

Journal of Electronic Imaging

JElectronicImaging.org

Nonconvex compressive video sensing

Liangliang Chen
Ming Yan
Chunqi Qian
Ning Xi
Zhanxin Zhou
Yongliang Yang
Bo Song
Lixin Dong



Liangliang Chen, Ming Yan, Chunqi Qian, Ning Xi, Zhanxin Zhou, Yongliang Yang, Bo Song, Lixin Dong, "Nonconvex compressive video sensing," *J. Electron. Imaging* **25**(6), 063003 (2016), doi: 10.1117/1.JEI.25.6.063003.

Nonconvex compressive video sensing

Liangliang Chen,^{a,*} Ming Yan,^{b,c} Chunqi Qian,^d Ning Xi,^e Zhanxin Zhou,^a Yongliang Yang,^a Bo Song,^a and Lixin Dong^a

^aMichigan State University, Department of Electrical and Computer Engineering, East Lansing, Michigan 48824, United States

^bMichigan State University, Department of Computational Mathematics, Science and Engineering, East Lansing, Michigan 48824, United States

^cMichigan State University, Department of Mathematics, East Lansing, Michigan 48824, United States

^dMichigan State University, Department of Radiology, East Lansing, Michigan 48824, United States

^eThe University of Hong Kong, Department of Industrial and Manufacturing Systems Engineering, Hong Kong

Abstract. High-speed cameras explore more details than normal cameras in the time sequence, while the conventional video sampling suffers from the trade-off between temporal and spatial resolutions due to the sensor's physical limitation. Compressive sensing overcomes this obstacle by combining the sampling and compression procedures together. A single-pixel-based real-time video acquisition is proposed to record dynamic scenes, and a fast nonconvex algorithm for the nonconvex sorted ℓ_1 regularization is applied to reconstruct frame differences using few numbers of measurements. Then, an edge-detection-based denoising method is employed to reduce the error in the frame difference image. The experimental results show that the proposed algorithm together with the single-pixel imaging system makes compressive video cameras available. © 2016 SPIE and IS&T [DOI: 10.1117/1.JEI.25.6.063003]

Keywords: nonconvex optimization; compressive sensing; compressive video sampling; spatial and temporal resolutions.

Paper 16288 received May 2, 2016; accepted for publication Oct. 20, 2016; published online Nov. 15, 2016.

1 Introduction

Video acquisition captures time-dependent natural scenes and brings real-time images directly to screens for immediate observation. It not only serves for the live television (TV) production, but also for security, military, and industrial operations including professional video cameras, camcorders, closed circuit TV, webcams, camera phones, and special camera systems. In traditional video acquisition, e.g., H.261, H.265, and MPEG series, the sampling and compression procedures are implemented in sequential order. The Nyquist–Shannon sampling theorem requires the sampling rate to be at least twice that of the signal frequency for guaranteed exact recovery. The compression procedure is implemented by video compression chipsets¹ or separate software.²

Although state-of-the-art video cameras can record most nature scenes, they do not work for very high-resolution images or high fps videos because the growth in data storage, communication, and processing is far behind the growth in data generation. In space exploration, an image of the shuttle discovery flight deck could be 2.74 gigapixels,³ and a bubble dynamics research needs a 500-fps video microscopy.⁴ More importantly, commercialized high-performance video cameras are extremely expensive, e.g., the price of a basic model with 7500 fps, one-megapixel resolution, and 12-bit color depth (FASTCAM SA5 from Photron) is around \$100,000.

The limitation comes from weak light irradiation and the readout bandwidth when capturing high-speed objects at a high resolution. As shown in Fig. 1 and Eq. (1), the reflected illumination is collected by sensor arrays in a limited space–time volume

$$\mathbf{J} = 10^{15} F^{-2} t I_{src} R q \Delta^2. \quad (1)$$

The number of electrons (\mathbf{J}) accumulated on each pixel is reversely proportional to the square of the ratio of the focal length to the aperture of the lens (F), but proportional to exposure time (t), incident illumination (I_{src}), scene reflectivity (R), quantum efficiency (q), and the pixel size (Δ^2).⁵ In video sensing, the exposure time (t) corresponds to the temporal resolution and the pixel size (Δ^2) is related to the spatial resolution. In other words, the temporal and spatial resolutions are mutual restraint in conventional video cameras due to the imaging sensors' requirement on the minimum number of accumulated electrons and the fixed number of total electrons. The spatial resolution will decrease when the temporal resolution increases. Another limitation is the sensor's readout speed. The readout timing includes an analog-to-digital conversion, clear charge from the parallel register, and shutter delay, e.g., a one-megapixel, 1000 fps, and 16-bit color camera will need a 4-GB/s readout circuit.

To obtain high-resolution images and high fps videos, the sampling rate has to be reduced, and compressive sensing technique can be applied. Compressive sensing⁶ allows combining both sampling and compression procedures together. This paradigm directly samples the signal in a compressed form such that the sampling rate can be significantly reduced. Compressive sensing has attracted extreme interest in imaging,⁷ geophysical data analysis,⁸ control and robotics,⁹ communication,¹⁰ and medical imaging processing.¹¹

Compressive sensing has been applied in compressive video sensing since 2006, when the single-pixel camera setup was first used for video sampling.¹² In this first approach, the three-dimensional (3-D) video was reconstructed with all the measurements together using 3-D wavelets as a sparse representation. This method cannot be used for real-time video streaming without incurring latency and

*Address all correspondence to: Liangliang Chen, E-mail: chenlia5@msu.edu

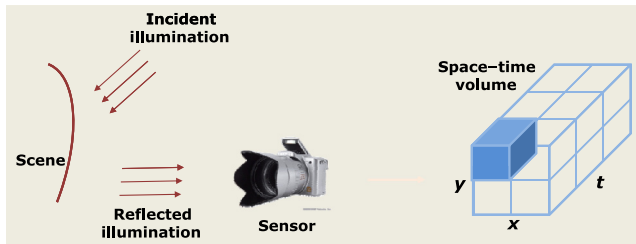


Fig. 1 Light illumination in single-lens reflex cameras.

delay because all the measurements have to be obtained before the reconstruction starts. Since then, in order to reconstruct the frames one by one for the purpose of real-time streaming, most approaches reconstruct or sample reference frames with more measurements and find the differences between two consecutive frames with fewer measurements. There are mainly two types of strategies: sampling the frame and sampling the difference between frames. In the first sampling method, in order to obtain a continuous video, motion estimation techniques are applied to recover frames from reference frames. For example, the evolution of dynamic textured scenes was modeled as a linear dynamical system.¹³ A multiframe motion estimation algorithm was proposed.¹⁴ The latest compressive video sensing research learned a linear mapping between video sequences and corresponding measured frames.¹⁵ In addition, the correlation between consecutive frames in the frequency domain¹⁶ and other transform domains¹⁷ was also used.

There are also several approaches in sampling the difference between two frames. For example, Stankovic et al.¹⁸ split the video frame into nonoverlapping blocks of equal size, and compressive sampling was performed on sparse blocks determined by predicting sparsities based on previous reference frames, which were sampled conventionally. The remaining blocks were sampled fully. It would be time-consuming to determine the sparse blocks because every block has to be tested. In addition, directly sampling the difference between two consecutive frames was employed¹⁹ to save the sampling time.

Though compressive sensing techniques are used in video sensing, most of the approaches use the convex ℓ_1 minimization to approximate the nonconvex ℓ_0 minimization, which is a nondeterministic polynomial-time (NP)-hard and difficult to solve. The compressive sensing theorem can reduce the number of measurements using the ℓ_1 minimization. However, with nonconvex regularizations, it can reduce the number of measurements and thus the sampling rate further so as to achieve real-time video capturing. Recently, there are many nonconvex regularizations proposed to obtain better performance than the ℓ_1 norm in compressive sensing.^{20,21,22}

In this paper, a single-pixel compressive video sensing framework based on the nonconvex sorted ℓ_1 regularization is proposed for fast and super resolution video. In this framework, we sample reference frames using the spatial sparsity (individual image sparsity) and the difference between two frames using the temporal sparsity. In Sec. 2, we first give a short review about compressive sensing and nonconvex solvers. Then, we propose our nonconvex compressive video sensing framework. The experimental results are depicted in Sec. 3.

2 Compressive Video Sensing

2.1 Compressive Sensing

The core of compressive sensing is recovering the sparse vector $\mathbf{x} \in \mathbb{R}^n$ from a small number of linear measurements $\mathbf{y} = \Phi\mathbf{x}$, where $\Phi \in \mathbb{R}^{m \times n}$ is the measurement matrix ($m \ll n$). There are many solutions for the underdetermined linear system if \mathbf{y} is in the range of Φ , and we are interested in finding the sparsest one among all the solutions. However, finding the sparsest solution is NP-hard. Therefore, instead of solving the NP-hard problem, people are looking into alternative approaches. Convex approaches are of great interest because there are lots of algorithms for solving these convex problems and it is easy to analyze the solutions of the convex problems. If \mathbf{x} is sparse and Φ satisfies some conditions such as the null space property,²³ the incoherence condition,²⁴ and the restricted isometry property,²⁵ the following problem is equivalent for finding the sparsest solution:

$$\tilde{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to } \Phi\mathbf{x} = \mathbf{y}. \quad (2)$$

When there is noise in the measurements, i.e., $\Phi\mathbf{x} + \mathbf{n} = \mathbf{y}$ with \mathbf{n} being the white Gaussian noise, we solve the following problem instead:

$$\tilde{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 + \frac{\lambda}{2} \|\Phi\mathbf{x} - \mathbf{y}\|^2, \quad (3)$$

where λ is a parameter for balancing the data fitting term and the regularization term. In order to solve these convex ℓ_1 problems, many algorithms are proposed.^{26,27}

Although the ℓ_1 minimization is fully understood and stable with theoretical guarantee, the number of required measurements is still high, and the performance is not good in many applications with a small number of measurements. For example, radiologists want to reduce more projections and thus radiation than that required for ℓ_1 minimization in computed tomography. For the difference between two frames in a video, we want to decrease the number of measurements further such that it can realize higher fps videos than current cameras can produce. In order to recover signals from even fewer measurements, nonconvex regularizations are applied, and a short review will be given in Sec. 2.2.

2.2 Nonconvex Optimization Problems for Compressive Sensing

In this section, we review several nonconvex regularizations for compressive sensing and their corresponding algorithms. Denote $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$, the truth sparse signal as \mathbf{x}_0 , and \mathbf{x}^l as the l 'th iteration.

The ℓ_p ($0 \leq p \leq 1$) term is commonly used,²⁸ and it has ℓ_0 and ℓ_1 as special cases. Because of the nonconvexity, it recovers sparse signals with even fewer measurements than the convex counterpart, ℓ_1 . To solve the nonconvex problems, there are several approaches. We describe three of them on both the noise-free and noisy cases. First, two reweighted algorithms for the following noise-free case are presented:

$$\tilde{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_p \quad \text{subject to } \Phi \mathbf{x} = \mathbf{y}. \quad (4)$$

The iteratively reweighted ℓ_1 minimization (IRL1)²⁰ replaces the ℓ_p term using a weighted ℓ_1 term with the weights depending on the previous iteration. The iteration is expressed as

$$\mathbf{x}^{l+1} = \arg \min_{\mathbf{x}} \sum_{i=1}^n \frac{1}{(|x_i^l| + \varepsilon)^{1-p}} |x_i| \quad \text{subject to } \Phi \mathbf{x} = \mathbf{y}. \quad (5)$$

For every iteration, a weighted ℓ_1 minimization problem has to be solved and iterative algorithms are applied.

Similarly, the iteratively reweighted least squares^{21,22} replace the ℓ_p term using a weighted least squares term with the weights depending on the previous iteration. The iteration is expressed as

$$\mathbf{x}^{l+1} = \arg \min_{\mathbf{x}} \sum_{i=1}^n \frac{1}{(|x_i^l|^2 + \varepsilon)^{1-p/2}} |x_i|^2 \quad \text{subject to } \Phi \mathbf{x} = \mathbf{y}. \quad (6)$$

In this case, there is an analytical solution for the weighted ℓ_2 minimization problem, since it is equivalent to a least squares problem.

Except for these two reweighted algorithms for solving ℓ_p minimization problems, some algorithms for solving convex optimization problems are applied to solve nonconvex problems with general nonconvex regularizations.²⁹ One example is the forward-backward iteration. In each forward-backward iteration, for solving

$$\tilde{\mathbf{x}} = \arg \min_{\mathbf{x}} r(\mathbf{x}) + \frac{\lambda}{2} \|\Phi \mathbf{x} - \mathbf{y}\|^2, \quad (7)$$

where $r(\mathbf{x})$ is a nonconvex regularization term including $\|\mathbf{x}\|_p$ and the following mentioned nonconvex sorted ℓ_1 as special cases, a proximal mapping of the nonconvex regularization term follows a gradient descent on the data fidelity term, i.e.,

$$\mathbf{x}^{l+1} = \arg \min_{\mathbf{x}} \tau r(\mathbf{x}) + \frac{1}{2} \|\mathbf{x} - (\mathbf{x}^l - \tau \lambda \Phi^T (\Phi \mathbf{x}^l - \mathbf{y}))\|^2. \quad (8)$$

However, for ℓ_p minimization, there are only analytical solutions when $p = 0, 1/2, 2/3,$ and 1 .³⁰

The success of ℓ_p minimization and both iterative algorithms for solving ℓ_p minimization problems depicts that it is better to assign small weights for components with large absolute values and large weights for zero components and components with small absolute values. A nonconvex sorted ℓ_1 that assigns weights based on the ranking of absolute values was developed by Huang et al.³¹ Let the coefficients $\{\omega_i\}_{i=1}^n$ be a nondecreasing sequence of nonnegative real numbers, i.e., $0 \leq \omega_1 \leq \dots \leq \omega_n \neq 0$. The nonconvex sorted ℓ_1 regularization is defined as

$$r_{\omega}(x_1, x_2, \dots, x_n) = \omega_1 |x_{[1]}| + \omega_2 |x_{[2]}| + \dots + \omega_n |x_{[n]}|, \quad (9)$$

where $|x_{[1]}| \geq \dots \geq |x_{[n]}|$ are the absolute values of the components in \mathbf{x} ranked in decreasing order. Two special cases of nonconvex sorted ℓ_1 are 2-level ℓ_1 with $w_1 = w_2 = \dots = w_k = a_1 < 1 = w_{k+1} = \dots = w_n$ and iterative support detection (ISD) with $w_1 = w_2 = \dots = w_k = 0 < 1 = w_{k+1} = \dots = w_n$. In addition, Huang et al. suggested a way for adaptively changing the weights during the iteration instead of having a fixed set of weights for better performance. The proposed update rule is

$$\omega_i^l = \begin{cases} 1, & \text{if } i > K^l, \\ e^{-r(K^l-i)/K^l}, & \text{otherwise,} \end{cases} \quad (10)$$

where r controls the rate of decreasing ω_i from 1 to 0 and K^l is the smallest i such that $|x_{[i+1]}^l - x_{[i]}^l| > \|x^l\|_{\infty} / \beta$ with some positive β .³²

2.3 Video Compressive Sampling

A video can be considered as a series of images, as shown in Fig. 2 (left), where the coordinate space (x, y, t) consists both the spatial domain (x, y) and the temporal domain (t) . Each frame could be realized as a static natural image that is redundant because natural images are intrinsically sparse in a specific domain.^{24,33} Another redundancy happens between similar frames in the temporal domain. As shown in Fig. 3, more than 85% of the pixels have no significant changes. Therefore, difference coding³⁴ in MPEG and H.265 series reuses existing frames and updates only the pixels with significant changes.

As discussed in Sec. 1, the objective of compressive video sensing is to combine both compression and sampling procedures to achieve the signal compression in hardware. In our proposed compressive video sensing, there are two types of image frames: intraframes (I-frames in H.264 or reference frames) and interframes (P-frames in H.264), shown in Fig. 4. The compressive sampling is applied on both I-frames and P-frames, where P-frames are reconstructed by the difference between P-frames and their previous frames.

Since I-frames are considered as static images and the image compressive sampling has already been studied for single-pixel cameras,^{7,35} a total variation algorithm³⁶ is applied to recover intraframes from the I-frame sampling. For the P-frames, because the difference between similar frames is sparse, a nonconvex regularization is adopted to reduce the number of samples and thus increase the compression ratio. We compare the performance of four different

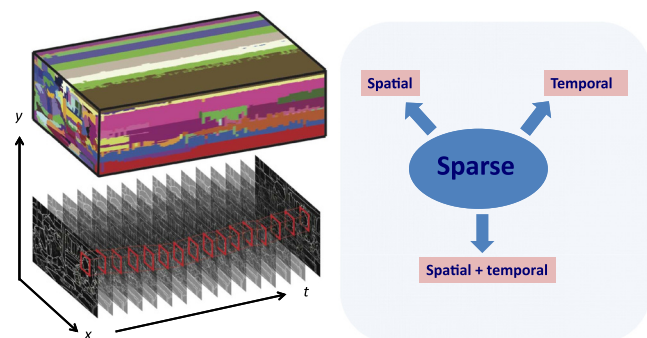


Fig. 2 Sparsity in videos.

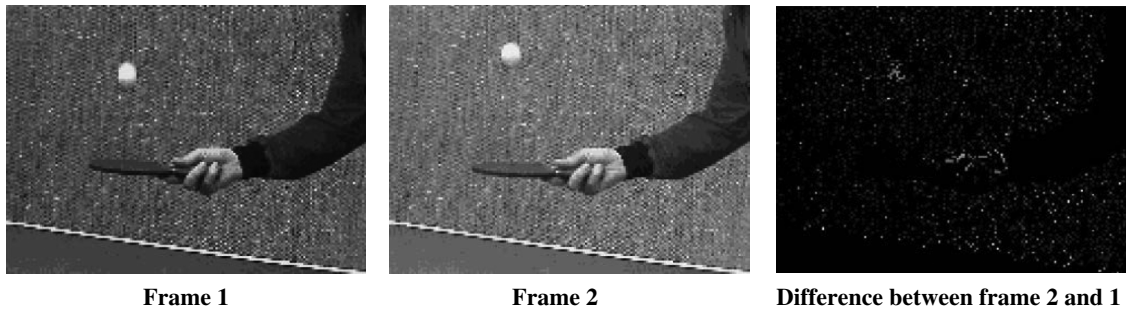


Fig. 3 Difference between two consecutive frames. The difference between the left and middle images is shown on the right. We can see that most pixels are unchanged in these two figures.

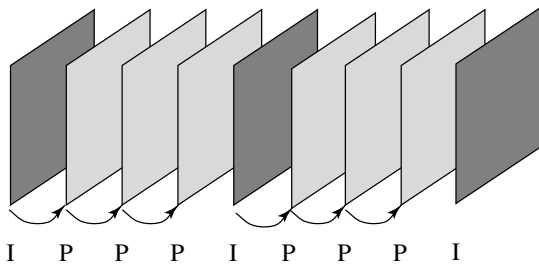


Fig. 4 A frame sequence with one I-frame (reference frame) and three P-frames.

nonconvex regularizations numerically and choose the best in the experiment. The four regularizations are: ℓ_p with IRL1, ISD, 2-level, and the nonconvex sorted ℓ_1 (m -level). In IRL1, the weights are updated by

$$\omega_i^l = \frac{1}{|x_i| + \max(0.5^{l-1}, 0.8^8)} \quad (11)$$

For 2-level, we choose $a = 0.6$. For m -level, we choose $\beta = 7$ and $r = 0.1$.

We compare the runtimes, root-mean-square error (RMSE), and the peak signal-to-noise ratio (PSNR) for these four algorithms on the difference between two consecutive frames (64×64) in Fig. 5. The difference between the left and the middle images in Fig. 5 is shown on the right. We choose the measurement matrices to be randomized Bernoulli matrices with ± 1 entries. The sampling rate (the number of measurements/the number of pixels) is changed

from 6% to 35%. The comparison result is shown in Fig. 6, where the x -axis represents the sampling rate. When the number of measurements is small, nonconvex algorithms are unstable because they can easily be trapped at stationary points and the strategy for adaptively updating weights may not work so well. Overall, m -level is the most efficient and effective algorithm among all these four algorithms. Therefore, we choose m -level in our experiments in Sec. 3.

Though nonconvex algorithms are able to recover sparse signals accurately from a small number of linear measurements, there is still error due to the hardware noise and the modeling error. For example, there is noise in the measurements and the algorithms cannot recover the sparse signals exactly. In Fig. 7, we show the exact difference image between two frames on the left and compare it with that recovered using the nonconvex sorted ℓ_1 on the middle. It is noticed that there are many isolated pixels with small non-zero values in the recovered difference image, and these pixels are supposed to have zero values. In order to improve this, we develop a simple and effective method to remove these pixels and update only the pixels in the areas with significant changes.

We apply the Sobel operator with a pair of 3×3 convolution masks on the recovered difference image to find the edges since the Sobel kernels compute the gradient with smoothing in both the horizontal and vertical directions. Then a threshold is selected to obtain a binary mask that indicates the pixels with large gradient values. However, it does not delineate the outline of the changing area of interest. Then the binary gradient mask is dilated using the vertical structuring element followed by the horizontal structuring

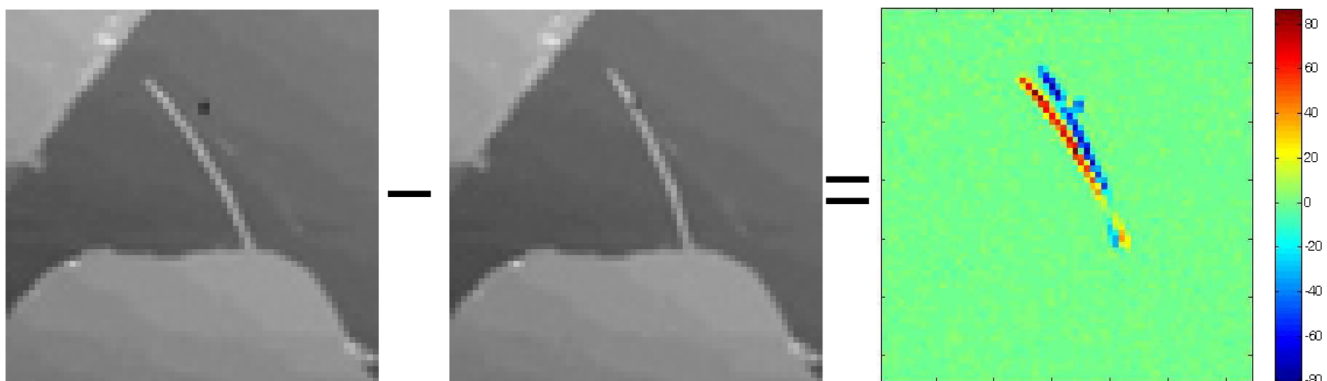


Fig. 5 Difference image of two consecutive frames; the difference between the left and middle images is shown in the right.

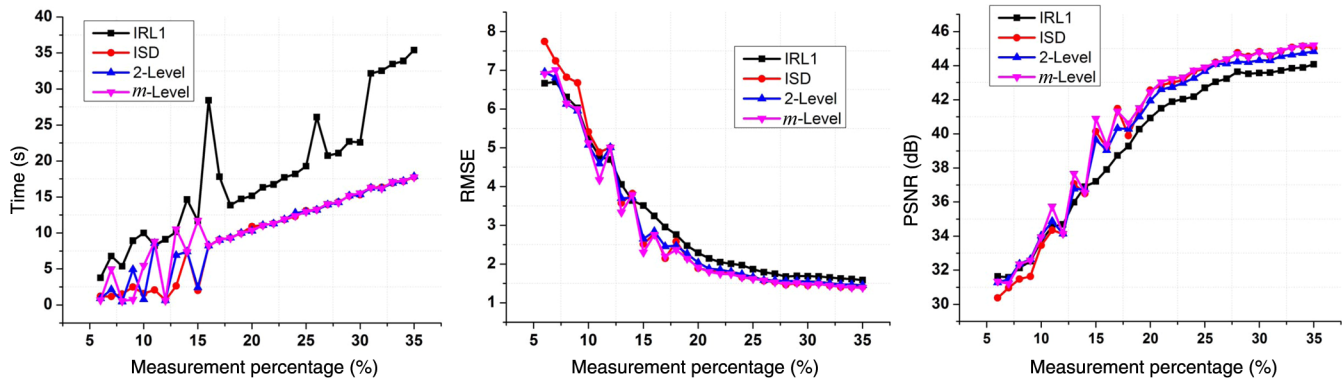


Fig. 6 Comparison of four nonconvex algorithms for signal recovery at different sampling rates. Overall, the *m*-level is the most efficient and effective algorithm.

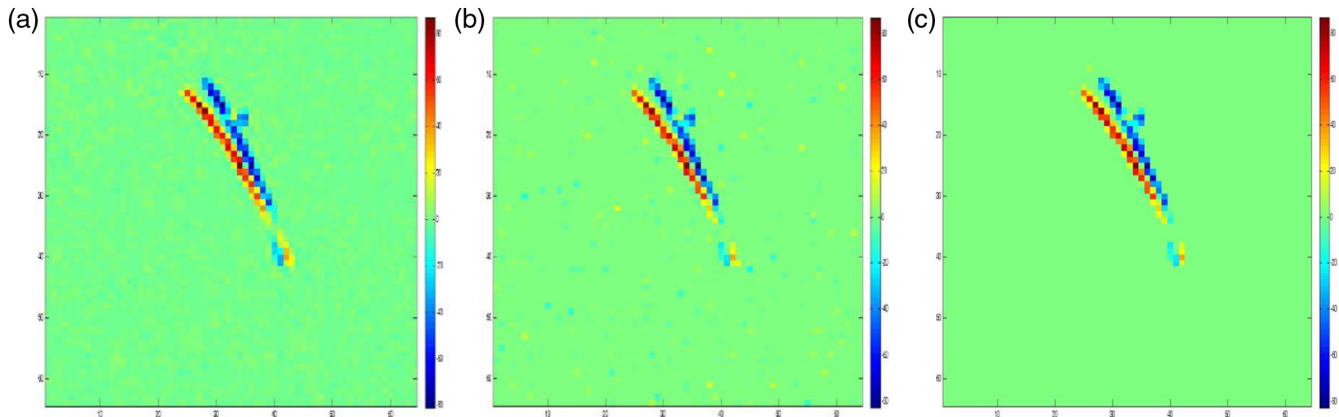


Fig. 7 Frame difference recovery comparison: (a) ground truth, (b) recovery by nonconvex algorithm, and (c) after denoising.

element for a better outline. Because the mask shows only the edges of the difference image and the areas with significant changes are inside the edges, the whole areas with significant changes are obtained via filling the holes inside the edges using a flood fill operation via the MATLAB[®] function “imfill.” This method keeps the most significant changes and removes error on the difference image so as to reduce the

reconstruction error in P-frames. Figure 7(c) shows the performance of this postprocessing (denoising) procedure. The flow chart for this procedure is described in Fig. 8.

Due to the frame difference sensing mechanism, the reconstruction error accumulates because every time we reconstruct P-frames using the difference between two consecutive frames. The error in the first P-frame is accumulated to the second P-frame. Therefore, the reconstruction of the first P-frame after I-frames is very important, and an improvement on this frame also improves following P-frames.

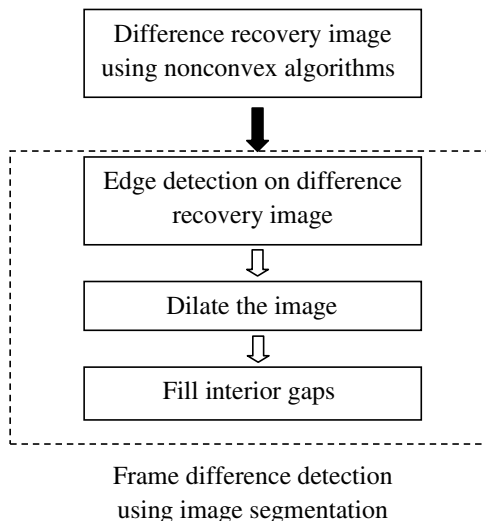


Fig. 8 Flow chart of denoising using image segmentation.

Table 1 PSNR values for the five reconstructed P-frames with four methods: difference images between two consecutive images without the denoising step (*m*-level); difference images between two consecutive images with the denoising step (denoising); difference images between P-frames and the I-frame without the denoising step (*m*-level*); and difference images between P-frames and the I-frame with the denoising step (denoising*).

	P ₀₁	P ₀₂	P ₀₃	P ₀₄	P ₀₅
<i>m</i> -level	40.8987	37.4587	36.2745	35.6323	35.0012
Denoising	42.3382	37.5839	36.6928	35.9371	35.0856
<i>m</i> -level*	40.8987	39.5386	40.2128	39.3341	39.5685
Denoising*	42.3382	40.5984	41.5240	40.7008	41.0858

Table 2 PSNR values for the five reconstructed P-frames with four methods: difference images between two consecutive images without the denoising step (m -level); difference images between two consecutive images with the denoising step (denoising); difference images between P-frames and the I-frame without the denoising step (m -level*); and difference images between P-frames and the I-frame with the denoising step (denoising*).

	P ₀₁	P ₀₂	P ₀₃	P ₀₄	P ₀₅
m -level	2.2994	3.4167	3.9157	4.2163	4.5340
Denoising	1.9482	3.3678	3.7317	4.0709	4.4901
m -level*	2.2994	2.6891	2.4883	2.7532	2.6799
Denoising*	1.9482	2.3802	2.1396	2.3523	2.2504

On the other hand, if the number of P-frames between two consecutive I-frames is small, we can compute the difference image between the P-frame and the previous I-frame instead to avoid the accumulated error from previous P-frames.

The next numerical experiment shows that we can apply the simple denoising procedure to improve the reconstruction results of the first P-frame and all the P-frames after that. In this numerical experiment, there are five P-frames after one I-frame. In Fig. 9, all five P-frames are plotted. The first row has five ground true frames (P₀₁ to P₀₅). For the second and third rows, we show the reconstruction results using the difference image between two consecutive frames, and the reconstruction results using the difference image between P-frames and the I-frame are shown in the fourth and fifth rows. The reconstruction results using m -level without the denoising step are shown in the second row (P₁₁ to P₁₅) and the fourth row (P₃₁ to P₃₅). The reconstruction results with the denoising step are shown in the third row

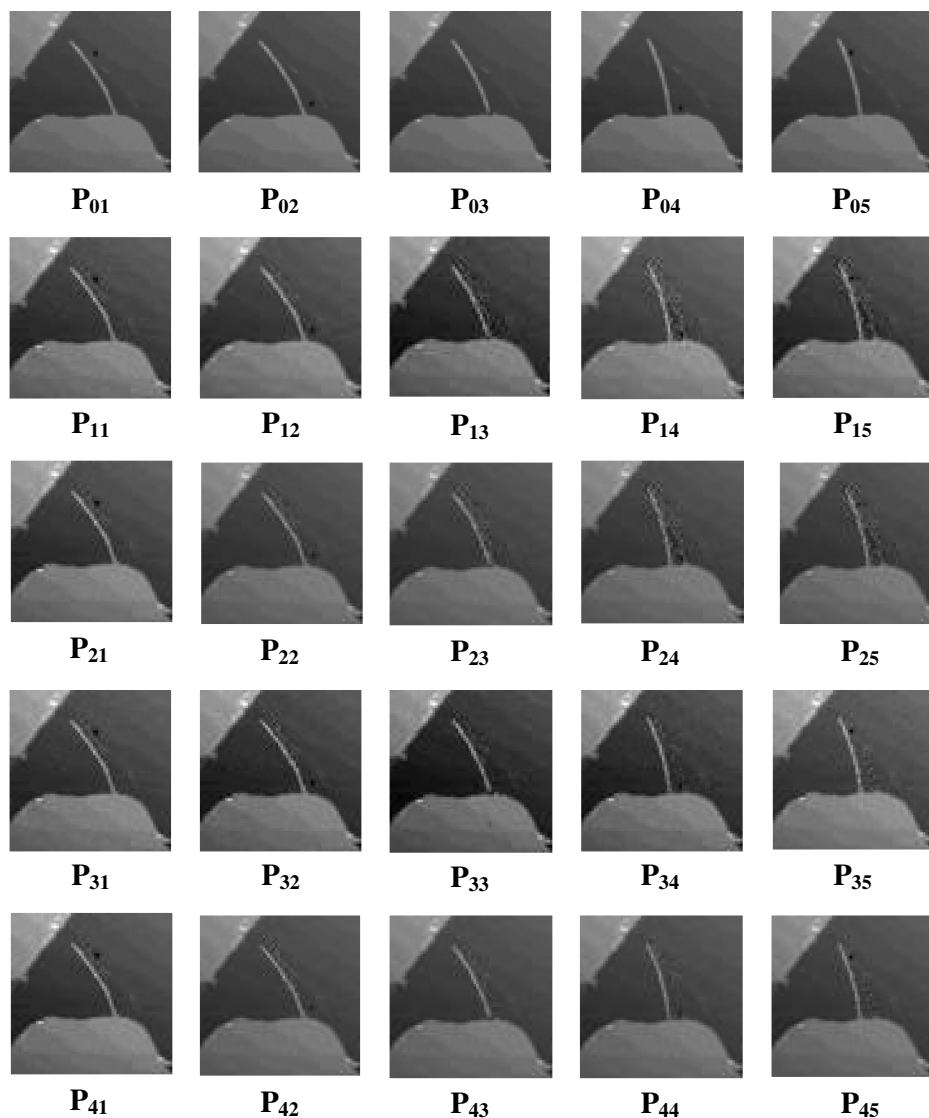


Fig. 9 Accumulation error, ground true frames (P₀₁ to P₀₅), m -level without the denoising (P₁₁ to P₁₅), m -level with the denoising (P₂₁ to P₂₅), m -level directly to I-frame without the denoising (P₃₁ to P₃₅), m -level directly to I-frame with the denoising (P₄₁ to P₄₅).

Table 3 P-frames reconstruction algorithm.

Algorithm

Initialize \mathbf{x}_0 , β , r and τ

for $l = 1$: **maxit**

 a. Compute K^l

 b. Update ω^l

 c. Apply one forward-backward iteration and check stopping rules.

end

d. Find the areas with significant changes

e. Reconstruct the P-frame by updating only the pixels values in the areas identified in the previous step.

(P_{21} to P_{25}) and the fifth row (P_{41} to P_{45}). The PSNR and RMSE values are shown in Tables 1 and 2. From both tables, we can see that the PSNR value is decreasing and the RMSE value is increasing for the five P-frames, if the difference images between two consecutive frames are used and the denoising step improves all P-frames, especially the first P-frame. However, if all the P-frames are compared with the I-frame, the improvement of the denoising step is large for all five P-frames. This numerical experiment suggests that we may choose to compare P-frames with the previous I-frame instead of the previous frame because the error in the previous P-frames will be accumulated.

The whole algorithm for P-frames reconstruction is depicted in Table 3. The steps (a) to (c) show the nonconvex sorted ℓ_1 calculation process, while steps (d) to (e) demonstrate

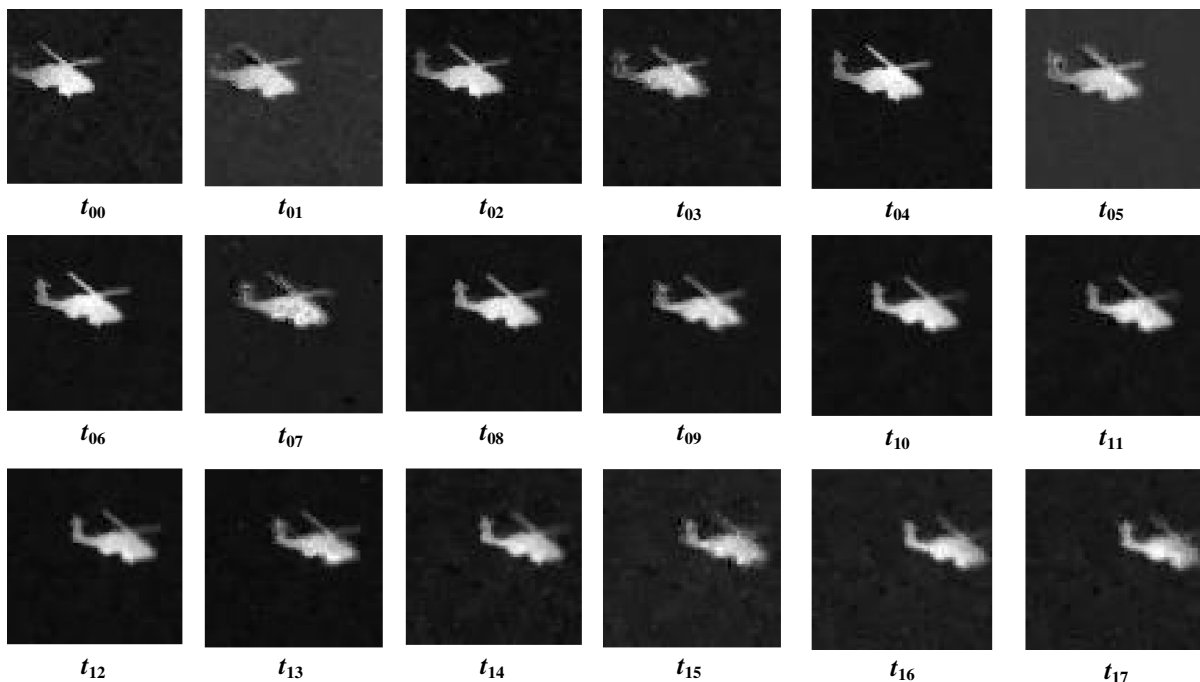
the edge-detection denoising procedure to reduce the error in the compressive video sensing.

3 Experiments

The projection measurement matrices can be implemented by spatial light modulators such as the digital micromirror device (DMD) and the liquid crystal on silicon. The DMD runs as fast as 32,000 Hz, and we use a DMD with 6000 Hz in the experiments. A DMD chip has several thousand microscopic mirrors arranged in a rectangular array on its surface. These mirrors correspond to the pixels in the image to be reconstructed. The mirrors can be individually rotated ± 12 deg to an on or off state. These two states correspond to ± 1 in the Bernoulli matrix. During the sampling process, the measurement matrix is sent to the DMD controller row by row. The matrices for P-frames are selected from the rear end of the matrix for the previous I-frame, e.g., if the previous I-frame measurement matrix is $\Phi \in \mathbb{R}^{m \times n}$, then the P-frame measurement matrix will be $\Phi(m - p + 1 : m, :) \in \mathbb{R}^{p \times n}$ with $p \ll m$. During the experiments, the irradiator (THORLABS LIU850A) is 850 nm near the IR source, and a silicon photodiode (THORLABS FDS1010) is chosen as the receiver sensor.

We validate the proposed nonconvex compressive video sensing system using two experiments: a linear moving object and a rotating object. In the first experiment with a linear moving airplane in Fig. 10, the frame rate is 10 fps. There is only one P-frame between two consecutive I-frames, i.e., $t_{00}, t_{02}, \dots, t_{16}$ are I-frames, while $t_{01}, t_{03}, \dots, t_{17}$ are P-frames. The sampling ratios are 18% and 8.5% for I-frames and P-frames, respectively. The proposed system records the whole scene in real time.

The second experiment is to capture the rotation of a fan. As shown in Fig. 11, each blade is designed with a different length for easy identification. There are three P-frames between two consecutive I-frames, and each row in Fig. 11

**Fig. 10** Moving object video recording.

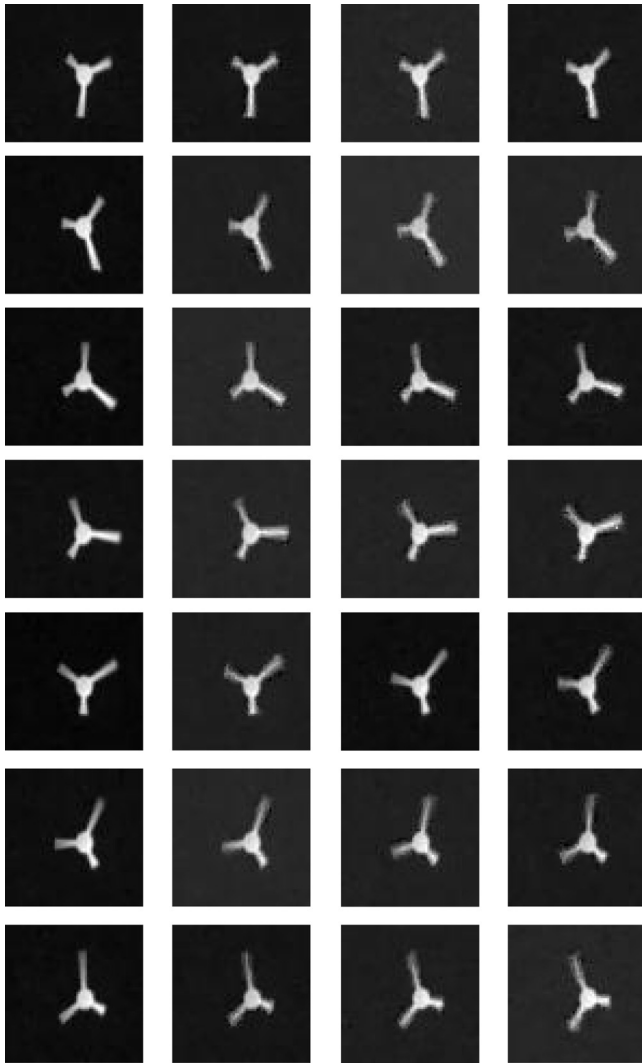


Fig. 11 Rotating object video recording.

shows one I-frame on the first column and three P-frames after the I-frame on the last three columns. The frame rate is 18 fps, and the sampling ratios are 20% and 9% for I-frames and P-frames, respectively.

4 Conclusions

Nonconvex compressive sensing algorithms require a fewer number of linear measurements to reconstruct a sparse signal than convex algorithms. In this work, the nonconvex sorted ℓ_1 approach is employed to reconstruct the difference images, which are sparse, and decrease the sampling rate. Furthermore, an edge-detection-based denoising step is applied to reduce the error on the difference image. Thus, it requires a smaller number of measurements compared to the conventional compressive video sensing. We tested our algorithm on the real-time video reconstruction in the experiments. Though the frame rate in the experiments is only 18 fps, it can reach up to 105 fps based on current DMD mirror speed (maximum 32,000 Hz).

Acknowledgments

This research work was partially supported under National Science Foundation Grants Nos. IIS-0713346

and DMS-1621798, Office of Naval Research Grants Nos. N00014-04-1-0799 and N00014-07-1-0935, the U. S. Army Research Laboratory, and the U. S. Army Research Office under Grant No. W911NF-14-1-0327.

References

1. M. Irvin, T. Kitazawa, and T. Suzuki, "A new generation of MPEG-2 video encoder ASIC and its application to new technology markets," in *Int. Broadcasting Convention Conf.*, Vol. 1, pp. 391–396 (1996).
2. G. J. Sullivan and T. Wiegand, "Video compression from concepts to the H.264/AVC standard," *Proc. IEEE* **93**(1), 18–31 (2005).
3. Gigapan, "Space shuttle discovery flight deck by national geographic," 2012, <http://www.gigapan.com/gigapans/102753> (July 2015).
4. M. Hefner, D. Duckett, and A. Loening, "High-speed video microscopy and computer enhanced imagery in the pursuit of bubble dynamics," *Ultrason. Sonochem.* **7**, 229–233 (2000).
5. O. Cossairt, M. Gupta, and S. K. Nayar, "When does computational imaging improve performance?" *IEEE Trans. Image Process.* **22**(2), 447–458 (2013).
6. D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory* **52**(4), 1289–1306 (2006).
7. H. Chen et al., "Infrared camera using a single nano-photodetector," *IEEE Sens. J.* **13**(3), 949–958 (2013).
8. Y. Wang, J. Cao, and C. Yang, "Recovery of seismic wavefields based on compressive sensing by an ℓ_1 -norm constrained trust region method and the piecewise random sub-sampling," *Geophys. J. Int.* **187**(1), 199–213 (2011).
9. B. Song et al., "Compressive feedback-based motion control for nanomanipulation: theory and applications," *IEEE Trans. Rob.* **30**(1), 103–114 (2014).
10. P. Zhang et al., "A compressed sensing based ultra-wideband communication system," in *IEEE Int. Conf. on Communications*, pp. 1–5 (2009).
11. M. Lustig et al., "Compressed sensing MRI," *IEEE Signal Process. Mag.* **25**(2), 72–82 (2008).
12. M. B. Wakin et al., "Compressive imaging for video representation and coding," in *Proc. of Picture Coding Symp.*, Vol. 1, pp. 1–6 (2006).
13. A. C. Sankaranarayanan et al., "Compressive acquisition of dynamic scenes," in *European Conf. on Computer Vision*, pp. 129–142 (2010).
14. S. Bi et al., "Compressive video recovery using block match multi-frame motion estimation based on single pixel cameras," *Sensors* **16**(3), 1–8 (2016).
15. M. Iliadis, L. Spinoulas, and A. K. Katsaggelos, "Deep fully-connected networks for video compressive sensing," arXiv:1603.04930 (2016).
16. J. Chen et al., "Residual distributed compressive video sensing based on double side information," *Acta Autom. Sin.* **40**(10), 2316–2323 (2014).
17. N. Eslahi, A. Aghagolzadeh, and S. Mehdi, "Image/video compressive sensing recovery using joint adaptive sparsity measure," *Neurocomputing* **200**, 88–109 (2016).
18. V. Stankovic, L. Stankovic, and S. Cheng, "Compressive video sampling," in *16th European Signal Processing Conf.* Vol. 1, pp. 1–6 (2008).
19. J. Zheng and E. L. Jacobs, "Video compressive sensing using spatial domain sparsity," *Opt. Eng.* **48**, 087006 (2009).
20. E. Candes, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.* **14**(5), 877–905 (2008).
21. R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3869–3872, IEEE (2008).
22. I. Daubechies et al., "Iteratively reweighted least squares minimization for sparse recovery," *Commun. Pure Appl. Math.* **63**(1), 1–38 (2010).
23. A. Cohen, W. Dahmen, and R. DeVore, "Compressed sensing and best k-term approximation," *J. Am. Math. Soc.* **22**(1), 211–231 (2009).
24. J. A. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory* **50**(10), 2231–2242 (2004).
25. E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory* **52**(2), 489–509 (2006).
26. D. Needell and J. Tropp, "CoSaMP: iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmon. Anal.* **53**(12), 93–100 (2010).
27. J. Yang and X. Yuan, "Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization," *Math. Comput.* **82**(281), 301–329 (2013).
28. R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization," *IEEE Signal Process. Lett.* **14**(10), 707–710 (2007).
29. "Fast L_1 - L_2 minimization via a proximal operator," arXiv:1609.09530 (2016).
30. Z. Xu et al., " $l_{1/2}$ regularization: a thresholding representation theory and a fast solver," *IEEE Trans. Neural Networks Learn. Syst.* **23**(7), 1013–1027 (2012).
31. X. L. Huang, L. Shi, and M. Yan, "Nonconvex sorted ℓ_1 minimization for sparse approximation," *J. Oper. Res. Soc. China* **3**, 207–229 (2015).

32. Y. Wang and W. Yin, "Sparse signal reconstruction via iterative support detection," *SIAM J. Imaging Sci.* 3(3), 462–491 (2010).
33. B. Olshausen and D. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature* 381, 607–609 (1996).
34. D. N. Hein and N. Ahmed, "Video compression using conditional replenishment and motion prediction," *IEEE Trans. Electromagn. Compat. EMC-26*(3), 134–142 (1984).
35. M. Duarte et al., "Single-pixel imaging via compressive sampling," *IEEE Signal Process. Mag.* 25(2), 83–91 (2008).
36. C. Li, "An efficient algorithm for total variation regularization with applications to the single pixel camera and compressive sensing," Master Thesis, pp. 10–80, Rice University (2009).

Liangliang Chen received his bachelor's and master's degrees in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2009 and 2007, respectively. Currently, he is pursuing his PhD at Michigan State University, East Lansing. His research interests include infrared sensor and imaging, ultraweak signal detection in nanosensors, signal processing, analog circuits, and carbon nanotube/graphene nanosensors.

Ming Yan received his PhD from the University of California, Los Angeles, in 2012. He is an assistant professor at the Department of Computational Mathematics, Science and Engineering and the Department of Mathematics, Michigan State University. His research interests include signal and image processing, optimization, and parallel and distributed methods for large-scale datasets.

Chunqi Qian received his BS degree in chemistry from Nanjing University and his PhD in physical chemistry from the University of California, Berkeley, in 2007. Following postdoctoral trainings at the National High Magnetic Field Laboratory and the National Institutes of Health, he joined Michigan State University as an assistant professor in radiology. His research interest includes the development and application of imaging technology in biomedical research.

Ning Xi received his DSc degree in systems science and mathematics from Washington University in St. Louis, Missouri, USA, in 1993. Currently, he is the chair professor of robotics and automation at

the Department of Industrial and Manufacturing System, and director of Emerging Technologies Institute of the University of Hong Kong. He is a fellow of the Institute of Electrical and Electronics Engineers (IEEE). His research interests include robotics, manufacturing automation, micro/nanomanufacturing, nanosensors and devices, and intelligent control and systems.

Zhanxin Zhou received her bachelor's and master's degrees in control engineering from the Second Artillery Engineering College, Xi'an, China, in 1992 and 1997, respectively. She received her PhD in control engineering from Beijing Institute of Technology, Beijing, China, in 2008. Her research interests include infrared imaging, imaging enhancement, nonlinear filter and optimal control.

Yongliang Yang received his BS degree in mechanical engineering from Harbin Engineering University, Harbin, China, in 2005. He received his MS and PhD degrees from the University of Arizona, Tucson, USA, in 2012 and 2014, respectively. He has been a research associate at Michigan State University since 2014. His research interests include micro/nanorobotics and their application in biomedicine.

Bo Song received his BEng degree in mechanical engineering from Dalian University of Technology, Dalian, China, in 2005, and his MEng degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 2009. Currently, he is pursuing his PhD at the Department of Electrical and Computer Engineering, Michigan State University, East Lansing. His research interests include nanorobotics, nonvector space control, compressive sensing, and biomechanics.

Lixin Dong received his BS and MS degrees in mechanical engineering from Xi'an University of Technology, Xi'an, China, in 1989 and 1992, respectively, and his PhD in microsystems engineering from Nagoya University, Nagoya, Japan, in 2003. He is an associate professor at Michigan State University. His research interests include nanorobotics, nanoelectromechanical systems, mechatronics, mechanochemistry, and nanobiomedical devices. He is a senior editor of the *IEEE Transactions on Nanotechnology*.