2017 International Symposium on Nonlinear Theory and Its Applications,
NOLTA2017, Cancun, Mexico, December 4-7, 2017

Super-resolution Image Reconstruction Method Based on Non-linear Image Enhancement Filters

Tomio Goto, Tsubasa Munezawa and Satoshi Hirano

Faculty of Engineering, Nagoya Institute of Technology
Gokiso-cho Showa-ku, Nagoya 466-8555, Japan
Email: t.goto@nitech.ac.jp

Abstract– Super-resolution is one of the most important technologies for image processing because high resolution displays have been widely used although low resolution moving pictures have been playing. A lot of super-resolution methods have been proposed and there are two kinds of methods; reconstructed type and leaning type of super-resolution. The main problem for the learning type of super-resolution is long computational time to search its dictionary. In this study, we propose a novel super-resolution method based on non-linear image enhancement filters and experimental results show that the performance of super-resolution is improved.

1. Introduction

We have been developing a super-resolution system, utilizing total variation (TV) regularization, for moving picture media [1][2][3]. We have achieved a system that has near-adequate performance with a realistic computational time [3]. In parallel, we have developed an MPEG2 compression artifact reduction system utilizing TV regularization [4]. In this paper, we propose a new system that combines our super-resolution [3] and noise reduction systems [4] for the 4K-HDTV application.

In the application of super-resolution systems for HDTV, the ASIC approach is considered one solution. However, it requires a large development cost and is not suitable for products in the non-volume zone. Therefore, we consider using a parallel-processor chip as a more realistic solution.

We tried to implement our system on a CELL processor that was developed as a multi-core graphical processor for game machines [5]. However, because of a limitation of the computational ability of the CELL processor, we could not obtain sufficient system performance.

In this paper, we propose implementing our new system on the latest GPU. Because the system is complicated and the number of pixels for 4K-HDTV is large, we require a special programing technique for the GPU and suitable system revisions for the GPU architecture. Our goal is to obtain a one-frame computational time of less than 16.7 ms, which is the HDTV one-frame time. In this paper, we describe our new system and several GPU programming techniques that we use to achieve our goal.

Super-resolution (SR) technology generates an estimated high-frequency signal outside the original Nyquist frequency region on the increased pixel-number display. The regenerated signal is simply an estimation or most-likely image for human sense. It is a different technology from de-convolution that restores the original image within the Nyquist frequency region.

For one-frame super-resolution, there are two approaches. The first is the learning-based or example-based methods that utilize a high-resolution-image data base [6][7][8]. The other is edge enhancement utilizing a non-linear filter such as total variation (TV) regularization [9][10].

2. Proposed Method

2.1. System Block Diagram

Figure 1 shows the block diagram of the complete system. The input signal is decomposed by a TV filter into the structure component and the texture component. The texture component is first processed by a noise reduction filter, then up-sampled by a bi-cubic filter, and finally processed by the pulse enhancement filter (PEF). The structure component is edge-enhanced by the shock filter. Then the two signals are combined.

2.2. TV Regularization

Total variation (TV) regularization is a minimization problem of $F(u)$ given by the ROF model [11] described by Eqs. (1) and (2),

$$\inf_u F(u) = \sum_{i,j} \left|\nabla u_{i,j}\right| + \lambda \sum_{i,j} \left|f_{i,j} - u_{i,j}\right| \tag{1}$$

$$v = f - u \tag{2}$$

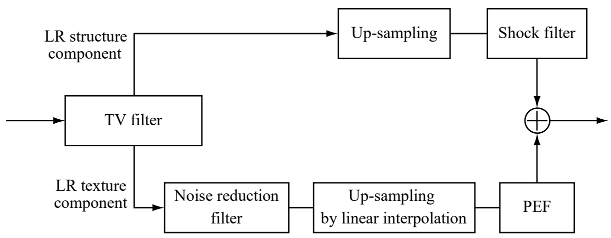

Fig. 1. Super-resolution system block diagram.

where f is the input signal, u is a variable called the structure component, v is called the texture component, and λ is a small positive constant.

In order to solve Eq. (1), Chambolles' well-known projection method [12] given by Eqs. (3) and (4) is utilized, where p is a dual vector.

Equation (3) requires an iterative calculation. Usually, 20-50 iterations are required to obtain acceptable performance.

$$p_{i,j}^{(n+1)} = \frac{p_{i,j}^{(n)} + \tau/\lambda \left\{\nabla\left(f + \lambda div\, p^{(n)}\right)\right\}_{i,j}}{\max\left[1, \tau/\lambda \left|\nabla\left(f + \lambda div\, p^{(n)}\right)_{i,j}\right|\right]} \tag{3}$$

$$v = \lambda div\, p \tag{4}$$

However, this method has particularly good features compared with linear filtering. One is that the edge component is included in the structure component and the noise and high frequency signal are included in the texture component. It is impossible to separate the edge signal from the high frequency signal by any linear filters.

In this paper, we call it the TV filter that generates the structure component u and the texture component v by TV regularization process according to Eqs. (3) and (4).

2.3. Edge Component Enhancement

A shock filter [13] is a type of PDE (partial differential equation) filter given by Eq. (5), where $u_{i,j}^{(n)}$ is a pixel value and α is a positive small constant.

This filter is a non-linear filter and has the effect of sharpening the edge component. To improve the jagged artifact, the Laplacian $\Delta u = u_{xx} + u_{yy}$ is replaced by $v_{\eta\eta} = (K_\sigma * u)_{\eta\eta}$, where K_σ is the Gaussian kernel and η is the gradient operator as shown in Eq. (6) [14][15].

$$u_{i,j}^{(n+1)} = u_{i,j}^{(n)} - \alpha \times sign\left(\Delta u_{i,j}^{(n)}\right)\left|\nabla u_{i,j}^{(n)}\right| \tag{5}$$

$$u_{i,j}^{(n+1)} = u_{i,j}^{(n)} - \alpha \times sign\left(v_{i,j}^{(n)}\right)_{\eta\eta}\left|\nabla u_{i,j}^{(n)}\right| \tag{6}$$

The shock filter shows a very positive effect on the edge sharpness, but it generates an artifact on the pulse shape signal as shown in Fig. 2.

Therefore, its application has been very limited to areas such as blind-deconvolution. We discovered that the shock filter is very effective in our system because the structure component consists of edge signals but not pulse signals. Hence, the combination of the TV filter and shock filter is considered a very effective and realistic solution [3]. Several additional improvement methods were proposed in [3].

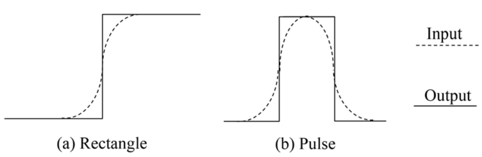

Fig. 2. Output signal of shock filter

From our experiments, we found that the proper iterative number n is usually around five. However, by careful parameter setting, even the $n = 1$ case shows sufficiently good performance.

2.4. Pulse Component Enhancement

The texture component consists primarily of pulse and noise. We tried a learning-based method, but concluded that it was not effective for our application. Therefore, we adopted a non-linear pulse enhancement filter (PEF).

The basic idea is to calculate the N-th power of a signal as shown by Eq. (7) [3].

$$y = A|x|^N sign(x) \tag{7}$$

The coefficient A is given by Eq. (8).

$$A = k\left|x_{\max}\right|^{1-N} \tag{8}$$

Figure 3 shows the waveform of the N-th power filter. The pulse width is narrowed by the N-th power operation and normalization. This means that the N-th power filter generates a high frequency component beyond the original Nyquist frequency.

Based on our experience, the suitable value of N is from 1.2 to 2.0 in order to obtain natural pictures. The value of k in Eq. (8) is set to 1.5 to 2.0.

2.5. Noise Component Reduction

The compression noise such as block noise and mosquito noise is almost decomposed into the texture component. Figure 4 shows our compression noise

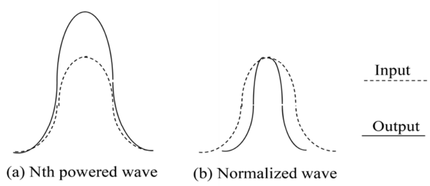

Fig. 3. Output signal of PEF

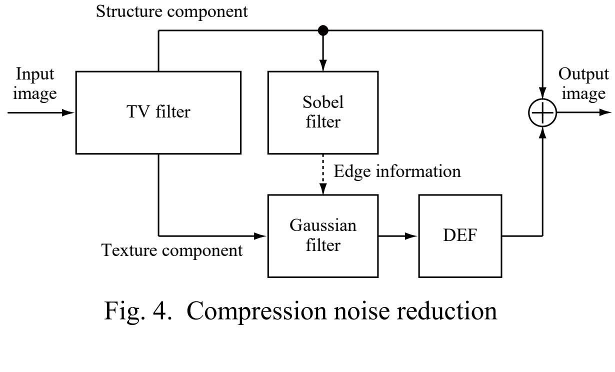

Fig. 4. Compression noise reduction

reduction system [4]. The block noise is reduced by the standard de-blocking filter (DEF: de-blocking edge filter) and the mosquito noise is reduced by the LPF that is set to affect only the edge portion according to the information from the structure component.

The main block of the TV filter can be used in both the super-resolution and noise reduction systems. Therefore, the noise reduction system is fully integrated in the super-resolution system as shown in Fig. 1.

3. GPU Implementation

3.1. Features of GPU Implementation

A GPU is a parallel processor that was developed as an accelerator for computer graphics [16]. A GPU has multiple core processors grouped into several streaming multi-processors (SMP) with an on-chip shared memory. Input image data is divided into several blocks and processed in parallel by each SMP. This architecture realizes very high-speed image-processing execution. However, for our application, there are two bottlenecks with a GPU. One is the data transfer between the on-chip shared memory and outside global memory. If this operation occurs frequently, the total computational time is increased considerably even with the high speed SMP computation. Another is the branch instruction. This is done sequentially and hence cannot use the GPU's parallel processing capability.

3.2. TV Filter and Shock Filter

The TV filter given by Eq. (3) and the shock filter given by Eq. (6) require iterative calculations that are different from ordinary convolutional linear filters. In the iterative operation, the data from the shared memory must be transferred to the global memory and then rewritten for each iterative operation. This operation increases the computation time significantly. We note that Eqs. (3) and (6) are simple calculations between neighboring pixels. Therefore, we set an overlap area between each calculation block as shown in Fig. 5. Because of this overlap calculation, the data transfer between the shared memory and global memory at each iterative operation can be avoided. The effect of this method is proportional to the iteration number.

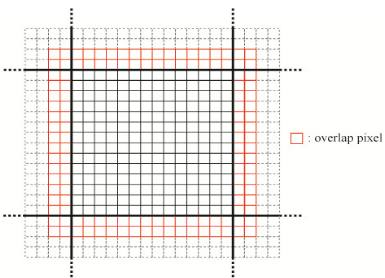

Fig. 5. Block calculation

3.3. Pulse Enhancement Filter (PEF)

The pulse enhancement filter (PEF) requires a calculation of Eqs. (7) and (8). The heaviest task in the PEF processing is to search for the top of each pulse and obtain the peak value of Eq. (8). In the original PC program, a peak search of the many pixels in the large area is carried out to determine the peak value of each pulse. This calculation generates many branch instructions that are not suitable for the parallel GPU architecture.

3.4. Convolution

There are a lot of convolution calculations in this system. We adopt an effective convolution method proposed in [17], [18]. This method prefetches image regions to register, and do more work per thread with fewer threads. This method gives more flexibility to the compiler, and reduces the total number of requests for data of the off-chip memory. For example, the processing time of 5x5 Gaussian filter is reduced to half.

4. Experimental Results

4.1. Image Quality

Figure 6 shows an image of the super-resolution (SR) of this system implemented on a GPU. The picture quality is almost the same as in the PC calculation.

Figure 7 shows the performance of MPEG2 compression noise reduction (NR). We can see the MPEG2 noise such as block noise and mosquito noise are emphasized by SR, and noise reduction (NR) is necessary for super-resolution (SR) of compressed pictures.

4.2. Computational Time

Table 1 shows a comparison of GPU computational time. The GPU is an NVIDIA Geforce GTX TITAN [19]. A CUDA development environment is used [20]. The original input image is a one-frame HDTV image with 1920 x 1080 pixels, and the SR output image is 4K-HDTV image with 3840 x 2160 pixels.

In Table 1, "conventional method" means the direct implementation of the PC program on the GPU. "Proposed method" is the improved programing based on the proposed methods described in Sections 3.2-3.4.

The time-reductions are remarkable in the TV filter using the method of Section 3.2 and DEF using the method of Section 3.3. The total processing time is reduced from 24.72 ms to 13.07 ms. This value is less than the one-frame HDTV time of 16.7 ms.

5. Conclusion

We have proposed a novel super-resolution system accompanied by a compression noise reduction function for 4K-HDTV super-resolution. This system consists of non-linear filters such as TV regularization, shock, and pulse enhancement filters. We have proposed an optimum GPU programming design for each non-linear filter, such as overlapped block division, block searching, and efficient convolution calculation. As a result, we have succeeded in reducing the total computational time to 13.07 ms, which is less than the one-frame HDTV time of 16.7 ms.

It is proved, therefore, that the proposed super-resolution system can be implemented into 4K-HDTV using a standard GPGPU.

(a) Linear interpolation image (b) SR image

Fig. 6. Super-resolution image

(a) Original image (b) MPEG2 image

(c) SR image without NR (d) SR image with NR

Fig. 7. Noise reduction

Table 1. Computational time [ms]

	Conventional method	Proposed method
TV filter	6.80	3.67
Shock filter	4.62	3.19
PEF	11.35	3.88
Gaussian and Sobel filter	1.12	0.53
Others	2.22	2.08
Total	24.72	13.07

References

[1] A. Yoshikawa, S. Suzuki, T. Goto, S. Hirano and M. Sakurai, "Super Resolution Image Reconstruction using Total Variation Regularization and Learning-based Method", *IEEE International Conference on Image Processing* (*ICIP2010*), pp.1993–1996, Sep. 2010.

[2] M. Sakurai, Y. Sakuta, M. Watanabe, T. Goto and S. Hirano, "Super-resolution utilizing Total Variation Regularization and Shock Filter", *IEEE International Conference on Image Processing* (*ICIP2012*), pp.2221–2224, Oct. 2012.

[3] M. Sakurai, Y. Sakata, M. Watanabe, T. Goto, S. Hirano, "Super-Resolution through Non-Linear Enhancement Filters", *IEEE International Conference on Image Processing* (*ICIP2013*), pp.854–857, Sep. 2013.

[4] T. Goto, Y. Kato, S. Hirano, M. Sakurai and T. Q. Nguyen, "Compression Artifact Reduction based on Total Variation Regularization Method for MPEG-2", *IEEE Trans. on Consumer Electronics*, Vol.57, No.1, pp.253–259, Mar. 2011.

[5] Y. Sakuta, A. Tsutsui, T.Goto, M. Sakurai, M. Sakai, "Super-Resolution utilizing Total Variation Regularization on CELL Processor", *IEEE International Conference on Consumer Electronics* (*ICCE2012*), pp.729–730 Jan. 2012.

[6] W. T. Freeman, T. R. Jones, and E. C. Paztor, "Example Based Super-resolution", *IEEE Computer Graphics and Applications,* Vol.22, No.2, pp.56–65, Apr. 2002.

[7] H. Chang, D. Y. Young and Y. Xiong, "Super-Resolution through Neighbor Embedding", *IEEE International Conference on Computer Vision and Pattern Recognition* (*CVPR2004*), Vol.1, pp.275–282, Jul. 2004.

[8] J. Yang, J. Wright, Y. Ma and T. S. Huang, "Image Super-Resolution Via Sparse Representation", *IEEE Trans. on Image Processing*, Vol.19, No.11, pp.2861–2837, Nov. 2010.

[9] A. Marquina, S. J. Osher, "Image super-resolution by TV-regularization", *Journal of Scientific Computing*, Vol.37, No.3, pp.367–382, Dec. 2008.

[10] T. Saito and T. Komatsu, "Image Processing Approach Based on Nonlinear Image-Decomposition", *IEICE Trans. Fundamentals*, Vol.E92-A, No.3, pp.696–707, Mar. 2009

[11] L. Rudin, S. Osher and E. Fetami, "Nonlinear Total Variation based Noise Removal Algorithm", *Physica D*, Vol.60, pp.259–268, Nov. 1992.

[12] A. Chambolle, "An Algorithm for Total Variation Minimization and Applications", *SIAM J. Mathematical Imaging and Vision*, Vol.20, No.1, pp.89–97, Jan. 2004.

[13] S. Osher and L. I. Rudin, "Feature-oriented Image Enhancement using Shock Filters", *SIAM J. Numerical Analysis*, Vol.27, No.4, pp.919–940, Aug. 1990.

[14] L. Alvarez and L. Mazorra, "Signal and Image Restoration using Shock Filters and Anisotropic Diffusion", *SIAM J. Numerical Analysis*, Vol.31, No.2, pp.590–605, Apr. 1994.

[15] C. Ludusan, O. lavialle, S. Pop, R. terebes and M. Borda, "Image Enhacement using a New Shock Filter Formalism", *ACTA Technica Napocensis*, Vol.50, No.3, pp.27–30, 2009.

[16] David B. Kirk and Wen-mei W. Hwu, "Programming Massively Parallel Processors", *Morgan Kaufmann*, Feb. 2010.

[17] F.N. Iandola, D. Sheffield, M. J. Anderson, P. M. Phothilimthana, and K. Keutzer, "Communication–mnimizing 2D Convolution in GPU Registers", *IEEE International Conference on Image Processing* (*ICIP2013*), pp.854–857, Sep. 2013.

[18] I. Back, T. Foley, D. Horn, J. Sugerman, K.Fatahlian, M.Houston, et al., "Brook for GPUs: Stream Computing on Graphics Hardware", *ACM Trans. on Graphics*, Vol. 23, Issue. 3, pp.777–786, Aug. 2004.

[19] E. Lindholm, J. Nickolls, S. Oberman and J. Montrym, "NVIDIA Tesla: A Unified graphics and computing architecture", *IEEE Micro*, Vol.28, Issue. 2, pp.39–55, Apr. 2008.

[20] J. Sanders and Edw ard Kandrot, "CUDA by Example: An Introduction to General-Purpose GPU Programing", *SIAM J. Numerical Analysis*, Vol.31, No.2, pp.590–605, Apr. 1994.