Pixel enlargement in high-speed camera image acquisition based on 3D sparse representations

A. Hirabayashi, N. Nogami, J. White
Ritsumeikan University
College of Information Science and Engineering
Kusatsu, Shiga 525-8577, Japan
akirahrb@media.ritsumei.ac.jp

L. Condat
University of Grenoble Alpes
GIPSA-lab, F38000 Grenoble, France
laurent.condat@gipsa-lab.grenoble-inp.fr

***Abstract*—We propose an algorithm that enhances the pixel number in high-speed camera image acquisition. In high-speed cameras, there is a principle problem that the number of pixels reduces when the number of frames per second (FPS) increases. To suppress this problem, we first propose an optical setup that randomly selects some percent of pixels in an image. Then, the proposed algorithm reconstructs the entire image from the selected partial pixels. In this algorithm, we exploit not only sparsity within each frame but also sparsity induced from the similarity between adjacent frames. Based on the two types of sparsity, we define a cost function for image reconstruction. Since this function is convex, we can find the optimal solution by using a convex optimization technique, in particular the Douglas-Rachford Splitting method, with small computational cost. Simulation results show that the proposed method outperforms a conventional method for sequential image reconstruction with sparsity prior.**

I. Introduction

High speed cameras are capable of capturing images more than one hundred frames per second (fps). Originally they were used for engineering measurements, especially in the automotive industry. Recently, they have been used for sports training or entertainment. High spec products can capture 4.91 mega (2560×1920) pixel images by two thousand fps [1]. For casual purposes, “iPhone 6 plus” [2] and “Go Pro Hero 4” [3] are also useful because they are able to capture images at 240 and 120 fps, respectively.

One issue of high speed cameras is the decrease of pixels when fps increases. For example, in the above camera, the pixel number decreases from 2 mega, to 920 kilo pixels when fps increases form 4,500 to 10,000. The reason of this phenomenon is that time for swipe out is proportional to the number of image pixels while the increase of fps number suppresses the time for swipe out. Our goal is to keep the pixel number as high as possible even when fps increases. Our idea is that: a camera captures randomly selected pixels only, say 25%, by an optical setup such as the one shown in Fig. 1. This is the multiple pixel version of the single pixel camera [4]. In particular, we adopt block random selection, as shown in Fig. 2. That is, an image is divided into pixel blocks of, say 2×2, and one pixel out of the block is randomly selected. Then, an image processing technique recovers the entire original image by filling in the missing pixels. It should be noted that this is spatio-temporal image reconstruction problem. Thus, we can exploit sparsity not only within images, but also along the time axis. There is a lot of research relevant to this problem [5]–[10]. Early work by Wakin *et al.* regarded this problem as a three dimensional sensing problem [5]. It is, however, difficult to implement such a sensing mechanism in high speed cameras. Kang and Lu exploited similarities between adjacent frames [6]. They changed the compression rate depending on frames (key or non-key) and used efficient optimization initial values for each. However, such change of compression rate is difficult in high speed cameras because of a limited swipe out time. Vaswani proposed methods supposing that sparsity pattern (support of the sparsifying transform vector) changes slowly over time [7], [8]. It is, however, difficult to choose appropriate thresholds for adding and deleting small part of the support. Another approach is based on dictionary learning [9]. This approach requires a high computational cost. To reduce the cost, images are divided into small patches, which results in block noise and occasionally the need for post processing [11], [12].

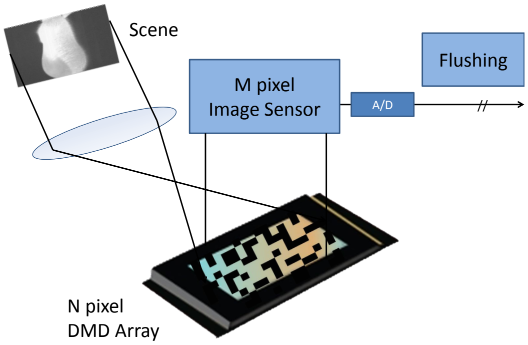


Fig. 1. Proposed optical setup for pixel enlargement in high speed camera image acquisition.

In this paper, we propose a simple, yet novel approach based on the combination of ℓ_1 norms within each frame and between adjacent frames. That is, we reconstruct images by minimizing the sum of ℓ_1 norms of sparsifying transform's coefficients and the difference of adjacent frames under the observation constraint. The latter is similar to the total variation along a time axis, which has already been implemented in *e.g.* [10]. This work did not, however, combine the total

The author thanks for Ritsumeikan Program to Support General Research Activities (Kiban-kenkyu).

variation with ℓ_1 norm within each frame. The proposed cost function amounts to the sum of three convex terms which are proximable, but not differentiable. We regard the sum of the ℓ_1 norm of the difference of adjacent frames and the observation constraint term as a single part, which is still proximable. Then, the cost function can be regarded as a sum of two proximable and non-differentiable terms. It is effectively minimized by the Douglas-Rachford splitting algorithm [13]. We show by simulations that the proposed method outperforms conventional methods [8] with small computational cost.

The rest of the present paper is organized as follows. In section 2, we formulate the image reconstruction problem and define a criterion (cost function) for the image reconstruction in this paper. In section 3, we propose a fast reconstruction algorithm based on the Douglas-Rachford splitting method. In section 4, we show the effectiveness of the proposed method by computer simulations. Section 5 concludes the paper.

II. Image Reconstruction by Convex Optimization

Suppose that a fixed high speed camera captures a scene at a high frame rate and a sequence of images $\boldsymbol{x}_r \in \mathbb{R}^N$ ($r = 1, \dots, R$) is obtained[1]. Such a sequence is sometimes stacked and forms a long vector, as seen in [10]. That approach results in a huge computational cost. We treat the sequence frame by frame as done in [8]. Pixels in the rth image $\boldsymbol{x}_r$ is randomly selected so that M pixels are remaining ($M < N$). Let A_r and $\boldsymbol{y}_r \in \mathbb{R}^M$ be a random selection matrix (M rows are randomly selected from the identity matrix of the corresponding size) and a vector consisting of the selected pixels. Then, it holds that

$$\boldsymbol{y}_r = A_r \boldsymbol{x}_r, \qquad (r = 1, 2, \dots, R). \tag{1}$$

Note that the random selection pattern in A_r is generated at every frame, not fixed. Our goal is to estimate the image sequence $\{\boldsymbol{x}_r\}_{r=1,\dots,R}$ from $\{A_r\}_{r=1,\dots,R}$ and $\{\boldsymbol{y}_r\}_{r=1,\dots,R}$. Because of this goal, we do not take blur nor noise into account.

We solve this problem by using two priors. First, we suppose that each captured image is sparse in an appropriate sparsifying transform domain, such as discrete wavelet or cosine transform domains. In simulations, we adopted DCT for the sparsifying transform since the target image sequence is about a natural scene. Second, because of the high frame rate, the difference between adjacent frames is small. Further, if only a small part in the scene is moving and other objects do not move so much, then the difference is not only small, but also sparse. Based on these two assumptions, we reconstruct the image $\boldsymbol{x}_r$ by using the following cost function for the DCT coefficient vector $\boldsymbol{u} = (u_n) \in \mathbb{R}^N$:

$$\hat{\boldsymbol{u}}_r = \mathop{\arg\min}_{A_r C^T \boldsymbol{u} = \boldsymbol{y}_r} \{ \|\boldsymbol{u}\|_1 + \lambda \|C^T \boldsymbol{u} - \hat{\boldsymbol{x}}_{r-1}\|_1 \} \;\; (r = 2, \dots, R), \tag{2}$$

where $\|\cdot\|_1$ is the ℓ_1 norm of the corresponding vector and C^T is the transpose of the two-dimensional DCT matrix C, thus the inverse transform. The rth frame $\boldsymbol{x}_r$ is then estimated by $\hat{\boldsymbol{x}}_r = (\hat{x}_{r,n}) = C^T \hat{\boldsymbol{u}}_r$. For $r = 1$, we obtain $\hat{\boldsymbol{u}}_1$ by setting 0 for λ, thus (2) amounts to the standard ℓ_1 norm minimization as in the compressed sensing [14].

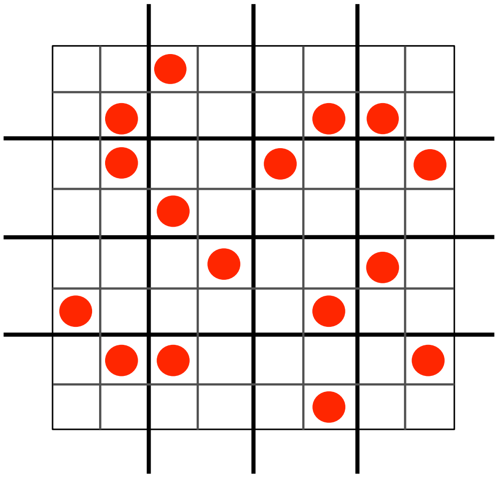


Fig. 2. Block random selection. Single pixel (shown by red) is randomly selected from each 2×2 block.

III. Reconstruction Algorithm

We solve the problem (2) by using the Douglas-Rachford splitting (DRS) algorithm [13]. Let S be a set of $\boldsymbol{u}$ satisfying $A_r C^T \boldsymbol{u} = \boldsymbol{y}_r$. This is a convex set. Then, (2) is equivalent to

$$\hat{\boldsymbol{u}}_r = \mathop{\arg\min}_{\boldsymbol{u} \in \mathbb{R}^N} \{ \|\boldsymbol{u}\|_1 + \lambda \|C^T \boldsymbol{u} - \hat{\boldsymbol{x}}_{r-1}\|_1 + \imath_s(\boldsymbol{u}) \}, \tag{3}$$

where $\imath_s(\boldsymbol{u})$ is the indicator function that takes value 0 if $\boldsymbol{u} \in S$, $+\infty$ else. Now, our problem becomes the minimization of the sum of three convex terms, which are not differentiable but proximable. As is well known, the proximity operator of the first term $\|\boldsymbol{u}\|_1$ is $\mathrm{prox}_{\theta\|\cdot\|_1}(\boldsymbol{u}) = (\mathrm{softthreshold}(u_n, \theta)) \in \mathbb{R}^N$ with $\theta = 1$, where

$$\mathrm{softthreshold}(u, \theta) = \begin{cases} u - \theta & \text{if} \quad u \geq \theta, \\ u + \theta & \text{if} \quad u \leq -\theta, \\ 0 & \text{if} \quad -\theta < x < \theta. \end{cases} \tag{4}$$

We next turn our attention to the second and the third terms. Let us denote the sum of the two terms by $g(\boldsymbol{u})$:

$$g(\boldsymbol{u}) = \lambda \|C^T \boldsymbol{u} - \hat{\boldsymbol{x}}_{r-1}\|_1 + \imath_s(\boldsymbol{u}).$$

The proximity operator of $g(\boldsymbol{u})$ can be computed by the following operations. First, apply the inverse two-dimensional DCT to $\boldsymbol{u}$ as $C^T \boldsymbol{u} \equiv \boldsymbol{v} = (v_n) \in \mathbb{R}^N$. Then, for the pixels in the mask A_r, replace the values v_n by the the corresponding element of $\boldsymbol{y}_r$. For the other pixels, apply the operation

$$v_n \leftarrow \mathrm{softthreshold}(v_n - \hat{x}_{r-1,n}, \lambda) + \hat{x}_{r-1,n}. \tag{5}$$

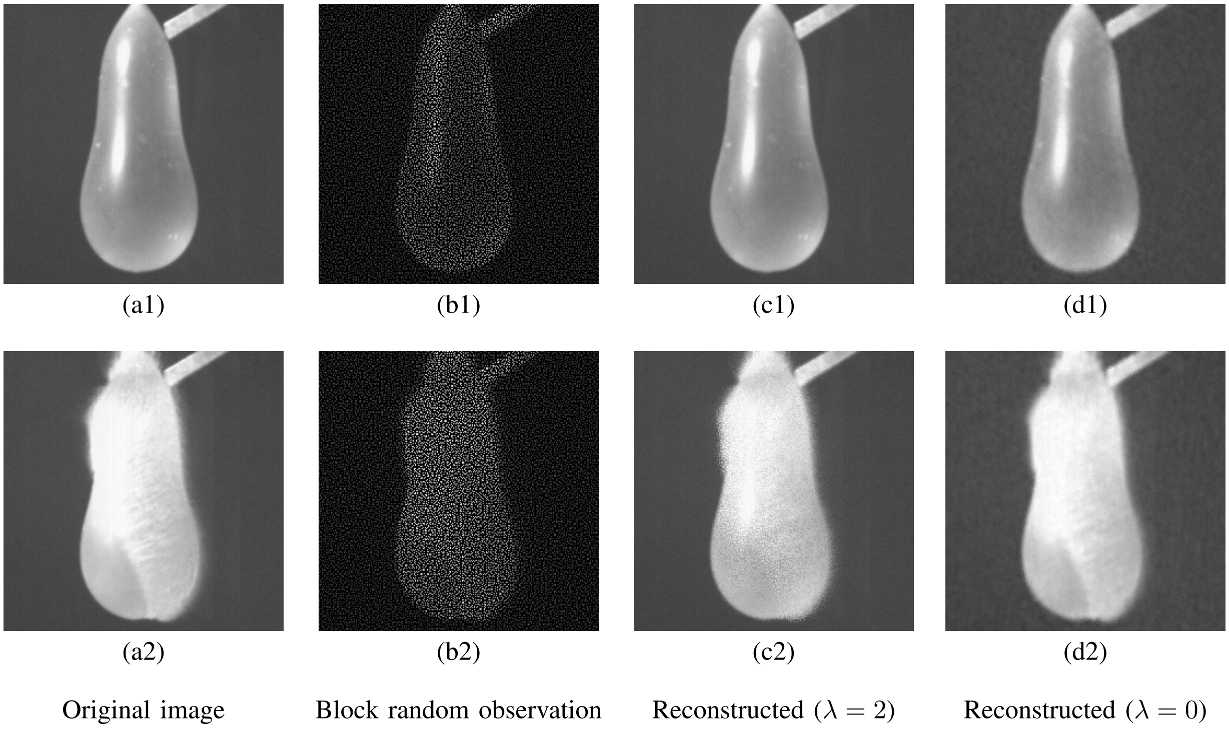
Finally, apply the two-dimensional DCT to the updated vector $\boldsymbol{v}$ for returning to the DCT domain. These operations result in $\mathrm{prox}_g(\boldsymbol{u})$.

We are now ready to solve the problem (2) by the DRS algorithm. Namely, we view (3) as

$$\hat{\boldsymbol{u}}_r = \mathop{\arg\min}_{\boldsymbol{u} \in \mathbb{R}^N} \{ \|\boldsymbol{u}\|_1 + g(\boldsymbol{u}) \},$$

where both terms are non-differentiable but proximal. Thus, the problem (2) can be solved by the following DRS algorithm:

[1] Images are raster scanned and regarded as column vectors.

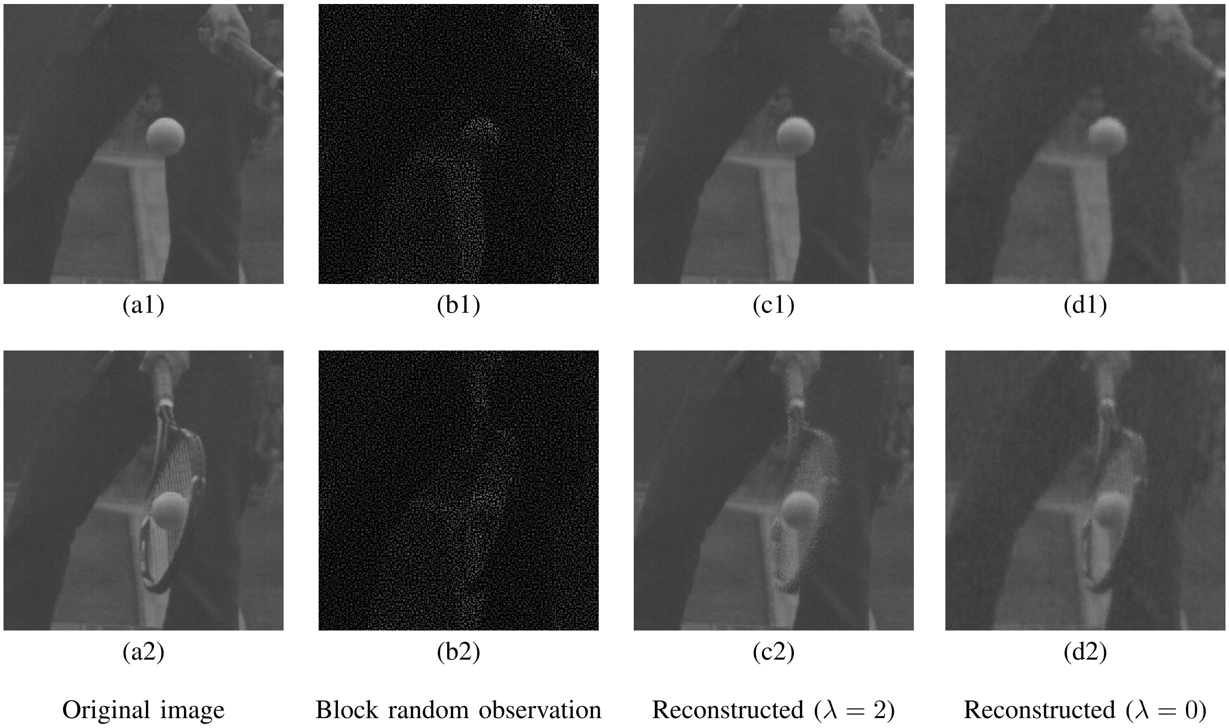


(a1) (b1) (c1) (d1)

(a2) (b2) (c2) (d2)

Original image Block random observation Reconstructed ($\lambda = 2$) Reconstructed ($\lambda = 0$)

Fig. 3. The reconstructed image with λ=2 and 0 for the image sequence of “water balloon”.

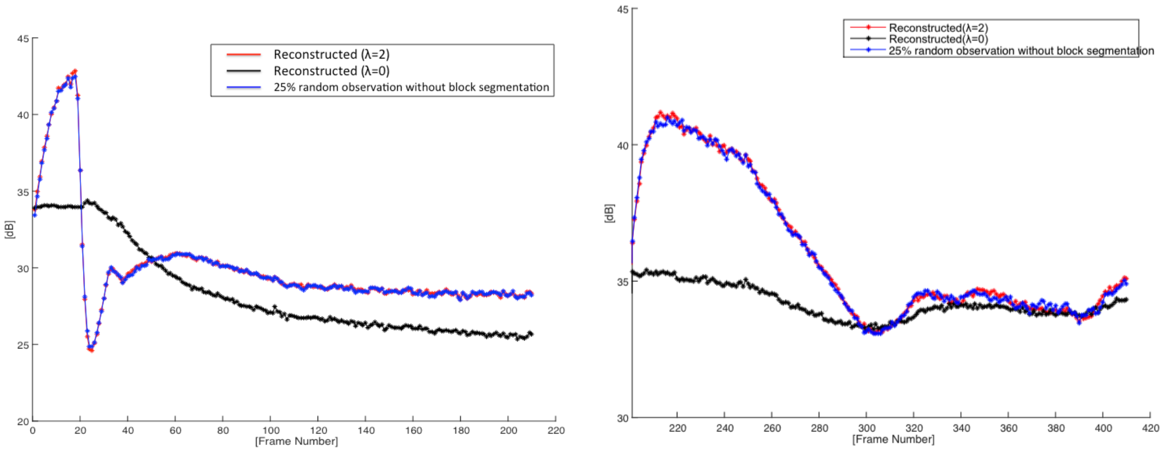


(a1) (b1) (c1) (d1)

(a2) (b2) (c2) (d2)

Original image Block random observation Reconstructed ($\lambda = 2$) Reconstructed ($\lambda = 0$)

Fig. 4. The reconstructed image with λ=2 and 0 for the image sequence of “tennis”.



(a) “water balloon”

(b) “tennis”

Fig. 5. PSNR [dB] of the reconstructed images with and without reference to the adjacent frame.

```
**Algorithm 1:** Image recovery for $r$th frame
Input: $\boldsymbol{y}_r$, $A_r$, $\hat{\boldsymbol{x}}_{r-1}$
Output: $\hat{\boldsymbol{x}}_r$
1. Set $\gamma > 0$, $\delta \in ]0, 2[$
2. Set zero vector for $\boldsymbol{v}$ as an initial value.
3. Repeat the following two operations:
       $\boldsymbol{u}_r \leftarrow \mathrm{prox}_{\gamma\|\cdot\|_1}(\boldsymbol{v})$,
       $\boldsymbol{v} \leftarrow \boldsymbol{v} + \delta\{\mathrm{prox}_{\gamma g}(2\boldsymbol{u}_r - \boldsymbol{v}) - \boldsymbol{u}_r\}$,
   until a stopping condition is met.
4. Compute $\hat{\boldsymbol{x}}_r = C^T \boldsymbol{u}_r$
```

The proximity operator $\mathrm{prox}_{\gamma g}$ can be computed simply by replacing λ in (5) by $\gamma\lambda$. The parameters γ and δ control the speed of convergence. Through several trials, we adopted $\gamma = 20$ and $\delta = 1.7$ in the following simulations. Several stopping criteria exist in the literature. Here, we simply stop the algorithm after twenty iterations, since we observed that this is sufficient to ensure convergence with high precision. For simple presentation, we explained the algorithm using the raster scan, which results in huge sensing and DCT matrices. However, we implemented the program by Matlab exploiting two-dimensional expression to reduce both computational time and memory use. Thus, we can compute relatively high dimensional images, say 256×256, which was not computed by the method in [8] provided from [15].

IV. Simulations

A water balloon bursting scene was captured indoors by Optronis CR450x3, nac Image Technology at 6,000 fps, and 210 frames of uncompressed 256×256 images were obtained. Fig. 3 shows those images in column (a). The image (a1) is a frame of little change while (a2) is a frame of intense change. Similarly, a tennis playing scene was captured outdoors by the same camera and a sequence of images of the same specifications was obtained. Fig. 4 shows those images in column (a). The image (a1) shows a frame with less motion while (a2) shows that of impact.

Pixels in these images are selected in the block random manner, as shown in Fig. 2. The block size was 2×2. That is, we have 25% pixels of the entire image as measurements. Images in column (b) are the randomly selected pixels.

The column (c) shows the reconstructed images by the proposed method, using $\lambda = 2$. The images in column (d) are reconstructed images without reference to the adjacent image, using $\lambda = 0$. One can watch videos of the image sequences from http://www.ms.is.ritsumei.ac.jp/HSC/. Comparing (c) with (d) in Fig. 3, the reconstructed image for (a1) is of better quality than those obtained not referring to the adjacent frames. Even when image change is intense as in (a2), the background of the reconstructed images by the proposed method is of good quality, while we can see mosaic artifact around splashing water areas. As for Fig. 4, the image (c1) is of good quality while the image (c2) shows a blurry area in the racket. Fig. 5 shows the peak signal to noise ratio (PSNR) in dB of the reconstructed image $\hat{\boldsymbol{x}}_r$ with respect to the frame number r, defined as

$$\mathrm{PSNR}_r = 20 \log_{10} \frac{255\sqrt{N}}{\|\hat{\boldsymbol{x}}_r - \boldsymbol{x}_r\|_2} [\mathrm{dB}],$$

where N is the number of pixels. The red line is the values of PSNR_r of the reconstructed images with reference to the adjacent image. The black line shows those of the reconstructed images without referring to the adjacent image. Figure (a) shows the result for “water balloon”. The maximum and minimum of PSNR_r were 44.86dB and 24.39dB, respectively, with an average of 30.15dB when the image was reconstructed by the proposed method with reference to the adjacent image. On the other hand, the maximum and minimum were 34.38dB and 25.34dB, with an average of 28.38dB when the image was reconstructed without reference to the adjacent image. The average for the method with reference is higher than that without reference by 1.7dB. Note that the reconstructed images without reference were better than the referenced results between 20 and 40 frames. This is because the change of image is intense within these frames. The blue line shows PSNR_r of images reconstructed by the proposed method with reference to the adjacent frame from 25% random observation *without* block segmentation. The maximum and minimum were 44.65dB and 24.15dB, respectively, and the average was 30.13dB. Hence,

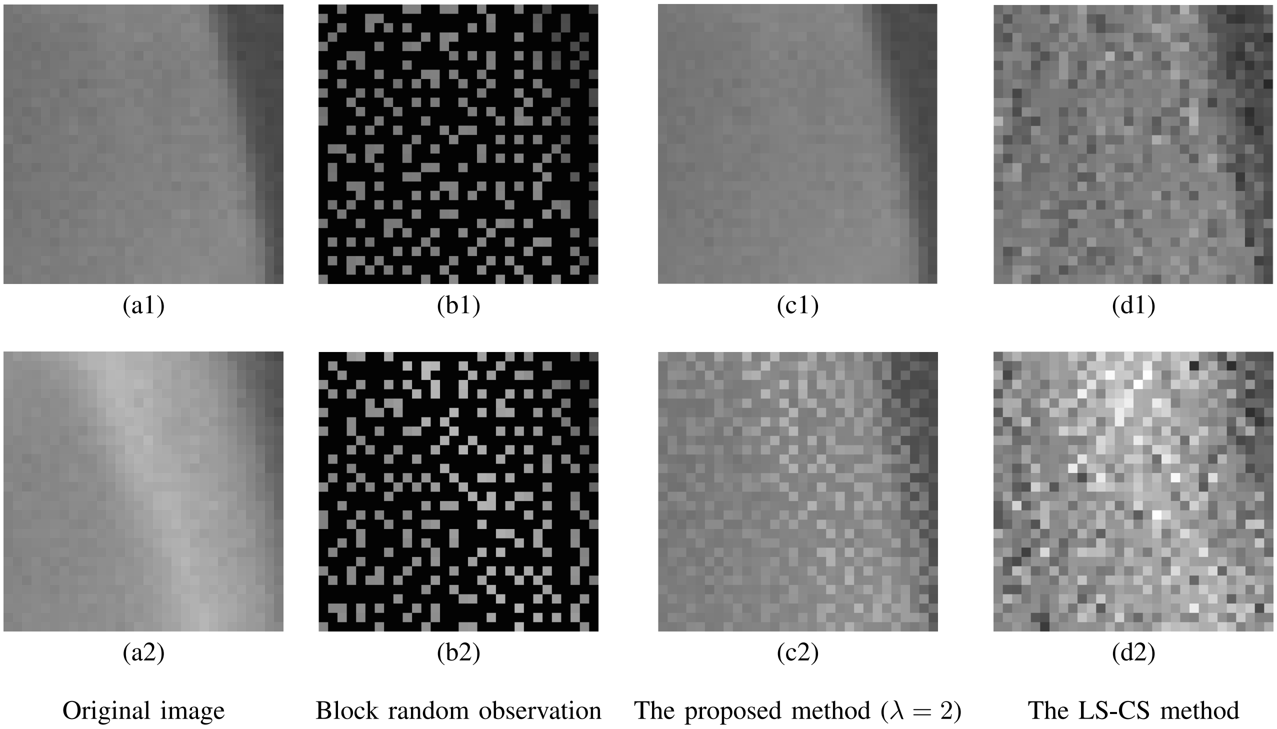


Fig. 6. Reconstructed images by The proposed method and The LS-CS method

it can be seen that the random observations with and without block segmentation amount to almost a similar result. Figure (b) shows the PSNRs for the sequence of "tennis". In this case, the results with the reference to the previous frame are mostly better than those without the reference.

Next, we compare the proposed method to the LS-CS method in [8]. We run the method by using the program provided in [13]. Because of the CVX toolbox used in the program, it was not capable of processing image sizes of 256×256 pixels. Due to this we cropped the original image of "water balloon" to 30×30 pixels. These results are shown in Fig 6. The original images are in the column (a) in the figure, while 25% random observations with block segmentation are in column (b). The reconstructed results by the proposed method and the LS-CS methods are in columns (c) and (d), respectively. The image (a1) was reconstructed with high quality by the proposed method even with this small size of image. This is because the motion was not much in this frame, while for the image (a2), the water surface is moving, so that the reconstructed image is noisy. The change of the PSNR [dB] of these methods are shown in Fig. 7, where the red and black lines indicate the value of the proposed and the LS-CS methods, respectively. The maximum and minimum for the proposed method were 43.91dB and 16.56dB, respectively with the average 31.52dB, while the maximum and the minimum for the LS-CS method were 34.85dB and 6.44dB, respectively with the average 14.08dB. The average of the proposed method is higher than that of the LS-CS method by 17.4dB. It takes 0.1 second to compute a single frame by the proposed method, while 100 seconds to do the same by the LS-CS method.

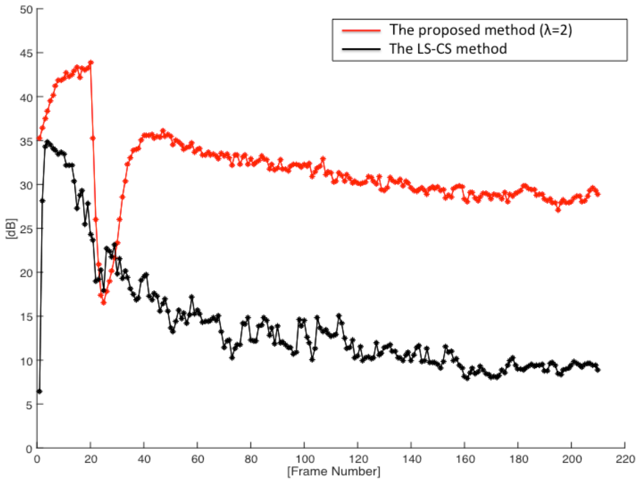


Fig. 7. PSNR [dB] of the reconstructed images by the proposed and the LS-CS methods.

V. Conclusion

To suppress the principle problem of high speed cameras, the number of pixels reduces when the number of frames per second (FPS) increases, the present paper proposed an optical setup that randomly selects some percent of pixels in an image and developed an algorithm that reconstructs the entire image from the selected partial pixels. In this algorithm, sparsity not only within each frame but also induced from the similarity between adjacent frames was exploited. Based on these two types of sparsity, a cost function for image reconstruction was defined using ℓ_1 norms. Since this function is convex, the optimal solution was efficiently found by using the Douglas-Rachford Splitting method, one of the convex optimization techniques. Simulation results showed that the proposed method is capable of recovery of high-quality 256×256 images from 25% of pixels that are randomly selected in a block-segmentation manner. We also found that, when there are intensive changes between adjacent frames, the reference to the adjacent frame should be reduced. To automatically control such reference is one of our future tasks.

Acknowledgements

The authors wish to thank nac Image Technology for providing the high speed camera systems. The authors also thank Mr. S. Ushifusa for valuable discussions.

References

[1] nac Image Technology, HX-3,"http://www.nacinc.com/ products/ memrecam-high-speed-digital-cameras/hx-3/."

[2] iPhone 6 plus, "https://www.apple.com/iphone-6/."

[3] GoPro Hero 4 Black, "http://shop.gopro.com/hero4/hero4-black/chdhx-401.html."

[4] R. Baraniuk, "Compressive sensing [lecture notes]," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, July 2007.

[5] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk, "An architecture for compressive imaging," in *2006 IEEE International Conference on Image Processing*, 2006, pp. 1273–1276.

[6] L. Kang and C. Lu, "Distributed compressive video sensing," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009)*, 2009, pp. 1169–1172.

[7] N. Vaswani, "Kalman filtered compressed sensing," in *15th IEEE International Conference on Image Processing (ICIP 2008)*, 2008, pp. 893–896.

[8] ——, "LS-CS-Residual (LS-CS): Compressive sensing on least squares residual," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4108–4120, 2010.

[9] H. Chen, L. Kang, and C. Lu, "Dictionary learning-based distributed compressive video sensing," in *Picture Coding Symposium (PCS)*, 2010, pp. 210–213.

[10] S. Chan, R. Khoshabeh, K. Gibson, P. Gill, and T. Nguyen, "An augmented Lagrangian method for total variation video restoration," *IEEE Transactions on Image Processing*, vol. 20, no. 11, pp. 3097–3111, 2011.

[11] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[12] Y. Song, Z. Zhu, Y. Lu, Q. Liu, and J. Zhao, "Reconstruction of magnetic resonance imaging by three-dimensional dual-dictionary learning," *Magnetic Resonance in Medicine*, vol. 71, no. 3, pp. 1285–1298, 2014.

[13] P.L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing, ''in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H.H. Bauschke, R.S. Burachik, P.L. Combettes, V. Elser, D.R. Luke, and H. Wolkowicz, Editors, pp. 185-212. Springer, New York, 2011.

[14] E. Candes and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, March 2008.

[15] SequentialCS, "http://www.ece.iastate.edu/˜namrata/research/sequentialcs.html."