

SEQUENTIAL IMAGE COMPLETION FOR HIGH-SPEED LARGE-PIXEL NUMBER SENSING

Akira Hirabayashi^{*}[†]Naoki Nogami^{*}Takashi Ijiri^{*}Laurent Condat^{*}

* Ritsumeikan University
 College Info. Science & Eng.
 Kusatsu, Shiga 525-8577, Japan

* University of Grenoble Alpes
 GIPSA-lab
 F38000 Grenoble, France

ABSTRACT

We propose an algorithm that enhances the number of pixels for high-speed camera imaging to suppress its main problem. That is, the number of pixels reduces when the number of frames per second (fps) increases. To this end, we suppose an optical setup that block-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, two types of sparsity are exploited. One is within each frame and the other is induced from the similarity between adjacent frames. The latter further means not only in the image domain but also in a sparsifying transformed domain. Since the cost function we define is convex, we can find the optimal solution using a convex optimization technique with small computational cost. Simulation results show that the proposed method outperforms the standard approach for image completion by the nuclear norm minimization.

Index Terms— High-speed camera, sparsity, compressed sensing, image completion, convex optimization

1. INTRODUCTION

High speed cameras, which are capable of capturing images at more than one hundred frames per second (fps), have been used in many scenes, such as engineering measurements, especially in the automotive industry, sports training, or entertainment. High spec products can capture 4.91 mega ($2,560 \times 1,920$) pixel images at two thousand fps [1]. A lower spec product can still capture 436 kilo ($1,136 \times 384$) pixel images at 960 fps [2].

The main issue of high speed cameras is the decrease of pixels when fps increases. For example, the second one of the abovementioned cameras can capture 1,871 kilo ($1,824 \times 1,026$) pixels at 240 fps and 948 kilo ($1,676 \times 566$) pixels at 480 fps. The reason of this phenomenon is that time for storing pixel values is proportional to the number of image pixels while the increase of fps number decreases the time for

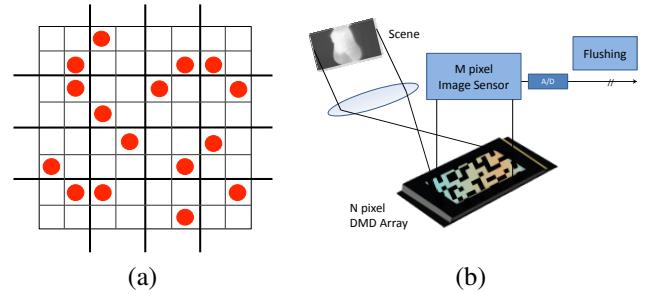


Fig. 1. (a) Block random selection and (b) its supposed optical setup. Single pixel (shown by red) is randomly selected from each 2×2 block.

storing. We aim to keep the pixel number as many as possible even when fps is a high value. Our idea is that a camera captures block-randomly selected partial pixels only. That is, an image is divided into several-pixel blocks and one pixel out of the block is randomly selected, as shown in Fig. 1 (a). We suppose to conduct this random selection by the optical setup shown in Fig. 1 (b), even though we have not implemented it yet. This is an extended version of the single pixel camera proposed in [3]. Our goal is to recover the sequence of entire images from that of the block-randomly selected partial pixel images with given random selection patterns.

There are lots of research relevant to this sequential image recovery problem [4–9]. Unfortunately, any of them hardly fits to the high-speed camera imaging setup. Thus, the present authors proposed a method that reconstructs images by minimizing the sum of ℓ_1 -norms of sparsifying transform's coefficients and the difference of adjacent frames under the observation constraint [10]. Although the method effectively reconstructs image sequences with mild motions, it could be defeated by a method without reference to the previous frame when an image sequence contains lots of quick moving objects. Thus, we propose a method that can recover a high-quality images even for such image sequences. Our idea here is to use the ℓ_1 -norm of the difference of the sparsifying transform's coefficients of adjacent frames in addition to that of the difference of images themselves. Since the problem ad-

[†] The author thanks for Ritsumeikan Program to Support General Research Activities. The author also wish to thank nac Image Technology for providing the high speed camera systems.

dressed here is an image completion problem or a matrix completion problem [11–13], we can apply the standard solution to the problems, i.e., the nuclear norm minimization. We show by simulations that the proposed method outperforms this standard method as well as our previously proposed method in [10].

The rest of the present paper is organized as follows. In section 2, we formulate the image reconstruction problem and define a cost function for the image reconstruction. Section 3 proposes a fast image reconstruction algorithm based on a convex optimization method. Simulation results are also shown in this section. Section 4 concludes the paper.

2. SEQUENTIAL IMAGE COMPLETION PROBLEM

Suppose that a fixed high speed camera captures a scene at a high frame rate and a sequence of images $\mathbf{x}_r \in \mathbb{R}^N$ ($r = 1, \dots, R$) is obtained. Pixels in the r th image \mathbf{x}_r is randomly selected so that M pixels are remaining ($M < N$). Let A_r and $\mathbf{y}_r \in \mathbb{R}^M$ be a random selection matrix (M rows are randomly selected from the identity matrix I of the corresponding size) and a vector consisting of the selected pixels. Then, it holds that

$$\mathbf{y}_r = A_r \mathbf{x}_r, \quad (r = 1, 2, \dots, R). \quad (1)$$

Note that the random selection pattern in A_r is generated at every frame, not fixed. Our goal is to estimate the image sequence $\{\mathbf{x}_r\}_{r=1,\dots,R}$ from $\{A_r\}_{r=1,\dots,R}$ and $\{\mathbf{y}_r\}_{r=1,\dots,R}$. Because of the high frame rate and thus high shutter speed, high-speed camera images are said to be noisy. Nevertheless, we did not take noise into account because we aim here to reconstruct the sequence $\{\mathbf{x}_r\}_{r=1,\dots,R}$. For the same reason, we do not take blur into account either.

We use two sequence of images throughout the paper. One sequence, say T, is about a tennis playing scene captured outdoors by a high-speed camera, Optronis CR450x3, nac Image Technology [14] at 6,000 fps. The other sequence, say B, is about a water balloon bursting scene captured indoors by the same camera at the same fps. Both sequences consist of 210 frames of uncompressed 256×256 images. These sequences can be seen from the URL: <http://www.ms.is.ritsumei.ac.jp/HSC>. Roughly speaking, the sequence T contains a small moving object (a ball and racket) with a still background, while the images in the sequence B are mostly filled with moving objects (splashing water).

Figure 2 shows the ℓ_1 -norms of differences of adjacent frame images (blue curves) and their DCT coefficients (red curves) in terms of frame number. We can see that, the ℓ_1 -norm of the DCT coefficient difference is mostly smaller than or equal to that of the image difference for the sequence T, while it is true for the frames between 20 and 45, but not for the frames other than that for the sequence B. We can understand this tendency as follows: when the moving area in the image is small, the difference of the DCT coefficient is small.

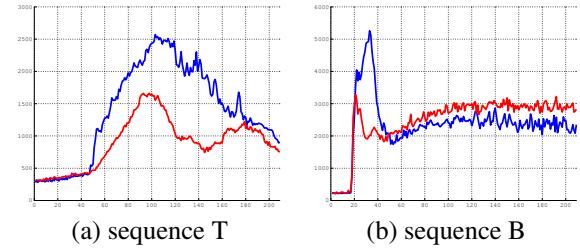


Fig. 2. ℓ_1 -norms of difference of adjacent frames and its DCT coefficients in terms of frame number.

On the other hand, when the moving area is large, the difference of the image itself is small compared to the difference of the DCT coefficients. Hence, by combining these two norms appropriately, we can achieve high-quality image recovery. Therefore, we propose to use the following cost function to recover image sequences:

$$\hat{\mathbf{x}}_r = \arg \min_{A_r \mathbf{x} = \mathbf{y}_r} \{ \|L\mathbf{x}\|_1 + \lambda_1 \|\mathbf{x} - \hat{\mathbf{x}}_{r-1}\|_1 + \lambda_2 \|L_2 \mathbf{x} - L_2 \hat{\mathbf{x}}_{r-1}\|_1 \}, \quad (2)$$

The three terms in the cost function express the following three priors. The first term means that each captured image is supposed to be sparse in the sparsifying transformed domain. The second term means the difference image between adjacent frames is sparse. The third term means the difference between the sparsifying transform's coefficients of adjacent frames is also sparse. The parameters λ_1 and λ_2 control the balance between the three terms. By setting $\lambda_2 = 0$, the cost function amounts to that used in [10]. We will show by simulations that the algorithm with the new cost function (2) outperforms that with the cost function in [10].

3. IMAGE RECOVERY ALGORITHM

The problem (2) can be efficiently solved using a convex optimization algorithm. Let S_r be a set of \mathbf{x} satisfying $A_r \mathbf{x} = \mathbf{y}_r$. This is convex. Then, (2) is equivalent to

$$\hat{\mathbf{x}}_r = \arg \min_{\mathbf{x} \in \mathbb{R}^N} \{ \|L\mathbf{x}\|_1 + \lambda_1 \|\mathbf{x} - \hat{\mathbf{x}}_{r-1}\|_1 + \lambda_2 \|L_2 \mathbf{x} - \hat{\mathbf{u}}_{r-1}\|_1 + \iota_{S_r}(\mathbf{x}) \}, \quad (3)$$

where $\hat{\mathbf{u}}_{r-1} = L_2 \hat{\mathbf{x}}_{r-1}$ and $\iota_{S_r}(\mathbf{x})$ is the indicator function that takes value 0 if $\mathbf{x} \in S_r$, $+\infty$ else. Now, our problem becomes the minimization of the sum of the four non-smooth terms including two composite terms. As is well known, the proximity operator of the ℓ_1 -norm term $\theta \|\mathbf{u}\|_1$ ($\theta > 0$) is $\text{prox}_{\theta \|\cdot\|_1}(\mathbf{u}) = (\text{softThresh}(u_n, \theta)) \in \mathbb{R}^N$, where

$$\text{softThresh}(u, \theta) = \begin{cases} u - \theta & (u \geq \theta), \\ u + \theta & (u \leq -\theta), \\ 0 & (-\theta < u < \theta). \end{cases}$$

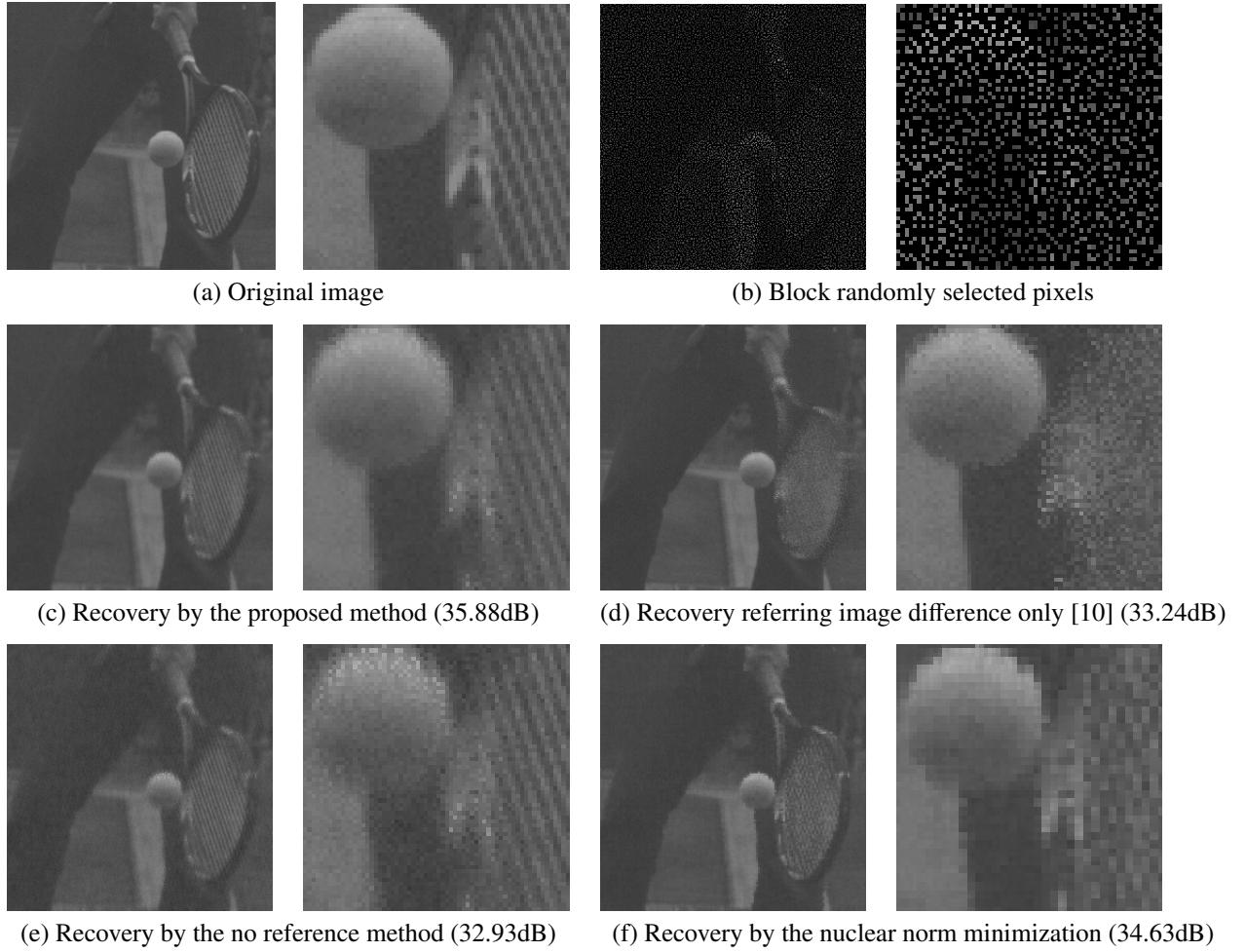


Fig. 3. Reconstructed images for an image in the sequence T.

Further, the proximity operator of $\theta\|\mathbf{x} - \hat{\mathbf{x}}\|_1$ is given by $\text{prox}_{\theta\|\cdot-\hat{\mathbf{x}}\|_1}(\mathbf{x}) = (\text{softThresh}(x_n - \hat{x}_n, \theta) + \hat{x}_n) \in \mathbb{R}^N$. Obviously, the proximity operator of the indicator function is the projection onto S_r . Since it was shown that the proximity operators for all functions in (3) can be computed in the closed forms, we can solve the problem (2) by using the simultaneous direction method of multipliers (SDMM) [15], as

Algorithm 1: Image recovery for r th frame

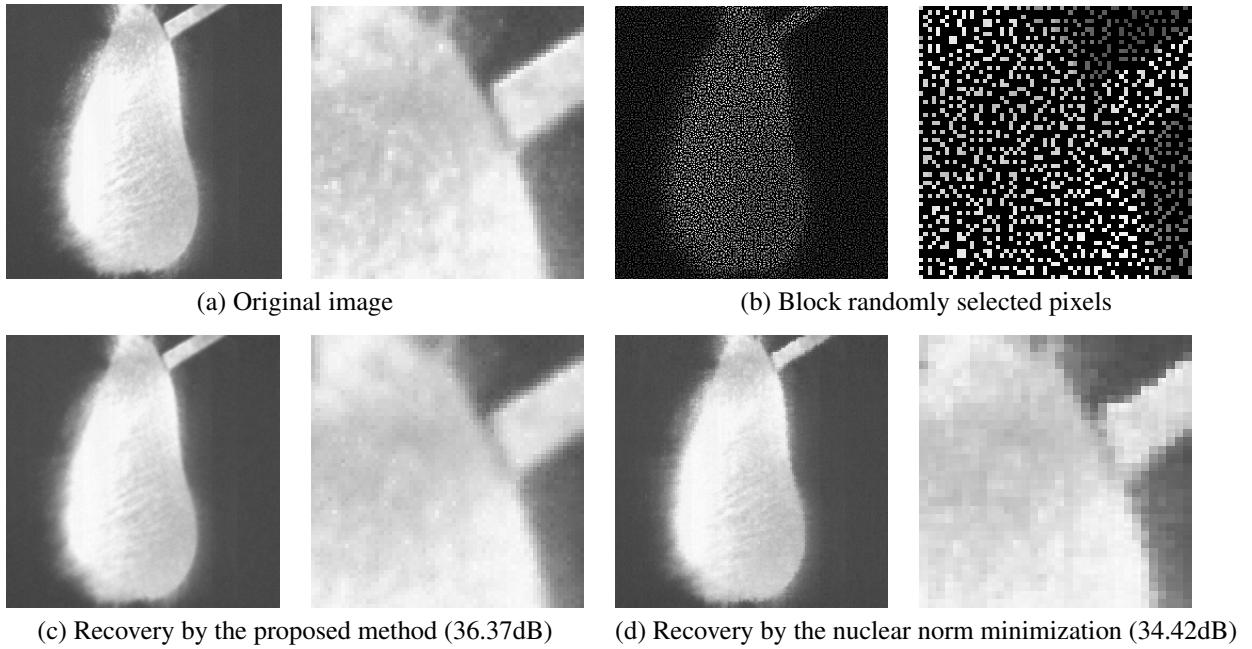
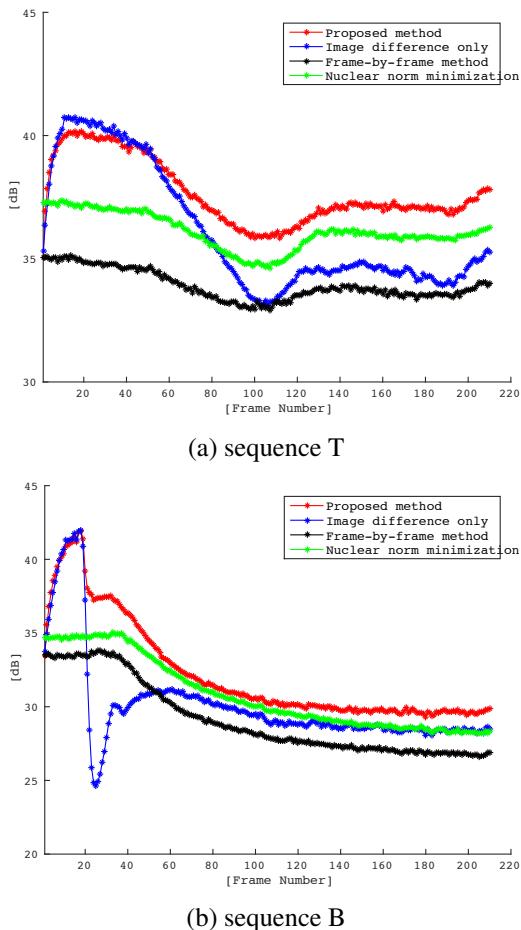
Input: $\mathbf{y}_r, A_r, \hat{\mathbf{x}}_{r-1}$
Output: $\hat{\mathbf{x}}_r$

1. Set $\gamma > 0$
2. Set $\text{proj}_{S_r}(\mathbf{0})$ for \mathbf{x} and $\mathbf{0}$ s for $\mathbf{u}_0, \dots, \mathbf{u}_3$.
3. Repeat the following operations:
for $i = 0 \sim 3$
 $s_i \leftarrow L_i \mathbf{x},$
 $\mathbf{v}_i \leftarrow \text{prox}_{g_i}(s_i + \mathbf{u}_i),$
 $\mathbf{u}_i \leftarrow \mathbf{u}_i + s_i - \mathbf{v}_i,$
 $\mathbf{x} \leftarrow Q^{-1} \sum_{i=0}^3 (\mathbf{v}_i - \mathbf{u}_i),$
until a stopping condition is met.

where $\mathbf{0}$ is a zero vector, $L_0 = L$, $L_1 = L_3 = I$, $g_0(\mathbf{u}) = \|\mathbf{u}\|_1$, $g_1(\mathbf{x}) = \lambda_1\|\mathbf{x} - \hat{\mathbf{x}}_{r-1}\|_1$, $g_2(\mathbf{u}) = \lambda_2\|\mathbf{u} - \hat{\mathbf{u}}_{r-1}\|_1$, and $g_3(\mathbf{x}) = \iota_{S_r}(\mathbf{x})$. We also used $Q = \sum_{i=0}^3 L_i^\top L_i$, which we assumed invertible. In the following simulations, both L and L_2 are two-dimensional discrete cosine transform (DCT). Then, Q reduces to $4I$ and Q^{-1} application in the update of \mathbf{x} can be simplified to just the division by four. Application of L and L_2 to \mathbf{x} can be computed by $O(N \log N)$. Under these conditions, Algorithm 1 converges quickly, say after twenty time iterations, which is the stopping condition in the simulations.

We applied Algorithm 1 to 25% block-randomly selected pixel images from the sequences T and B. The balancing parameters were set as $\lambda_1 = 0.5$ and $\lambda_2 = 1$.¹ For comparison, these sequences were reconstructed by the method in [10], the proposed method with $\lambda_1 = \lambda_2 = 0$ (no reference), and the nuclear norm minimization as well. The reconstructed sequence can be seen at the same URL, as well as the matlab codes. Figure 3 shows one frame in the reconstructed sequence for

¹These values were empirically determined, but λ_1/λ_2 is similar to the inverse ratio of the peaks in Figure 2.

**Fig. 4.** Reconstructed images for an image in the sequence B.**Fig. 5.** PSNRs of reconstructed images.

the sequence T. Figure (a) shows the target image. Figure (b) shows the block-randomly selected pixels. Figures (c), (d), (e), and (f) show the reconstructed images by the proposed method, the method in [10], the no reference method, and the nuclear norm minimization, respectively. PSNRs computed by $20 \log_{10} \frac{255 \sqrt{N}}{\|\hat{x}_r - x_r\|_2}$ [dB] are also shown in captions. We can see that the proposed algorithm outperforms the other methods both subjectively and objectively. Figure 4 shows one frame in the reconstructed sequence for the sequence B. Figures (a) and (b) show the target image and the block-randomly selected pixels, respectively. Figures (c) and (d) show the reconstructed images by the proposed method and the nuclear norm minimization, respectively. PSNRs with respect to frame number are shown in Figure 5, in which the red, blue, black, and green curves indicate PSNRs by the proposed method, the method in [10], the no reference method, and the nuclear norm minimization, respectively. We can see that, even though the peak value by the proposed method is lower than that by the method in [10] (blue), the proposed method (red) keeps high PSNR values for all frames.

4. CONCLUSION

We proposed a sequential image completion algorithm that enhances the number of pixels for high-speed camera imaging. The proposed algorithm exploited sparsity both within each frame and between frames. Simulation results showed that the proposed method outperforms not only the present authors' previously proposed method but also the standard nuclear norm minimization.

REFERENCES

- [1] nac Image Technology high speed camera HX-3. [Online]. Available: <http://www.nacinc.com/products/memrecam-high-speed-digital-cameras/hx-3/>
- [2] SONY high speed camera RX10II. [Online]. Available: <http://www.sony.jp/cyber-shot/store/special/dsc-rx10m2/>
- [3] R. Baraniuk, "Compressive sensing [lecture notes]," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, July 2007.
- [4] 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.
- [5] 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.
- [6] N. Vaswani, "Kalman filtered compressed sensing," in *15th IEEE International Conference on Image Processing (ICIP 2008)*, 2008, pp. 893–896.
- [7] ———, "LS-CS-Residual (LS-CS): Compressive sensing on least squares residual," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4108–4120, 2010.
- [8] H. Chen, L. Kang, and C. Lu, "Dictionary learning-based distributed compressive video sensing," in *Picture Coding Symposium (PCS)*, 2010, pp. 210–213.
- [9] 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.
- [10] A. Hirabayashi, N. Nogami, J. White, and L. Condat, "Pixel enlargement in high-speed camera image acquisition based on 3D sparse representations," in *Proceedings of 2015 IEEE International Workshop on Signal Processing Systems*, Hangzhou, 2015.
- [11] E. Candes and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, June 2010.
- [12] R. Ma, N. Barzigar, A. Roozgard, and S. Cheng, "Decomposition approach for low-rank matrix completion and its applications," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1671–1683, April 2014.
- [13] F. Cao, M. Cai, and Y. Tan, "Image interpolation via low-rank matrix completion and recovery," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 8, pp. 1261–1270, Aug. 2015.
- [14] nac Image Technology high speed camera CR450x3. [Online]. Available: <http://green.atengineer.com/pr/nacinc/20101021001.html>
- [15] H. Bauschke, R. Burachik, P. Combettes, V. Elser, D. Luke, and H. Wolkowicz, Eds., *Fixed-point algorithms for inverse problems in science and engineering*. New York: Springer, 2011.