SELF-TRAINING SUPER-RESOLUTION

RockHun Do and InSo Kweon
School of Electrical Engineering
Korea Advanced Institute of Science and Technology
Daejeon, Korea
rhdo@rcv.kaist.ac.kr, iskweon@ee.kaist.ac.kr

ABSTRACT

In this paper, we describe self-training super-resolution. Our approach is based on example based algorithms. Example based algorithms need training images, and selection of those changes the result of the algorithm. Consequently it is important to choose training images. We propose self-training based super-resolution algorithm which use an input image itself as a training image. It seems like other example based super-resolution methods, but we consider training phase as the step to collect primitive information of the input image. And some artifacts along the edge are visible in applying example based algorithms. We reduce those artifacts giving weights in consideration of the edge direction. We demonstrate the performance of our approach is reasonable several synthetic images and real images.
.

Keywords: example-based super-resolution,

1. INTRODUCTION

Resolution of a digital image is limited by optical characteristics and interference of CCD elements. Recently lots of researches called super-resolution, which try to overcome the limitation in software level, have been proposed. Super-resolution technique is mainly categorized into two methods.

First one is reconstruction based super-resolution. This mainly uses multiple low resolution images and combines those all together. Irani et al. [1] defined transformation from low resolution to high resolution as forward projection and from high resolution to low resolution as backprojection and produced high resolution image iteratively with bi-directional projections. Shechtman et al. [2] considered both of spatial and temporal axis and super-resolution is performed simultaneously in time and space. Ben-Ezra et al. [7] developed a jitter camera to move camera in pre-defined amount of subpixel which is necessary for combining low resolution images.

The other is example based super-resolution which is first introduced by Freeman et al. [3]. This uses relationship between low resolution image patch and high resolution image patch and applies it onto low resolution input image patches. Sun et al. [4] extracted patch pairs of low resolution and high resolution on primitive parts such as edges, corner, T-junction, and etc. to reduce error due to mismatches of low and high resolution patches. Kong et al. [5] effectively obtained high resolution videos using high resolution images which are regularly taken during capturing low resolution video and Chang et al. [8] used locally linear embedding on smoothness constraint. Method for training not for intensity of patches but for point spread function on patches was proposed [6].

Although example based super-resolution are arguably considered to generate the best results, there are some limitations. The result of example based super-resolution is highly dependent on training images. If acutance of an input image is totally different from that of training images, result images become unnatural. Fig.1 shows how training images have a great effect on the result image. (e) and (f) are super-resolved from (a) using training image (b) and (c), respectively. Even though (e) is reconstructed from180,000 patches of (b), it has more visible artifacts in the ellipse compared with (f) which is generated from only 5,000 patches of (c). And enormous amount of memory space is required and it consequently takes long time.

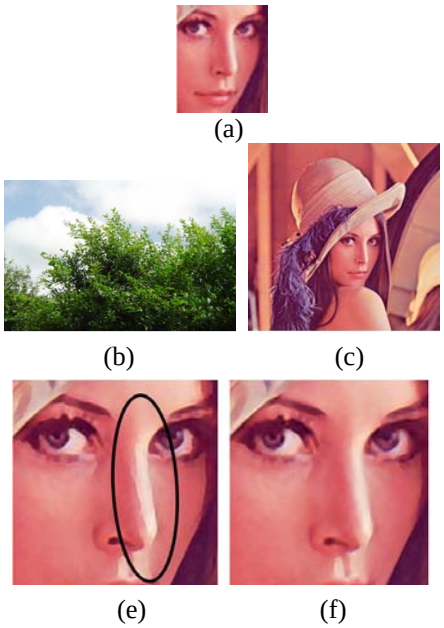


Fig.1: Effect of training images.
(a) is input low resolution image. (e) are super-resolved from 180,000 training patches of (b) and (f) are super-resolved from about 5,000 training patches of (c).

Our approach is motivated by recent progress [5, 9] which show great performance from low resolution videos and images using images which are very similar to the input low resolution image. In [9], one of low resolution stereo image is high-resolved from the other high resolution stereo image. And in [5], high resolution videos are obtained using high resolution images which are regularly taken during capturing low resolution video. In both of two papers images, which have similar primitives with an input low resolution image-video consists of sequence of low resolution images-, are used as training images.

In the rest of this paper, In Section 2, we describe details of self-training based super-resolution and in Section 3 validity in terms of ROC curve is described. Experimental results shown in Section 4 demonstrate that our approach effectively and efficiently reconstruct high resolution image with a small number of training patches. In Section we conclude.

2. DETAILS OF SELF-TRAINING

Fig. 2 shows outline of our approach. Proposed scheme consists of two steps. First input I_L is trained from itself. Input low resolution image I_L is filtered through Gaussian smoothing and down-sampled and up-sampled using simple interpolation.

$$I_{smooth}(x,y)=\frac{\sum_i\sum_j G(i,j)I_H(x+i,y+j)}{\sum_i\sum_j G(i,j)} \qquad (1)$$

$$G(i,j)=\frac{1}{2\pi\sigma^2}e^{-(i^2+j^2)/(2\sigma^2)}$$

where I_{smooth} is a filtered image, G is Gaussian function, and (x,y) is relative distance to (i,j) pixel. Then we obtain the image of same size of an input image but degraded in acutance. Super-resolved image I_H^{p*} is the solution for MAP (Maximum A Posterior) problem [4].

$$I_H^{p*}=\arg\max\, p\left(I_H^p \mid I_H^l\right) \qquad (2)$$

$$I_H^l=\left(I_{smooth}\downarrow s\right)\uparrow s \qquad (3)$$

where $\uparrow s, \downarrow s$ denotes up-sampling and down-sampling by the factor of s, respectively.

$$\begin{aligned} I_H^{p*} &= \arg\max\, p\left(C \mid I_H^l\right) \\ &\approx \arg\max \prod_k p\left(C_k \mid I_H^l\right) \\ &= \prod_k \arg\max\, p\left(C_k \mid I_H^l\right) \end{aligned} \qquad (4)$$

Since our region of interest (ROI) lies along the primitives, if we know about information on edge of an input image I_L^l, the edge can replace our goal image I_H^{p*}. (2) can be represented by each edge line C_k –continuously-linked edge- on the edge C.

2.1 Training Phase

Overview of training phase is shown in Fig. 2.In the training phase, edge prior on an input image is extracted.

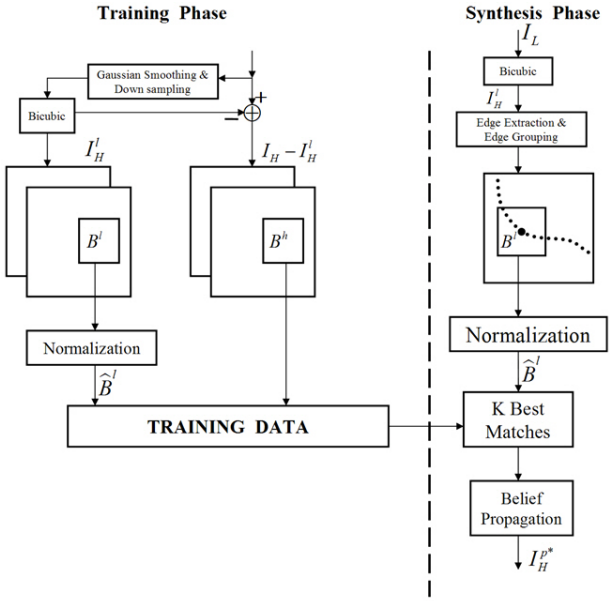


Fig. 2: Outline of self-training super-resolution

As mentioned previously, a low resolution input image passes Gaussian filter to remove aliasing artifact resulting from down-sampling without smoothing. Note that degree of smoothing is practically very important. Finding best matches fails unless variance of Gaussian is appropriate. After down-sampling and up-sampling a degraded image has same size with the input image but it lost acutance. We extract primitive (patch) pairs on both of the degraded image and difference between input and degraded images, containing high frequency details, on which edge of the input image extracted. Since neighboring patches have overlapping region to give smoothness constraint on neighboring patches, extraction intervals of patch pairs have to be less than patch size. In Fig.2 B^l and B^h denotes primitive on the input image and difference image respectively. Since mean and variance of B^l are extremely various, by doing normalization we remove mean and variance.

$$\begin{aligned} \hat{B}^l &= \frac{B^l-\mu^l}{c^l} \\ \mu^l &= B^l - E\left[B^l\right] \\ c^l &= E\left[\,\left|B^l - E\left[B^l\right]\right|\,\right] \end{aligned} \qquad (5)$$

2.2 Synthesis Phase

Basic structure for synthesis phase is shown in Fig. 2. I_L is up-sampled by magnification of s in (3) (I_H^l) . After edge extraction for I_H^l , we make nodes on the each edge. And find K best matching low resolution patches using SSD (Sum of Squared Difference), in which optimal solution set is computed, in training data set. Since $p\left(I_H^l\right)$ is constant, $p\left(C_k \mid I_H^l\right)$ can be represented as

$$\begin{aligned} p\left(C_k \mid I_H^l\right) &\propto p(C_k, I_H^l) \\ &= \sum_{n\in V} E_{one}(B_n^l) + \sum_{(n,m)\in \mathsf{E}} E_{two}(B_n^h, B_m^h) \end{aligned} \qquad (6)$$

In graph model represented as $G=(V,E)$, V is vertex and E is edge. n is index for node which is conceptual consideration of a patch, E_{one} denotes evidence function on a node, E_{two} denotes compatibility function between two adjacent nodes. This energy function can be optimized using dynamic programming [11], belief propagation (BP) [10], and etc. In next sub-section concept of BP are briefly introduced.

2.3 Belief Propagation

Key idea for Belief Propagation (BP) is to update messages iteratively. After selecting K best matches, we should compute the sequence of optimal patch indices. $E_{one}(B_n^l)$ in (6) is almost constant, because in selecting K best matches we already found the patches which has smallest SSD.

$E_{two}(B_n^h, B_m^h)$ can be computed in the overlapping region between adjacent patches as shown in Fig. 3. Energy is computed only in the dark rectangle. Then we can make random vector $(x_1,\ldots,x_M)$ whose element has K possibility. Here M is the number of nodes on one edge line.

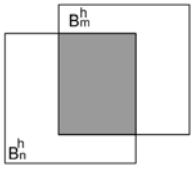


Fig. 3: Energy in compatibility function

For example, one edge line without any junction is considered. Messages are passing from the end of the edge to the other end of the edge as shown in Fig. 4. M_{nm} which denotes the message sent from node m to node n is defined as

$$M_{nm} = \min_{B_n^m} \left\{ E_{two}\left(B_n^h, B_m^h\right) + M_{pre} \right\}. \quad (7)$$

M_{pre} is the summation of all messages coming into m node. M_{nm} is a vector with K elements over all possible indices. M_{nm} means how proper node n is in term of node m. If node m believes node n has to be assigned to index i, i_{th} element of M_{nm} has minimum energy. Table 1 shows pseudo-code for belief propagation. M is the number of nodes.

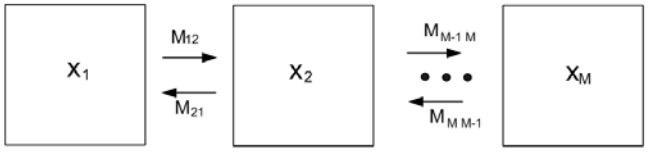


Fig. 4: Flow of message passing

3. VALIDITY OF SELF-TRAINING

There is something doubtful about that an input image can be used as a training image. In this section we show how valid our approach is for example based super-resolution. We present two reasons: one is straightness on the curve

Table1: Pseudo code for BP

$M_{01} \leftarrow \mathbf{0}$

$M_{M+1\,M} \leftarrow \mathbf{0}$

for n=1 : M

$$M_{n\,n+1} \leftarrow \min_{B_n^h} \left\{ E_{two}\left(B_n^h, B_{n+1}^h\right) + M_{n-1\,n} \right\}$$

$$M_{n\,n-1} \leftarrow \min_{B_n^h} \left\{ E_{two}\left(B_n^h, B_{n-1}^h\right) + M_{n+1\,n} \right\}$$

end

edges and the other is that our approach shows better performance in ROC curve tests.

3.1 Straightness

It is clear that an image such as Fig. 5(b) which has only straight components can be super-resolved successfully. The image shown in Fig. 5(a), however, seems hard to be super-resolved. Although the image has curvature on the primitives, it can be considered as straight line because curvature is relatively small compared with the patch size. Image primitives in the red circle in Fig. 5(a) can be super-resolved to the result in the blue circle.

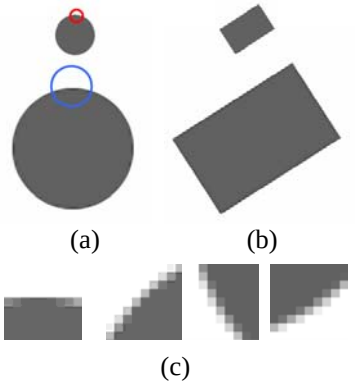


Fig. 5: Straightness

3.2 ROC curve test

The ROC (Receiver Operating Characteristic) [12] represents the proportion (y-axis) of patches whose error are smaller than *e'* (x-axis)to the given error *e'*. Error e' is defined as

$$e(\mathbf{x}) = \frac{\|\mathbf{x} - \mathbf{x}'\|}{\|\mathbf{x}\|}. \quad (8)$$

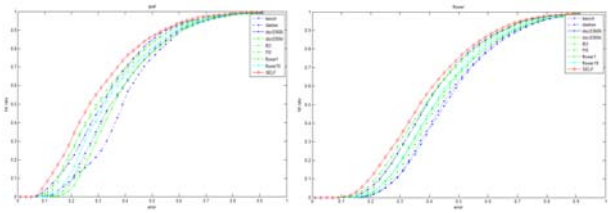
where **x** is patches to be tested and **x'** is ground truth. For the ROC curve test, we down-sampled a high resolution image and applied our approach to the low resolution image. The result is compared with original high resolution image. The ROC curve indicates the proportion of success, area under the graph line shows the performance. Fig. 8 shows ROC curves. We extract patches from input images (a) and (b) shown in Fig. 6 respectively (self-training). In general training, images shown in Fig. 7 were used. For more accurate evaluation after self-training algorithm, same number of patches with self-training was extracted in

general training. In Fig. 6(a) and (b) we extracted about 6000 patches. ROC curves of self-training for Fig. 6(a) which have lots of straight edges shows outstanding performance. In Fig. 6(b) which has more complex details, self-training shows smallest error rate as well. Self training is even better ROC than other images containing similar objects.

(a) (b)

Fig. 6: Input image in ROC test

Fig. 7: Training images



(a) (b)

Fig. 8: ROC curves

4. EXPERIMENTAL RESULTS

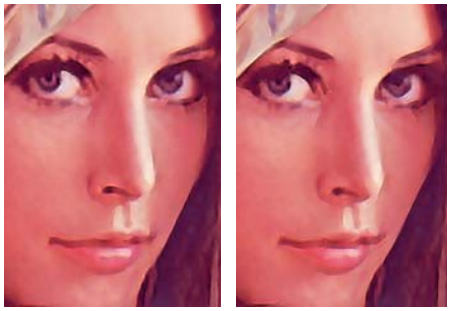
We did all experiments with Core2Duo E6700 2.0GHz cpu and 2GB DDR2 ram. We used 9X9 patch size and variance of Gaussian of 1.2.

4.1 Comparison with General Training

Images shown in Fig. 9 are used in qualitative and quantitative comparison. In general training 680,000 patches were extracted from Fig. 7. We do not include all details here because of space limit. Visual results for some parts of only Fig. 9(a),(e) are shown here. Although self-training has only about 5,000 training patches it reconstructs well compared with general training. Self-training can consequently reduces memory space and processing time remarkably. Processing time for Fig. 9 is listed in Table 2. Self-training takes about 1/7 processing time of general training.

(a) (b) (c) (d) (e)

Fig. 9: Images used in comparison with general training



(a) General training (b) self-training

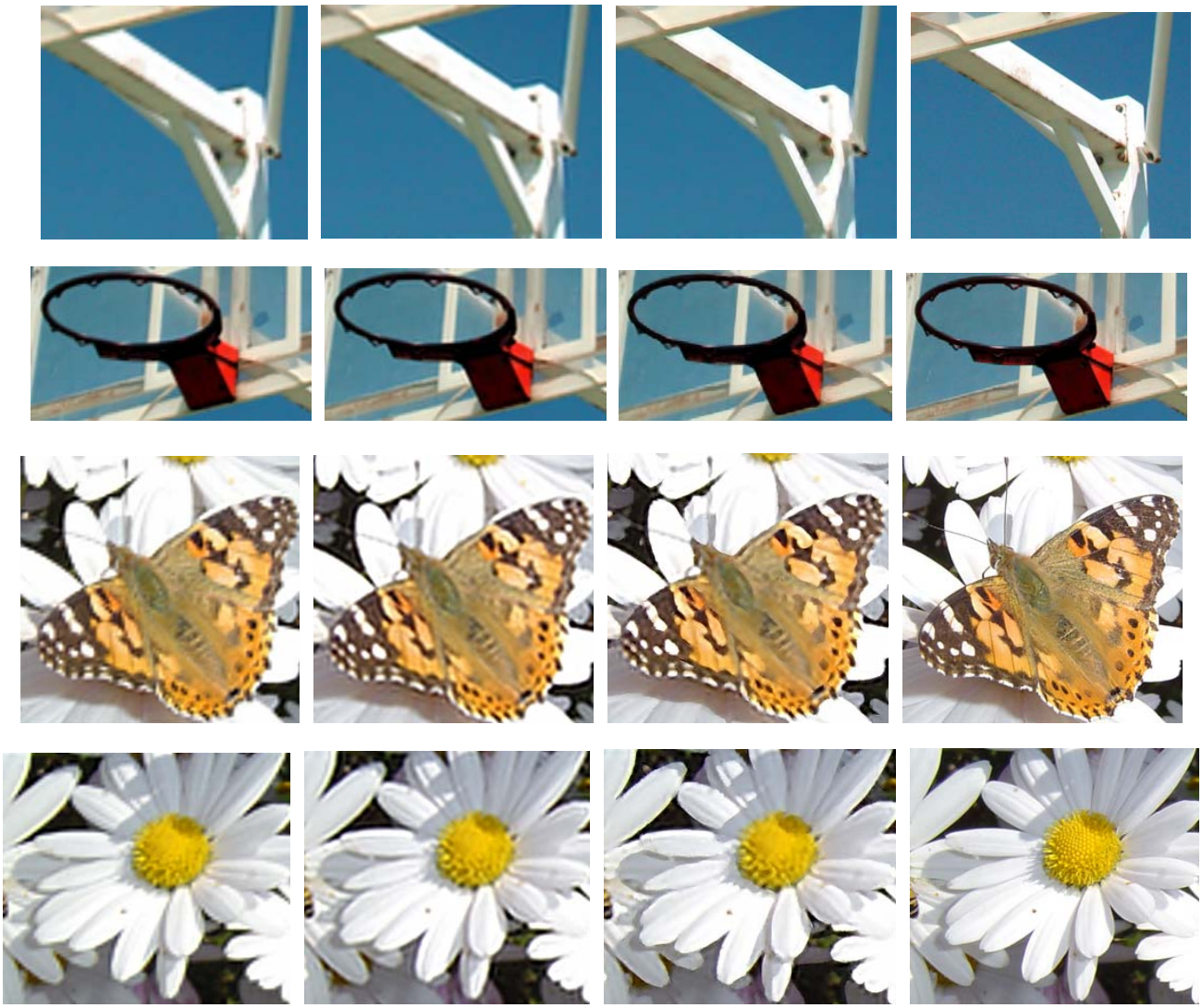
Fig. 10: Comparison with general training

Table 2: Processing time

	General training	Self-training
(a)	64.45	5.20
(b)	14.16	2.86
(c)	97.65	8.77
(d)	15.02	2.51
(e)	22.02	3.28

4.1 Comparison with Other Approaches

Fig. 11 shows comparison with bicubic+unsharp masking, IBP (Iterative Back Projection) [1] and original high resolution images for Fig. 9(c),(d). Self training preserve acutance of input images. In IBP ringing artifacts along the edge are visible. Bicubic with unsharp masking has jaggy artifacts. Since our approach uses single image, it can not reconstruct highly textured regions multi-frame super-resolution can do. It is inherent limitation of single frame super-resolution.



(a) Bicubic + unsharp (b) IBP (c) Self-training (d) Original image

Fig. 11: Comparison with other approaches

5. CONCLUSION

In this paper we described self-training super-resolution. Example based super resolution has inherent problem such as memory space and processing time. We dealt with these problems by considering an input image itself as a library for primitives. We showed validity of self-training super-resolution through ROC curves and consideration of patch size. Experimental results on measurement of processing time showed our approach is much faster than general training super-resolution with preserving qualities.

6. REFERENCES

[1] M. Irani and S. Peleg. Improving Resolution by Image Registration. *CVGIP*, 53(3):231-239, 1991

[2] Eli Shechtman, Yaron Caspi, and Michal Irani. Space-Time Super-Resolution. *TPAMI, 2005*

[3] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *IJCV,* vol. 40. pp.25-47,2000,

[4] J. Sun, N. Zheng, H. Tao, H. Shum. Image Hallucination with Primal Sketch Priors, *CVPR*, 2003

[5] D. Kong, M. Han, W. Xu, H. Tao, Y. Gong. A Conditional Random Field Model for Video Super-resolution, *ICPR*, 2006

[6] Q. Wang, X. Tang, H. Shum. Patch Based Blind Image Super-Resolution. *ICCV* ,2005

[7] M. Ben-Ezra, A. Zomet, S.K. Nayar. Video Super-Resolution using Controlled Subpixel Detector Shifts, *TPAMI*, vol. 27. pp.977-987, 2005

[8] H. Chang, D. Yeung, Y. Xiong. Super-Resolution through Neighbor Embedding, *CVPR*, 2004

[9] H. Sawhney, Y. Guo, K. Hanna, R. Kumar, S. Adkins, S. Zhou. Hybrid stereo camera: An ibr approach for synthesis of very high resolution stereoscopic image sequences. In *Proceedings of ACM SIGGRAPH 2001*, 451.460.

[10] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, San Mateo, California, 1988

[11] R. E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ.

[12] C. E. Metz. *Basic principles of ROC analysis*. Seminars in Nuclear Medicine, pp.283-298, 1978