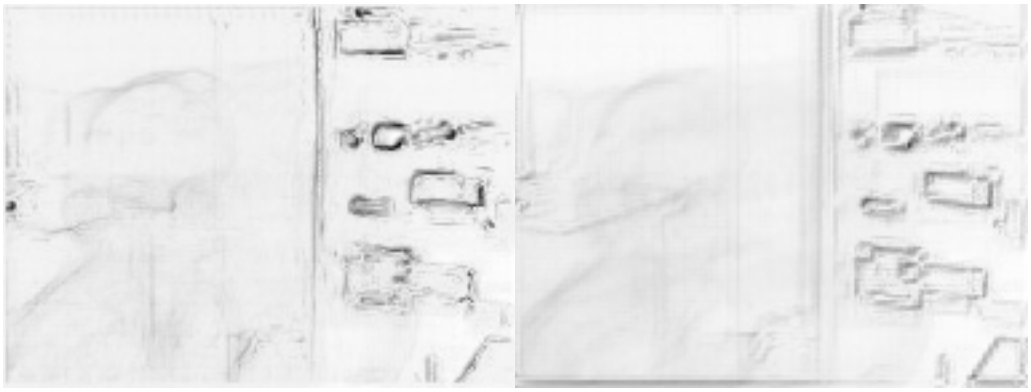


Fig. 7 The test video we used.



(a) (b)

Fig. 8 The average difference maps of Fig. 7 for (a) Bishop *et al*'s method [10] and (b) our algorithm.

Fig. 9 Another test video we used.

(a) (b)

Fig. 10 The average difference maps of Fig. 9 for (a) Bishop *et al*'s method [10] and (b) our algorithm.

6 Conclusion and Future Work

In this paper, we proposed an improved approach to enhance the quality of enlarged images by using an example-based algorithm with less noise and artifact problems, which can also be directly applied to enhance the enlarged videos. By separating the images and videos into three kinds of frequency parts, the artifact and flicker problems are reduced in the result images and videos. However, since the retrieved high frequency data is not the real one of the input images or videos, the result images and videos are only the estimated results. Hence, in the future work, we want to find another way to retrieve the real high frequency data from the photo website.

7 Acknowledgement

This paper was partially supported by CyberLink Corporation, Institute for Information Industry, National Science Council of Taiwan under NSC97-2622-E-002-010, and also by the Excellent Research Projects of the National Taiwan University under NTU95R0062-AE00-02.

References

[1] R. Fattal, “Image upsampling via imposed edge statistics,”, p.95, 2007.
[2] Q. Shan, Z. Li, J. Jia, and C.-K. Tang, “Fast image/video upsampling,” ACM Transactions on Graphics, vol.27, no.5, p.153, 2008.
[3] W. T. Freeman, T. R. Jones, and E. C. Pasztor, “Examplebased super-resolution,” IEEE Computer Graphics and Applications, vol.22, no.2, pp.56–65, 2002.
[4] J. van Ouwerkerk, “Image super-resolution survey,” Image and Vision Computing, vol.24, no.10, pp.1039–1052, 2006.
[5] S. Baker and T. Kanade, “Limits on super-resolution and how to break them,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.24, no.9, pp.1167–1183, 2002.
[6] S. C. Park, M. K. Park, and M.G. Kang, “Super-resolution image reconstruction: a technical overview,” IEEE Signal Processing Magazine, vol.20, no.3, pp.21–36, 2003.
[7] Z. Lin and H.-Y. Shum, “Fundamental limits of reconstruction-based superresolution algorithms under local translation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.26, no.1, pp.83–97, 2004.
[8] Z. Jiang, T.-T. Wong, and H. Bao, “Practical superresolution from dynamic video sequences,” Proceedings of 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.549–554, 2003.
[9] W. T. Freeman, E. C. Pasztor, and O. T. Carmicheal, “Learning Low-Level Vision,” International Journal of Computer Vision, vol.40, no.1, pp.25–47, 2000.
[10] C. M. Bishop, A. Blake, and B. Marthi, “Super-resolution enhancement of video,” Proceedings of 2003 Artificial Intelligence and Statistics, 2003.
[11] T. Q. Pham and L. J. van Vliet, “Resolution enhancement of low quality videos using a high-resolution frame,” Proceedings of SPIE, vol.6077, pp.1–10, 2006.
[12] H. Tong, M. Li, H. Zhang, and C. Zhang, “Blur detection for digital images using wavelet transform,” Proceedings of 2004 IEEE International Conference on Multimedia and Expo, pp.17–20, 2004.