FAST VIDEO SUPER-RESOLUTION VIA CLASSIFICATION

K. Simonyan, S. Grishin,
{simonyan, sgrishin}@graphics.cs.msu.ru

Moscow State University, Graphics & Media Lab

D. Vatolin, *Member, IEEE*, and D. Popov
{dmitriy, dpopov}@yuvsoft.com

YUVsoft Corp.

ABSTRACT

In this paper we propose a novel super-resolution algorithm based on motion compensation and edge-directed spatial interpolation succeeded by fusion via pixel classification. Two high-resolution images are constructed, the first by means of motion compensation and the second by means of edge-directed interpolation. The AdaBoost classifier is then used to fuse these images into an high-resolution frame. Experimental results show that the proposed method surpasses well-known resolution enhancement methods while maintaining moderate computational complexity.

Index Terms **—** super-resolution, dynamic super-resolution, video resolution enhancement, pixel classification, edge-directed interpolation, AdaBoost.

1. INTRODUCTION

The problem of video resolution enhancement is currently of great importance due to the emergence of high definition (HD) displays. Since most video content is still lower resolution, special algorithms are needed to convert video to higher resolution. Algorithms for video resolution enhancement can be divided into two groups according to the type of information used.

The first group is composed of algorithms that use the information only from the current video frame. Methods from this group (e.g. bilinear and bicubic interpolation) are widely used due to their low computational complexity. The second group consists of the so-called super-resolution (SR) algorithms. In this case the information from neighboring frames is also used, which leads to higher enlargement quality at the cost of higher computational complexity.

Most contemporary SR algorithms are aimed at producing one high-resolution (HR) frame from a set of low-resolution (LR) frames. According to [1], we call them static SR algorithms. Various approaches have been taken recently in this field. Farsiu *et al.* [2] examined the LR image acquisition process in whole. They treated SR as an energy minimization problem and used robust regularization based on bilateral total variation to construct an HR frame. This method assumes affine motion in the sequence and considers the point spread function (PSF) of the camera to be known. Jiang *et al.* [3] improved the classical iterative backward projection algorithm [4] by making it more robust to optical flow estimation errors and by modeling the PSF via an elliptical weighted area filter. Freeman *et al.* [5] proposed an example-based image magnification method. A large database storing the relationship between medium- and high-frequency patches was used to add plausible high-frequency information to the enlarged image. This technique combined with the reconstruction constraint was used by Wang *et al.* [6] for SR. The method, however, exhibits enormous computational complexity. Li [7] investigated the special case of SR given a set of LR frames acquired at different focal lengths. Static SR methods, if directly applied to each video sequence frame, can not provide the temporal consistency of the enlarged video, which impedes their application for video resolution enhancement.

The problem of producing HR video from LR video (dynamic SR) has also been addressed. Bishop *et al.* [8] extended the approach of [5] to video resolution enhancement; special priors were introduced to maintain the temporal consistency of the enlarged video. As any example-based method, it is quite dependent on the training set. Cheung *et al.* [9] applied epitomic analysis to video resolution enhancement. This technique, however, is suitable only for processing video containing scenes obtained at different focal lengths. Farsiu *et al.* [1] used the Kalman filter approximation to bind the frame being enlarged with the previously processed frame. A translational motion model was assumed.

Modern SR algorithms, therefore, exhibit two main shortcomings: huge computational complexity and the necessity of a priori knowledge about camera characteristics and motion models. These shortcomings restrict the algorithms' field of practical application mainly to cases where a sequence of LR images, acquired from spatially close points or with varying focal length, is to be processed. These algorithms are unsuitable for real-time or near-real-time resolution enhancement of video streams.

In this paper we propose a SR algorithm based on HR images fusion via pixel classification. The low computational complexity of the proposed method makes it suitable for fast video processing. Performance superiority over standard video upscaling methods is shown. The rest of the paper is organized as follows. In Section 2 our SR method is described. Experimental results are presented in

Section 3. Section 4 concludes the paper and points some directions of future research.

2. PROPOSED SUPER-RESOLUTION ALGORITHM

2.1. Algorithm Outline

We consider the case of two-times magnification of spatial resolution; the other cases can be easily reduced to this one. It is also assumed that a pixel of LR frame is the mean of four corresponding pixels of this frame in high resolution.

Our approach can be considered as an extension of that described in [1]. An unknown HR frame H^n (the upper index represents the frame number) is modeled by an output frame $\widetilde{H}^n$, which is constructed in three steps:

1. **Motion compensation**. The current LR frame L^n is compensated from already constructed previous HR frame $\widetilde{H}^{n-1}$ to form an HR image M (Section 2.2).
2. **Spatial interpolation**. L^n is upscaled using Edge-Directed Interpolation (EDI, Section 2.3) to form an HR image U.
3. **Fusion**. The resultant frame $\widetilde{H}^n$ is constructed via pixel-wise fusion of two HR images, M and U. Pixel fusion coefficients are calculated via the AdaBoost [10] classifier (Section 2.4).

The flowchart of the algorithm is presented in Fig. 1.

2.2. Motion Compensation

Block-based motion estimation (ME) with adaptive block size (16×16 and 8×8) is used, as it's fast and capable of capturing complex motion in a video sequence. Motion vectors are found with quarter-pixel accuracy. The ME algorithm takes the variance of blocks luminance into account, producing smooth motion vector fields in uniform areas. ME is performed in luminance plane only.

The architecture of SR ME differs from that of conventional ME, as the resolution of the current frame is two times smaller in each direction than the resolution of the reference frame. The same is true for the current block and its corresponding reference block. But under the assumption of dependency between LR pixel (pixel of LR frame) and corresponding HR pixels (pixels of HR frame) the adaptation of motion compensation to SR is straightforward. The following metric function ρ is used to compare an LR block B with an HR reference block pointed by a motion vector $\vec{d}$:

$$\rho\left(B, \vec{d}\right) = \sum_{(x,y)\in B} \left| L^n(x,y) - \widetilde{L}\left(x, y, \vec{d}\right) \right|, \qquad (1)$$

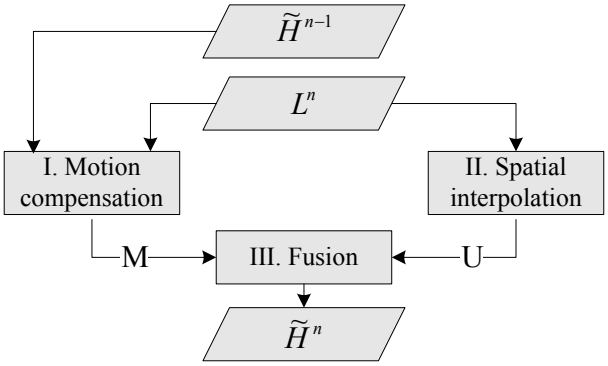

Fig. 1. Flowchart of the proposed algorithm.

where

$$\widetilde{L}\left(x, y, \vec{d}\right) = \frac{1}{4} \sum_{i,j=0}^{1} \widetilde{H}^{n-1}\left(2x + \vec{d}_x + i, 2y + \vec{d}_y + j\right) \qquad (2)$$

is the pixel of a reference block converted to low resolution using bilinear downsampling. Thus, the metric is an adaptation of the Sum of Absolute Differences (SAD) metric to the SR framework.

After a motion vector is estimated for each block, the HR motion-compensated frame M is built from the HR reference blocks.

2.3. Spatial Edge-Directed Interpolation

Lattices of LR and HR pixels are illustrated on Fig. 2; LR pixels are shown as circles, HR pixels – as squares. For LR pixel i the set of corresponding HR pixels is $\Pi(i) = \{j_k(i)\}_{k=1}^4$. Consider an area Ω formed by LR pixels i,k,l,m with intensities A, B, C, D. At first, the luminance variance of Ω is calculated. If it is less than a certain threshold T_1, the area is considered uniform, and bilinear interpolation is used for HR pixels belonging to Ω. Otherwise, two possible cases are considered: an edge passes through Ω, or Ω contains texture. To determine the right case the following values are compared:

$$d_1 = |B - C|, \quad d_2 = |A - D|,$$
$$v = \left(|A - C| + |B - D|\right)/2, \quad h = \left(|A - B| + |C - D|\right)/2. \qquad (3)$$

- If $d_2 > d_1 + T_2$, where T_2 is a threshold, the edge passes through LR pixels B and C (this case is depicted on Fig. 2). Pixels F and G are interpolated from B and C using linear interpolation. To obtain HR pixel H value, an imaginary line is drawn parallel to the edge through the pixel (dashed line in Fig. 2). Let H_1 and H_2 be the luminance values at intersections of the line with LR lattice. H_1 is linearly interpolated from C and D; H_2 is interpolated from B and D. And finally, H is linearly interpolated from H_1 and H_2. Pixel E is processed in the same way.

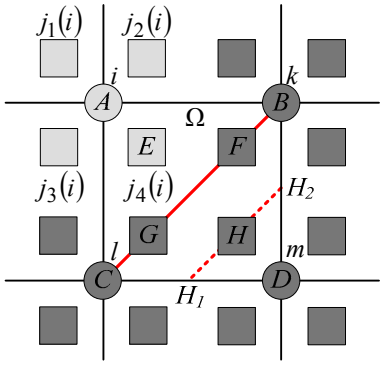

Fig. 2. Correspondence between LR and HR pixels.

- Else if $d_1 > d_2 + T_2$ the edge passes through LR pixels A and D, and the interpolation is performed similarly.
- Else if $|v - h| > T_2$ the horizontal or vertical edge is present, and the aforementioned interpolation technique is applied.
- Else there is no dominant edge direction, and Ω is considered belonging to the textured area. In this case Lanczos filter of radius 4 (lanc4) is used to obtain HR pixels' values.

2.4. Images Fusion via Classification

After M and U HR images are built, a per-pixel fusion process is performed to construct an output HR frame. We introduce a probabilistic framework for the fusion process, modeling an unknown HR pixel value H_j^n (the lower index represents the pixel number) by a random variate $\hat{H}_j^n$ for which two possible cases are considered:

$$\hat{H}_j^n \in \left\{\hat{M}_j = M_j + \Delta_{i\,j}^M, \hat{U}_j = U_j + \Delta_i^U\right\}, \quad j \in \Pi(i), i = 1,\ldots,\left|L^n\right|. \tag{4}$$

$\hat{M}_j$ and $\hat{U}_j$ are the pixels of M and U, corrected according to the upscaling errors Δ^M and Δ^U, which are defined as

$$\Delta_i^M = L_i^n - \underset{j\in\Pi(i)}{\text{mean}} M_j, \quad \Delta_i^U = L_i^n - \underset{j\in\Pi(i)}{\text{mean}} U_j. \tag{5}$$

It is assumed that, for each LR pixel i, all four pixels from $P(i)$ are simultaneously taken either from $\hat{M}$ or from $\hat{U}$.

Thus, the AdaBoost classifier can be employed to estimate the class c_i for each LR pixel i, where $c_i = 1$ if pixels from $P(i)$ are taken from $\hat{M}$, and $c_i = -1$ otherwise. The training set was constructed as follows: 20000 LR pixels were selected from the video sequences described in Section 3. For each LR pixel t from the training set the class was determined in the least squares sense, i.e., the following sums of squared errors were calculated:

$$SSE^{\hat{M}} = \sum_{j\in\Pi(t)}\left(H_j^n - \hat{M}_j\right)^2, SSE^{\hat{U}} = \sum_{j\in\Pi(t)}\left(H_j^n - \hat{U}_j\right)^2. \tag{6}$$

The class that corresponds to the lower sum was chosen. A Classification And Regression Tree (CART) of depth 5 was used as a weak learner.

The feature vector X consists of five features. The first two are the upscaling errors (5). The third is the luminance variance of neighboring LR pixels. The fourth is the sum of variances of the motion vector coordinates, calculated for neighboring blocks. The fifth feature is derived as follows: the difference between the sums of squared errors (6) is approximated by substituting L_i^n instead of H_j^n for each j from $\Pi(i)$:

$$SSE^{\hat{M}} - SSE^{\hat{U}} = \sum_{j\in\Pi(i)}\left(\hat{M}_j - \hat{U}_j\right)\left(\hat{M}_j + \hat{U}_j - 2H_j^n\right) \approx \approx \sum_{j\in\Pi(i)}\left(\hat{M}_j - \hat{U}_j\right)\left(\hat{M}_j + \hat{U}_j - 2L_i^n\right) = X_5. \tag{7}$$

Three boosting iterations were performed to train a classifier. A small number of weak learners in the committee ensured the low computational complexity of the classifier.

From our experiments, better results can be achieved, if H_j^n is modeled by the expectation of $\hat{H}_j^n$, i.e., by

$$\tilde{H}_j^n = \mathrm{E}\hat{H}_j^n = p_i\hat{M}_j + (1-p_i)\hat{U}_j, j \in \Pi(i), i = 1,\ldots,\left|L^n\right|. \tag{8}$$

Here, $p_i = P(c_i = 1 \mid X)$ is the class conditional probability, which, according to [10], can be written as

$$p_i = 1/(1 + \exp(-2F(X))), \tag{9}$$

where $F(X)$ is the classifier output. Such an approach can be somewhat argued by the fact that in cases where $F(X)$ is close to 0, it is difficult to determine the correct class. And in such cases (8) leads to the averaging of $\hat{M}_j$ and $\hat{U}_j$, which is slightly better than using one of them. Equation (9) can be presented in a more general form:

$$p_i = 1/(1 + \exp(-\alpha F(X))), \tag{10}$$

where $\alpha > 0$ is the "aggressivity" parameter.

For the enlargement of chrominance planes the fusion process (8) is applied with the same p_i values as are used for the luminance plane.

3. EXPERIMENTAL RESULTS

We used a test set of 9 HR video sequences in 1280×704 resolution exhibiting different types of motion and texture. LR video was derived from them applying bilinear downsampling by a factor of 1/2. For comparison, video

TABLE I
Comparison of Y-PSNR (dB) for Various Methods

Method	Video Sequence No.								
	1	2	3	4	5	6	7	8	9
Bicubic	31.36	31.54	39.46	29.85	24.56	31.98	35.10	40.27	28.52
Lanc4	32.03	32.58	40.36	30.81	25.33	32.69	36.43	41.27	29.48
Proposed EDI	32.23	32.63	40.62	31.23	25.33	32.94	36.44	41.31	29.80
Proposed SR with Lanc4	34.50	35.38	41.25	32.16	27.40	34.60	38.59	42.82	31.35
Proposed SR with EDI	34.85	35.47	41.39	32.47	27.41	34.85	38.62	42.91	31.82

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

Fig. 3. Visual quality comparison. (a) Ground truth; (b) lanc4 filter; (c) bicubic filter; (d) proposed SR with EDI.

upscaling was performed by the bicubic and lanc4 filters, the proposed EDI, and the proposed SR which was tested with different spatial interpolators: lanc4 filter and EDI. The results were obtained using $T_1 = 5, T_2 = 25, \alpha = 19$. Next, the Y-PSNR measure was calculated for each of the compared methods using ground truth HR video as a reference (pixels from the training set were not used in the calculation). The results of the comparison are presented in Table I. As can be seen, using EDI in the spatial interpolation step of SR is definitely better than using lanc4. In both cases the proposed SR maintains an acceptable processing speed of 2 frames per second, achieved on a single-core Athlon64 3600+ computer using a non-optimized C implementation.

An example of the application of the proposed SR algorithm is presented in Fig. 3 together with the results of the lanc4 and bicubic filters. For clarity, the presented images are enlarged two times by a nearest neighbor filter. The proposed SR algorithm provides a sharper and more detailed picture in comparison with other methods.

4. CONCLUSION

In this paper, we present a novel super-resolution method intended for fast video resolution enhancement. Our method provided better quality than did frequently used single-frame enlargement methods, as proven by both objective and subjective comparisons. The performance and the processing speed of our method can be further improved by using cascade classifiers and by employing more advanced edge-directed interpolation. Moreover, motion compensation can be applied to just highly textured areas, thus increasing the speed even further. All these directions form a subject of our future research.

REFERENCES

[1] S. Farsiu, M. Elad, and P. Milanfar, "Video-to-Video Dynamic Super-Resolution for Grayscale and Color Sequences", *EURASIP Journal on Applied Signal Processing,* No. Article ID 61859, pp. 1-15, 2006.

[2] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multi-frame super-resolution," *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1327–1344, October 2004.

[3] Z. Jiang, T.T. Wong, and H. Bao, "Practical super-resolution from dynamic video sequences", in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 549-554, June 2003.

[4] M. Irani and S. Peleg, "Improving Resolution by Image Registration", *Journal of Computer Vision, Graphics, and Image Processing*, vol. 53(3), pp. 231–239, May 1991.

[5] W. T. Freeman, T. R. Jones, and E. C. Pasztor. "Example-based super-resolution", in *IEEE Computer Graphics and Applications*, vol. 22(2), pp. 56–65, March/April 2002.

[6] Q. Wang, X. Tang, and H. Shum, "Patch Based Blind Image Super Resolution," in Proc. *IEEE International Conference on Computer Vision*, vol. 1, pp. 709-716, October 2005.

[7] X.Li, "Super-Resolution for Synthetic Zooming", *EURASIP Journal on Applied Signal Processing,* No. Article ID 58195, pp. 1-12, 2006.

[8] C. Bishop, A. Blake, and B. Marthi, "Super-resolution enhancement of video", in *C. M. Bishop and B. Frey (Eds.), Proc. of the Nineth Int. Workshop on Artificial Intelligence and Statistics,* January 2003.

[9] V. Cheung, B. J. Frey, and N. Jojic, "Video epitomes", in Proc. *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 42-49, June 2005.

[10] J. H. Friedman, T., Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting", *Dept. of Statistics, Stanford University Technical Report*, 1998.