Research Article Vol. 30, No. 3 / 31 Jan 2022 / Optics Express 3577
Optics EXPRESS

Temporal compressive imaging reconstruction based on a 3D-CNN network

LINXIA ZHANG,[1,2] EDMUND Y. LAM,[3] AND JUN KE[1,2,*]

[1]Beijing Institute of Technology, School of Optics and Photonics, Beijing 100081, China
[2]Key Laboratory of Photo-electronic Imaging Technology and System, Ministry of Education of China, Beijing 100081, China
[3]Department of Electrical and Electronic Engineering, University of Hong Kong, Pokfulam, Hong Kong SAR, China
[*]jke@bit.edu.cn

Abstract: In temporal compressive imaging (TCI), high-speed object frames are reconstructed from measurements collected by a low-speed detector array to improve the system imaging speed. Compared with iterative algorithms, deep learning approaches utilize a trained network to reconstruct high-quality images in a short time. In this work, we study a 3D convolutional neural network for TCI reconstruction to make full use of the temporal and spatial correlation among consecutive object frames. Both simulated and experimental results demonstrate that our network can achieve better reconstruction quality with fewer number of layers.

1. Introduction

Using compressed sensing (CS), an imaging system can overcome the limitations of a detector to obtain better system performance, such as improvement in spatial, temporal, or spectral resolution [1–6]. Temporal compressive imaging (TCI), or snapshot compressive imaging (SCI), is a CS technology to image fast moving objects [7–10]. There are also works for imaging fast moving objects using a single pixel camera [11,12]. In TCI, data are modulated and compressed before being sampled. This not only relaxes the requirement of the detector frame rate, but also effectively reduces the amount of data collected. Therefore, TCI can break through the data transmission and storage limitations of a camera especially in non-visible bands, such as in infrared (IR) [8,13].

For TCI, fast moving objects are reconstructed from measurements using computational algorithms. In most traditional reconstruction processes, an optimization problem is defined and then solved in an iterative manner by using algorithms such as optical flow [14], over-complete dictionary learning [15], two-step iterative shrinkage thresholding (TwIST) [16], and Gaussian mixture model (GMM) [17]. In the past few years, more algorithms are being developed. In 2016, Yuan developed the generalized alternating projection-based total variation minimization (GAP-TV) [18]. In 2018, Liu et al. integrated the nonlocal self-similarity of video/hyperspectral frames and the rank minimization approach with the snapshot compressive imaging sensing process (DeSCI) [19] for reconstruction.

Although iterative algorithms present good reconstruction performance, they generally take a long time to obtain results. The time required to reconstruct an image can be up to a few hours. On the other hand, recently deep learning has received a lot of attention due to its powerful reconstruction and denoising performance. Several groups have reported using neural networks for object reconstruction in TCI. For example, Iliadis et al. presented a deep learning framework for video compressive sensing [20], where they used a small block size and a fully connected network structure for TCI reconstruction. Ma et al. developed a deep tensor ADMM-Net for video SCI systems [21]. The network is inspired by the alternating direction method of multipliers (ADMM) algorithm, and they use fully connected and convolution layers to replace

#448490 https://doi.org/10.1364/OE.448490
Journal © 2022 Received 18 Nov 2021; revised 9 Jan 2022; accepted 9 Jan 2022; published 19 Jan 2022

Optics EXPRESS

the iterative process in ADMM. Meanwhile, Qiao et al. developed an end-to-end convolutional neural network (E2E-CNN) and a plug-and-play (PnP) framework with deep denoising priors to solve the reconstruction problem in TCI [22].

In these networks, 2D convolution kernels are used, which focus on the spatial correlation of object pixels. In contrast, we use 3D convolution instead of 2D convolution. A 3D convolution kernel can make better use of the temporal correlation of continuous object frames, and reduce the amount of network parameters. As a result, the network is simpler, more effective, and faster. Besides 3D kernels, we also use multiple residual blocks to improve the reconstruction quality.

An important concern, when working with neural networks, is about the training data. If a network is trained using simulated data, its performance may be degraded significantly when actual measurements are corrupted by system errors, such as aberrations in an optical setup. To deal with this issue, we further develop a calibration algorithm for TCI measurements. After the calibration, these measurements can be directly sent to the reconstruction network that has been trained using simulation data.

This paper is organized as follows. In Section 2, we present the principle and the mathematical model of TCI. Next, in Section 3, we first briefly discuss several existing TCI reconstruction networks. Then, we explain the advantages of a 3D convolution kernel and a 3D-CNN layer, followed by details of our 3DTCI network structure. In Section 4, we present the experimental results obtained using simulated data, and then discuss the optical experiment results, together with the calibration algorithm used for TCI measurements. Finally, we highlight some conclusions in Section 5.

2. Principle of temporal compressive imaging

In TCI, a system consists of a measurement acquisition part and an object reconstruction part, as shown in Fig. 1. In the measurement acquisition, multiple object frames are modulated by the templates. They are then accumulated into one measurement frame, which is collected by a detector array. In the reconstruction, original frames are recovered, assuming object sparsity in the temporal domain.

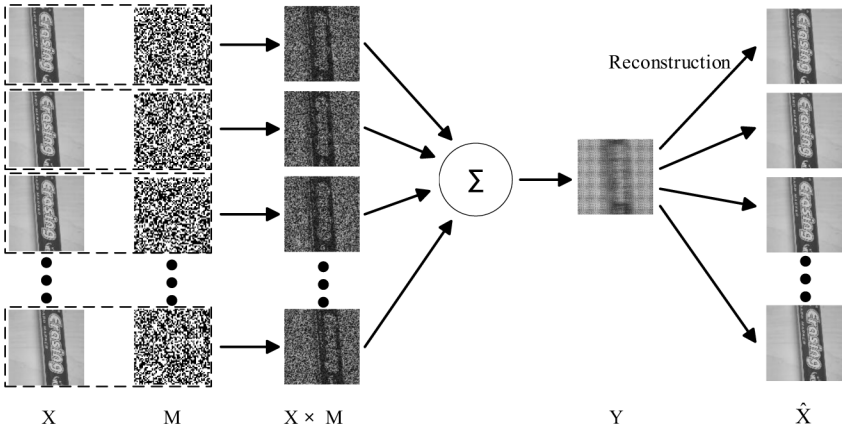


Fig. 1. Schematic diagram of temporal compressive imaging (TCI).

If the temporal compressive ratio is $t:1$, i.e., t object frames are compressed into one frame of measurements, the measurement acquisition process can be written as

$$Y = \sum_{k=1}^{t} M_k \odot X_k + N, \tag{1}$$

Optics EXPRESS

where $Y \in \mathbb{R}^{n_1 \times n_2}$ denotes the measurement frame, $X_k \in \mathbb{R}^{n_1 \times n_2}$ (with $k = 1, 2, \ldots, t$) represents the kth object frame, $M_k \in \mathbb{R}^{n_1 \times n_2}$ indicates the corresponding modulation template, the operator $\odot$ represents the dot product between two matrices, and $\boldsymbol{N}$ represents the noise. To simplify the notation, we rewrite Eq. (1) as

$$\boldsymbol{y} = \boldsymbol{\Phi x} + \boldsymbol{n}. \tag{2}$$

The measurement vector $\boldsymbol{y}$ has size $n_1 n_2 \times 1$. It is consisted of the pixels of a measurement frame Y in the lexicographical order. Similarly, we redefine an object frame X_k as a column vector $\boldsymbol{x}_k$. Then the original object, which represents $\{X_1, X_2, \ldots, X_t\}$, becomes $\boldsymbol{x} = [\ \boldsymbol{x}_1^T \quad \boldsymbol{x}_2^T \quad \cdots \quad \boldsymbol{x}_t^T\ ]^T$. The sensing matrix $\boldsymbol{\Phi} \in \mathbb{R}^{n_1 n_2 \times n_1 n_2 t}$ is defined as

$$\boldsymbol{\Phi} = \left[\ \mathrm{diag}(M_1) \quad \mathrm{diag}(M_2) \quad \cdots \quad \mathrm{diag}(M_t)\ \right], \tag{3}$$

where $\mathrm{diag}(M_k)$ is a diagonal matrix with diagonal elements set as the values of M_k. The vector $\boldsymbol{n}$ again represents the noise.

As discussed in the previous section, traditionally the original object frames can be reconstructed by solving an optimization problem, such as

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{\Phi x}\|_2 + \boldsymbol{Z}. \tag{4}$$

The regularization term $\boldsymbol{Z}$ incorporates the prior knowledge about the object. In this paper, instead of solving the above optimization problem iteratively, we focus on deep learning reconstruction algorithms.

3. Deep neural network based reconstruction for TCI

As discussed in Section 1, there have been several network setups proposed for TCI [20–22]. The fully connected (FC) network has a simple structure, consisting of up to 7 fully connected layers [20]. To reduce the amount of data, instead of full-size measurement frames, measurement blocks of size 11×11 are used as network inputs. The loss function used for training is the mean squared error (MSE). This network runs very fast, but it processes the object frames as one-dimensional data, and therefore loses the spatial-temporal 3D structure of an object frame sequence.

3.1. 2D networks for TCI reconstruction

Besides the fully connected network, there are several 2D networks which have been designed for TCI reconstruction. Here we discuss the ADMM-Net and E2E networks as examples. We refer them as 2D networks because only the height and width of a convolution kernel used in them can be controlled. We will discuss more on the definition of 2D and 3D convolutions in the following subsection.

ADMM-Net is motivated by the standard tensor ADMM algorithm [21]. The network is composed of K stages, which is equivalent to K iterations in the iterative algorithm. Therefore, each stage calculates the update of the reconstructed $\hat{\boldsymbol{X}}$ and the update of intermediate variables in one iteration of ADMM. Specifically, the calculation of $\hat{\boldsymbol{X}}$ is replaced by a fully connected network unit, while the calculation for updating intermediate variables is replaced by a combination of fully connected network unit and 2D convolution network units. ADMM-Net is an interpretable reconstruction scheme. More stages in the network are equivalent to more iterations. Thus, the reconstruction performance gets better, although with a longer run time.

Another network for TCI is an end-to-end deep learning network named E2E net [22]. It is based on an U-net encoder–decoder structure with residual connections. Five residual blocks are used in both the encoding and decoding. A skip connection is used to connect an encoding residual

Optics EXPRESS

block and its corresponding decoding residual block. The E2E net achieves fast reconstruction with high reconstruction quality. However, the convolution kernel used in E2E is still in 2D, which does not use the correlation between adjacent object frames.

As continuous object frames are highly similar between each other, if the correlation between them is used, it is expected to have improvement in network reconstruction performance. This is the basis why we investigate a 3D convolutional neural network (3D-TCI-CNN) in this work.

3.2. 3D CNN for TCI reconstruction

Fig. 2 presents the 2D and 3D convolution operations for a data cube of size $f_h \times f_w \times d_0$. In the 2D convolution as shown in Fig. 2(a), we can only control the height h_k and the width w_k of the kernel, while the depth is defined by the number of frames in the input data cube, d_0. Without zero padding, the data size obtained after convolution is $r_h \times r_w \times d_{r1}$, where $r_h = f_h - h_k + 1$, $r_w = f_w - w_k + 1$, and $d_{r1} = d_0 - d_0 + 1 = 1$. In the 3D convolution (Fig. 2(b)), we can control all three parameters, i.e., the height h_k, the width w_k and the depth d_1. Note that $d_1 \leq d_0$, and because the convolution is in 3D, the convolution result is a cube of size $r_h \times r_w \times d_r$, with $d_r = d_0 - d_1 + 1 \geq 1$. The number of images covered by the 3D convolution kernel can be controlled by setting d_1. In a convolutional layer of the network, multiple convolution kernels can be used. Thus, the output after a 3D convolutional layer is a 4D data of size $r_h \times r_w \times d_r \times c$, where c is the number of 3D convolution kernels.

Our design of the 3D convolutional network is shown in Fig. 3(a). To reduce the amount of data, we divide each measurement frame into $f_h \times f_w$ blocks for reconstruction. In the network, a measurement block is first expanded into a $f_h \times f_w \times t$ feature cube by a fully connected layer. The initial weights of the layer are set as $\mathbf{\Phi}^T$. Then, the feature cube is sent to a 3D CNN unit for

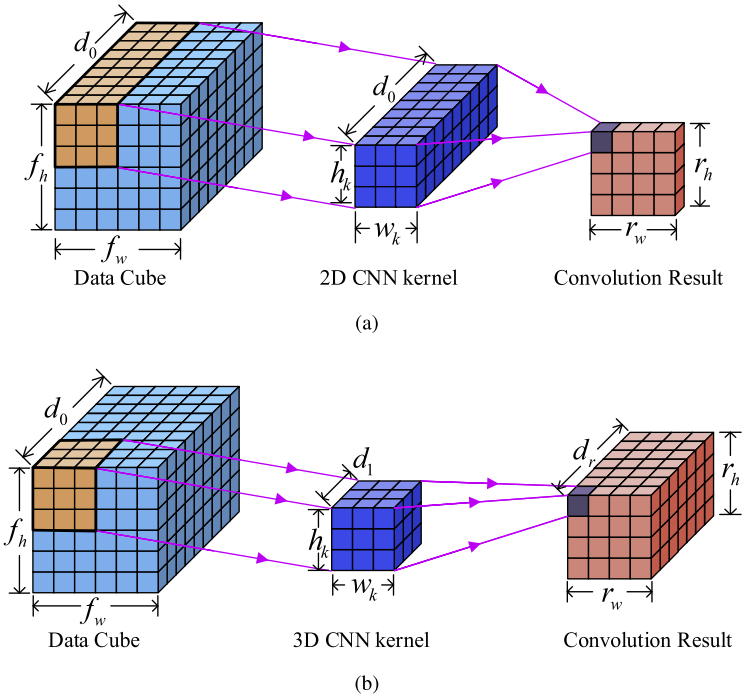


Fig. 2. (a) 2D and (b) 3D convolution operations. The dark blue cube represents the 2D and 3D kernels. The parameters, h_k, w_k and d_1 in (b) can be modified, while d_0 in (a) is fixed and determined by the data cube.

Optics EXPRESS

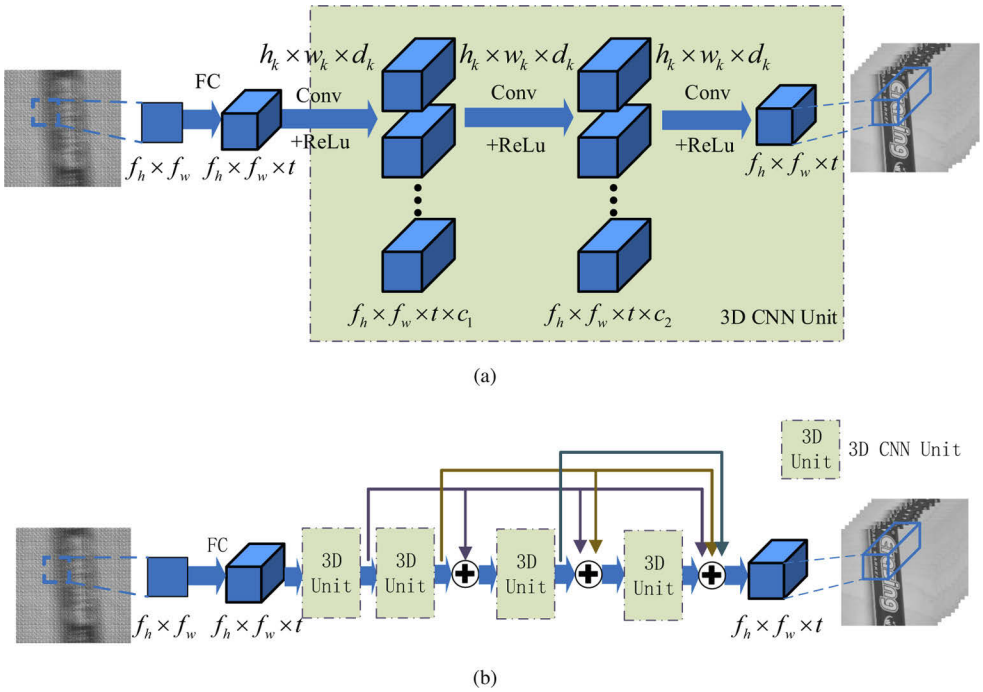


Fig. 3. 3D-TCI-CNN Networks. Structure diagram of (a) 3DTCI (3D-TCI-CNN) and (b) 3DTCI-R4 (3D-TCI-CNN with four 3D-CNN Units).

feature extraction and refinement. There are three convolution layers in the unit. As discussed above, the convolution kernel size is $h_k \times w_k \times d_1$. For the first two layers, there are c_1 and c_2 kernels, respectively. The third layer uses only one kernel. Then, the final output is a cube of size $f_h \times f_w \times t$, representing t frames of $f_h \times f_w$ object blocks.

To further improve the reconstruction quality, one way is to add more convolution units to learn the features better. However, increasing the depth of a network may result in losing object information. Hence, we use the idea of a residual network, adding jump connections between 3D CNN units to better retain the object information. Figure 3(b) shows such a network with four 3D-CNN Units. We refer it as 3DTCI-R4.

Note, in the following experiments, we set the block size $f_h \times f_w$ as 11×11 or 16×16 to reduce computational resource requirement. The convolution kernel size $h_k \times w_k \times d_k$ is set to $5 \times 5 \times 3$. The compression ratio t is $8:1$ or $9:1$.

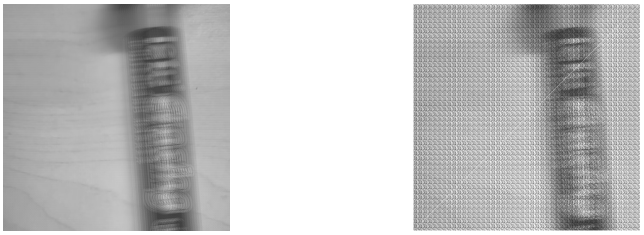
4. Experimental results

4.1. Numerical experiments

We first examine the reconstruction performance of the 3D TCI-CNN network by simulation experiments. The GPU used is NVIDIA RTX 2080Ti, which has 11GB device memory and 4352 CUDA cores. We collect several videos as the training date set, which contain over 4000 frames of moving objects. The resolution of one frame is 528×528. Each frame is divided into (11×11) blocks as training samples. We then simulate the TCI measurement collection process, where the temporal compression ratio is set to $9:1$. An image acquired using a low frame rate camera without TCI is shown in Fig. 4(a). It can be seen that the image is blurred due to the motion of the object. Using random binary patterns, a TCI measurement frame is shown in

Optics EXPRESS

Fig. 4(b). Then, the TCI measurement frame is sent to the trained network to reconstruct nine object frames.



(a) (b)

Fig. 4. Compressed image with 11% sampling rate. (a) Compressed image without masks. (b) Compressed image with masks.

Besides the 3D TCI-CNN network, we also use other algorithms for reconstruction, including iterative schemes such as GMM [23], GAP [18] and DeSCI [19], and deep learning methods such as FC [20], ADMM [21], and E2E [22] network. The reconstruction results are shown in Fig. 5. Figure 5(a) presents the true object frames. Figure 5(b) –(i) are the reconstruction results using different methods. To save space, only the first, fifth, and ninth reconstructed frames are presented in the figure. The peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) of the reconstructed objects and the run time are summarized in Table 1.

Table 1. The PSNR(dB)/SSIM/Time(s) values for reconstructions in Fig. 5. Notations 'size$_1$' and 'full' represent the blocks size (11×11) and the full object size, respectively.

Method	GMM	GAP	DeSCI	FC	E2E	ADMM	3DTCI	3DTCI-R4
	size$_1$	full	full	size$_1$	full	full	size$_1$	size$_1$
PSNR	25.00	27.26	41.10	23.76	30.99	28.37	29.63	31.11
SSIM	0.899	0.931	0.991	0.873	0.946	0.951	0.965	0.973
Time	17.41	36.46	42768	1.80	1.80	32.23	0.19	0.54

It can be seen from Fig. 5 and Table 1 that the reconstruction qualities by DeSCI, E2E, ADMM and 3DTCI-R4 are visually and quantitatively better. The DeSCI algorithm has the best reconstruction PSNR. However, it also has the longest run time, amounting to 11.88 hours. For E2E network, its run time is as short as 1.8s. Its reconstruction quality is also very good, except that in some frames, the residual noise is large, which reduces its PSNR and SSIM. The 3DTCI and 3DTCI-R4 networks achieve a good balance between the time consumption and reconstruction quality. The PSNR of the reconstruction using the latter is higher than 31dB, and the SSIM is higher than 0.97 with network run time less than 0.6s.

In order to verify the robustness of 3DTCI, four more simulation experiments are carried out using two horizontally moving and two rotating objects with size 385×385. The temporal compression ratio is still $9:1$. In this experiment, we collect over 3000 frames of moving objects with a resolution 385×385 to make the training set. Figure 6 presents the TCI measurement frame and the reconstructions using 3DTCI-R4 network. It can be seen that due to the movement or rotation of an object, the TCI measurement frames in the first row of Fig. 6 are blurred. The letters and numbers cannot be clearly recognized. However, the first, fifth and ninth reconstructed frames presented in lines 2, 3 and 4 have satisfactory visual quality. The PSNR, SSIM and the reconstruction time of the 3DTCI network and other reconstruction algorithms are shown in Table 2. From Table 2, once again we can observe that the reconstruction quality of 3DTCI-R4 network is the second best, while DeSCI presents the highest PSNR and SSIM values. However,

Optics EXPRESS

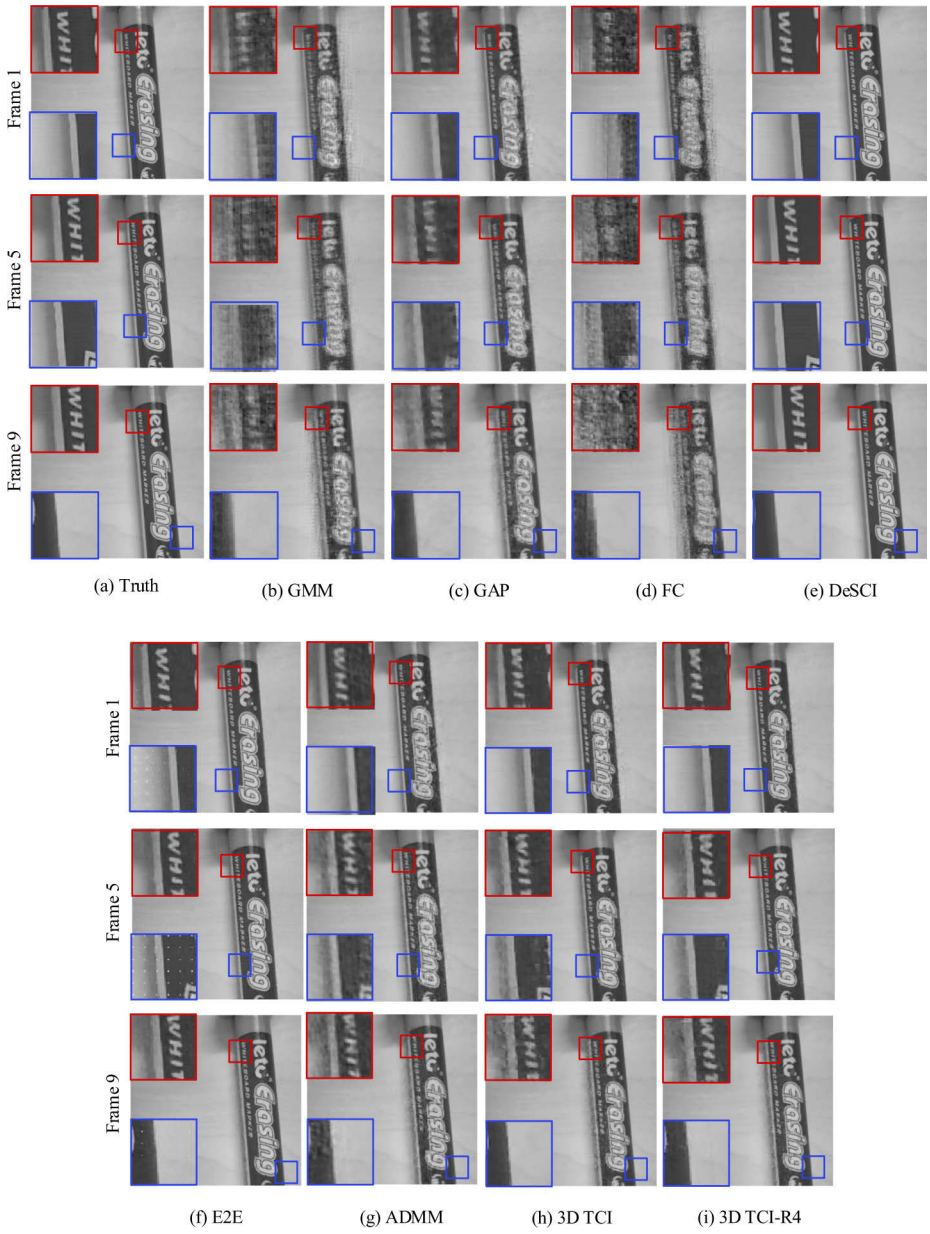


Fig. 5. The original frames and reconstructions using different methods.

the run time of 3DTCI-R4 is only 0.32s, far smaller than that of the DeSCI algorithm. This run time makes the reconstruction speed using 3DTCI-R4 reaching 27 frames per second (fps), which makes it very useful for real-time imaging applications.

In the third set of simulation experiments, we train the 3DTCI-R4 network on the `DAVIS2017` date set [24], and then test it on three other widely used date sets, namely, `Kobe` [17], `Aerial` and `Vehicle` [21]. The resolution of an object is 256×256. The training data contain continuous frames of 90 scenes. The compression ratio t is set to $8:1$. To understand the effect of block size to reconstruction quality, in this set of experiment, we train and test the network

Optics EXPRESS

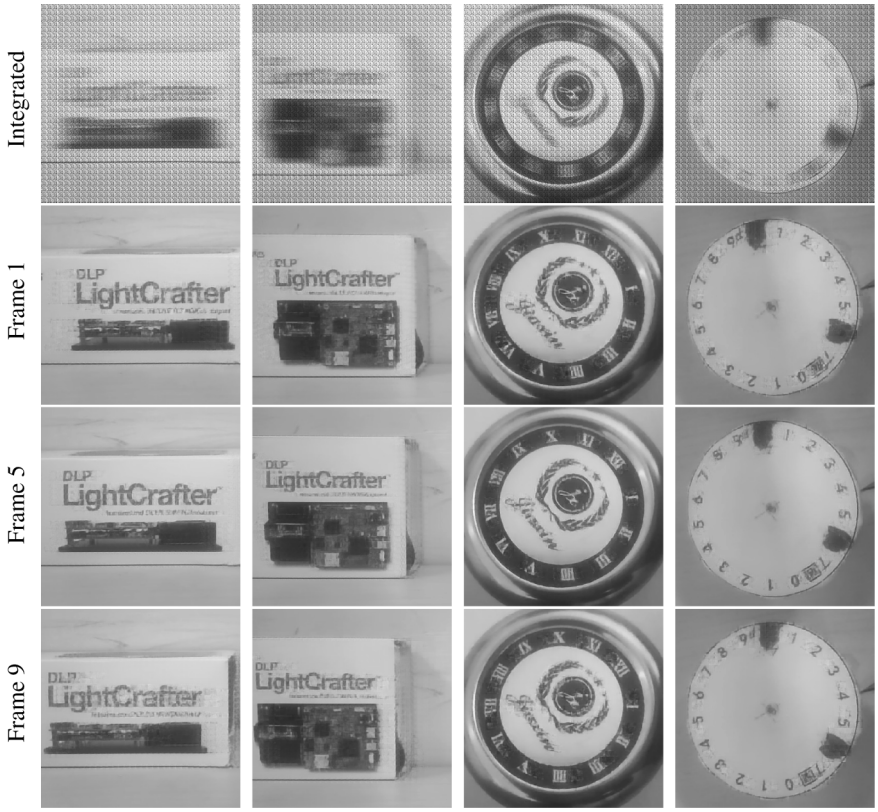


Fig. 6. Reconstruction results of moving/rotating targets using the 3DTCI-R4 network.

Table 2. The PSNR(dB)/SSIM/Time(s) values for reconstructions in Fig. 6, where 'size$_1$' and 'full' represent the blocks size (11×11) and the full object size, respectively.

Object		GMM	GAP	DeSCI	FC	E2E	ADMM	3DTCI	3DTCI-R4
		size$_1$	full	full	size$_1$	full	full	size$_1$	size$_1$
Words 1	PSNR	27.21	29.84	34.5	27.96	28.75	25.06	29.99	30.38
	SSIM	0.867	0.822	0.973	0.885	0.854	0.848	0.926	0.931
	Time	4.56	11.7	24372	0.80	0.16	18.6	0.09	0.31
Words 2	PSNR	28.80	30.59	35.70	28.97	29.47	24.68	30.47	31.43
	SSIM	0.878	0.912	0.963	0.877	0.851	0.805	0.912	0.925
	Time	4.67	11.5	23796	0.81	0.16	18.6	0.09	0.31
Nums 1	PSNR	23.99	26.69	34.27	25.42	28.24	23.15	25.03	27.22
	SSIM	0.780	0.889	0.977	0.832	0.914	0.800	0.822	0.892
	Time	4.59	11.5	24012	0.81	0.16	18.6	0.09	0.31
Nums 2	PSNR	28.74	31.49	40.03	29.64	32.36	29.07	30.07	31.74
	SSIM	0.871	0.932	0.987	0.894	0.947	0.906	0.910	0.935
	Time	4.70	11.4	23904	0.81	0.16	18.6	0.09	0.33
Average	PSNR	27.19	29.65	36.13	28.0	29.70	25.49	28.89	30.19
	SSIM	0.849	0.914	0.975	0.872	0.892	0.840	0.892	0.921
	Time	4.61	11.5	24021	0.81	0.16	18.6	0.09	0.32

Optics EXPRESS

3DTCI-R4 for 2 block sizes, (11×11) and (16×16). For the reconstruction methods, GMM, FC network and 3DTCI network, the block size is still set as (11×11). For the E2E network, we use the full size objects and also object blocks of size (16×16). The reconstruction results of various algorithms are shown in Table 3. It can be seen that the reconstruction performance of 3DTCI-R4 using block size (11×11) is slightly better than that of the E2E and ADMM methods. As the block size increases to (16×16), the PSNR value of 3DTCI-R4 increases to 28.13dB. Similarly, for the E2E network, as the block size is increased from (16×16) to full size, the reconstructions becomes better. However, even using full size objects, the reconstructions obtained using E2E are still worse than the reconstructions obtained using 3DTCI-R4 when the block size is (16×16). About the run time, we can see that it increases as the block size becomes larger for E2E and 3DTCI-R4. Thus, among the 5 network, 3DTCI-R4 presents the highest PSNR. From Table 3, we can also see that the SSIM of 3DTCI-R4 is the highest among the 5 network algorithms. The PSNR and SSIM of E2E algorithm are suboptimal. Note that also, 3DTCI-R4 net only contains 13 layers (convolution layers and full connection layers), which is half of the 24 layers in E2E and $1/7$ of the 99 layers in ADMM. Additionally, with more computing resources, 3DTCI network is expected to have improved reconstruction performance using slightly more layers.

Table 3. The PSNR(dB)/SSIM/Time(s) of reconstruction results using DAVIS2017 date sets, where 'size$_1$', 'size$_2$', and 'full' represent the blocks size (11×11), (16×16) and the full object size, respectively.

Object		GMM	GAP	DeSCI	FC	E2E	ADMM	3DTCI	3DTCI-R4
		size$_1$	full	full	size$_1$	size$_2$/full	full	size$_1$	size$_1$/size$_2$
Kobe	PSNR	23.94	26.45	33.25	28.22	24.99/29.02	30.15	28.30	29.03/29.61
	SSIM	0.800	0.885	0.952	0.873	0.781/0.861	0.890	0.873	0.893/0.905
	Time	5.87	6.50	6445	0.299	0.040/0.084	13.03	0.037	0.126/0.183
Aerial	PSNR	24.67	25.05	25.33	26.54	25.18/27.52	26.85	26.44	27.49/27.72
	SSIM	0.822	0.828	0.860	0.855	0.790/0.822	0.860	0.848	0.877/0.884
	Time	5.62	6.40	6440	0.293	0.041/0.083	13.02	0.034	0.126/0.183
Vehicle	PSNR	26.13	24.82	27.04	26.37	25.48/26.40	25.42	26.08	26.89/27.05
	SSIM	0.847	0.838	0.909	0.856	0.773/0.886	0.780	0.833	0.867/0.881
	Time	5.66	6.40	6450	0.288	0.042/0.084	13.02	0.033	0.125/0.183
Average	PSNR	24.91	25.44	28.54	27.05	25.21/27.65	27.47	26.91	27.80/28.13
	SSIM	0.823	0.850	0.907	0.861	0.781/0.876	0.843	0.851	0.879/0.890
	Time	5.72	6.43	6445	0.293	0.041/0.084	13.02	0.034	0.125/0.183

In the last experiment using simulated data set, we test the run time of the E2E, ADMM, 3DTCI and 3DTCI-R4 networks with different measurement frame sizes. The run times are plotted in Fig. 7. It is clear that ADMM takes too much time for reconstruction. When the object size is small, the run time of E2E is smaller than 3DTCI-R4, because 2D convolution kernels are used in E2E, which requires less calculation than 3D kernel usually. However, the run time of E2E is larger than 3DTCI-R4 when the object resolution becomes large. In Fig. 7, it means the pixel number larger than 450. The reason is that E2E processes a measurement frame as one block. As the object resolution increases, its request to computational resource increases fast. On the other hands, the run time of 3DTCI-R4 does not increase much because 3DTCI-R4 processes blocks with a pre-defined size.

Optics EXPRESS

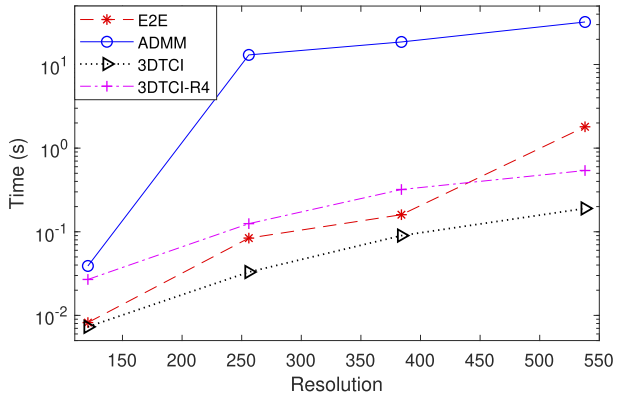


Fig. 7. The run time of E2E, ADMM, 3DTCI, and 3DTCI-R4 vs. measurement frame size or object resolution.

4.2. Optical experiments

4.2.1. Experimental platform and data preparation

We also test the 3DTCI-CNN network using a TCI experimental platform as shown in Fig. 8. Rotating numbers and small check boards are used as moving objects. These are printed on a paper plate that is controlled by a motor. In the experimental platform, the spatial light modulator is a Digital Micromirror Device (DMD), DLP6500. We use a random binary matrix as the modulation mask. The temporal compression ratio is 9 : 1. The camera used to acquire the compressed measurements is Basler ACA720-520.

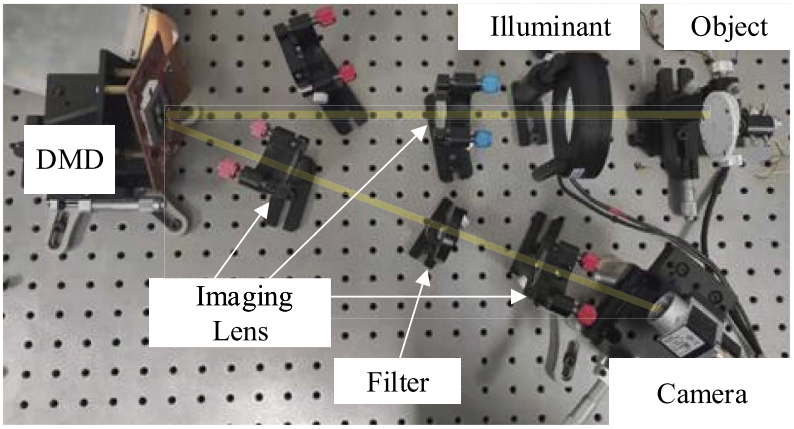


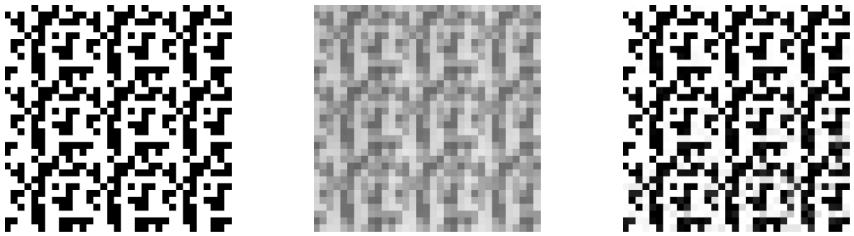
Fig. 8. Temporal compressed imaging experimental platform.

Notice that for optical experiments, it is difficult to prepare the training data, because the system is short of the original high-speed object frames as the network training label. Thus, we can only use simulated measurements for network training. We capture videos at a frame rate 100fps for the training set. Over 8000 images of size (121×121) are collected as the high speed object frames. Then, each set of 9 frames is numerically modulated using templates and compressed into one measurement frame. The high speed frames and the calculated measurement frames become a data pair for network training. For testing, the exposure time of the camera is set to $18ms$. Nine object frames are reconstructed from one measurement frame. Thus, the

Optics EXPRESS

reconstructed single frame is equivalently captured during $2ms$. The reconstructed video rate is 500fps.

Except for the lack of high speed object frames, above simulated TCI measurement frames are different from the acquired measurement frames due to errors such as noise, optical aberrations, or an inherently non-ideal system point spread function [9]. As shown in Fig. 9(a), the modulation template used in a simulation experiment is a standard binary template. However, the acquired images are blurred, as shown in Fig. 9(b). In order to solve this issue, we use the non-uniformity calibration method as discussed in [13] to correct the blurred measurement frame. Then, we send the results to the network, which is trained using simulated data, for reconstruction.



(a) (b) (c)

Fig. 9. (a) Standard mask; (b) Collected mask; and (c) Corrected mask.

The measurement calibration process is based on defining a calibration matrix $\boldsymbol{G}$. To compute $\boldsymbol{G}$, a uniform white light source is used to illuminate the DMD, on which a large number of random binary templates are loaded. Then, a detector array is used to take the measurement frames, such as the frame shown in Fig. 9(b). We use n column vectors $\{\boldsymbol{m}_i, i = 1, 2, \ldots, n\}$ to represent the templates on DMD. Another set of vectors $\{\boldsymbol{p}_i, i = 1, 2, \ldots, n\}$ are used to represent the corresponding measurement frames. Then, the measured frames and the binary patterns can be related as

$$\boldsymbol{P} = \boldsymbol{H}\boldsymbol{M}, \tag{5}$$

where $\boldsymbol{M} = \begin{bmatrix} \boldsymbol{m}_1 & \boldsymbol{m}_2 & \ldots & \boldsymbol{m}_n \end{bmatrix}$ and $\boldsymbol{P} = \begin{bmatrix} \boldsymbol{p}_1 & \boldsymbol{p}_2 & \ldots & \boldsymbol{p}_n \end{bmatrix}$.

Ideally, the calibration matrix $\boldsymbol{G}$ can be defined as $\boldsymbol{G} = \boldsymbol{H}^{-1}$, thus $\boldsymbol{G}\boldsymbol{P} = \boldsymbol{M}$. However, to make the calibration stable, a large number of templates are used. Thus, instead of finding $\boldsymbol{H}$ first, we use the least square method to define a matrix $\boldsymbol{G}$ as

$$\boldsymbol{G} = \boldsymbol{M}\boldsymbol{P}^T(\boldsymbol{P}\boldsymbol{P}^T)^{-1}. \tag{6}$$

Then, we can use the matrix $\boldsymbol{G}$ for calibration. The calibrated result of Fig. 9(b) is shown in Fig. 9(c). Clearly, the calibrated measurement frame is more similar to the original binary pattern.

4.2.2. Optical experiment results

Four rotating objects, number 4 and 8, a 4-bar target, and a check board are used in the experiment. The spatial resolution of the objects is 121×121. The temporal compressive ratio is still $9:1$. Figure 10(a)-(d) present the TCI measurement frames, while (e)–(h) are the calibrated results. Once again, the calibrated measurement frames are clearer with a higher spatial resolution. We then send the calibrated measurements to the trained network for reconstruction. The results are shown in Fig. 11.

Figure 11(a)-(h) present the reconstructions using GMM, GAP, DeSCI, FC, E2E, ADMM, 3DTCI, and 3DTCI-R4. The first and ninth frames of the reconstructed object sequence are shown. Clearly, all algorithms can reconstruct the 4 objects. For the first 3 objects that do not

Optics EXPRESS

have too many details, the reconstructions using various algorithms are similar. However, for more complex objects, such as a check board, the reconstructions using DeSCI, E2E, ADMM, and 3DTCI-R4 are better. Figure 12 presents the fifth reconstructed frame and its zoomed-in detail using the above four methods. Visually, it can be observed that DeSCI presents the best reconstruction. The reconstruction using the E2E network is more noisy. The reconstruction using the ADMM network is equally blurred. Using the 3DTCI-R4 network, although there is blockiness in the reconstructed frame, the resolution is the second best visually. To quantitatively evaluate the reconstructions, we calculate the SSIM values of the reconstructions and a standard check board. The SSIM values of the reconstructions obtained using DeSCI, E2E, ADMM, and 3DTCI-R4 are 0.5328, 0.4767, 0.4856, and 0.5044, respectively. Once again, we observe that 3DTCI-R4 present the second best results, while DeSCI presents the best performance in terms of SSIM. For better visualization, the pixel values along the green and the red lines in Fig. 12 are also plotted. The resulting curves are shown in Fig. 13. Once again, the reconstruction using the 3DTCI-R4 network has the second best contrast among the 4 methods.

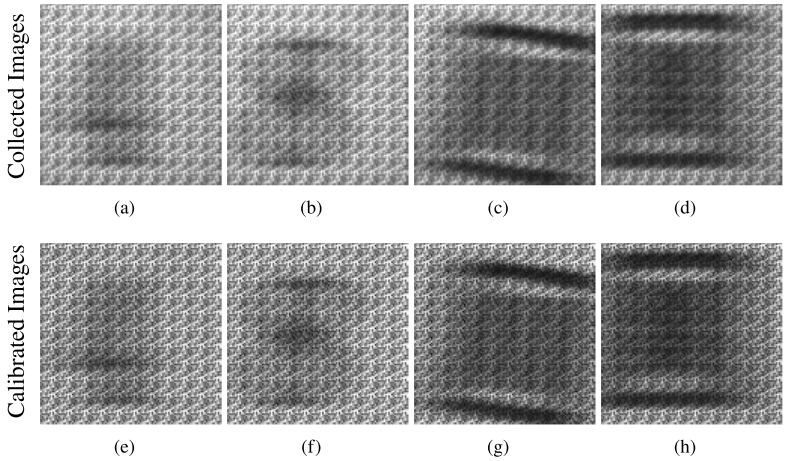
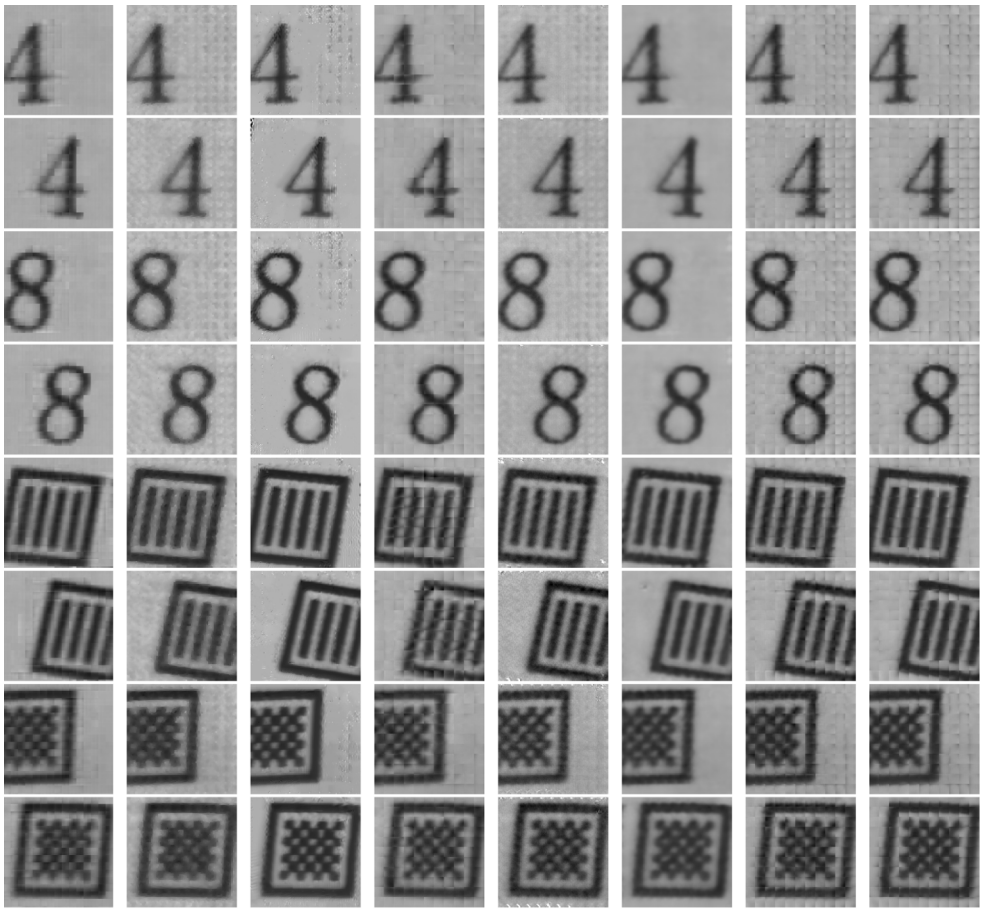


Fig. 10. (a)–(d) collected images; and (e)–(h) corrected images.

We also do an experiment to see how the residual calibration error affects reconstructions. Figure 14(a) and (b) are the reconstruction and its enlarged detail from a simulated measurement frame in the training set. Note there is no measurement error in the simulated measurement frame. The reconstruction algorithm is the 3DTCI-R4 network. We can see that the block artifact in Fig. 14 is less than it in Fig. 12(d). Thus, the artifact is not only caused by dividing a measurement frame into blocks, but also from residual calibration error. At the same time, the reconstruction quality in Fig. 14 is comparable to Fig. 12(d). The SSIM values for the two reconstructions are 0.5044 and 0.5414.

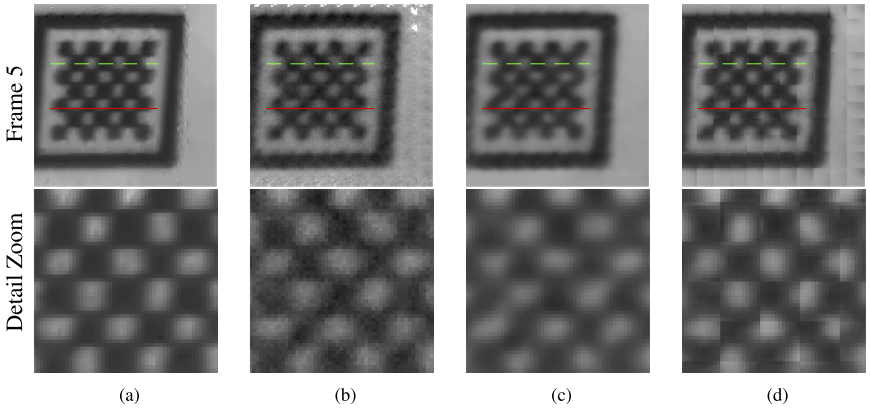
Using the 4 objects, we also compare the run time of the 8 methods. The results are shown in Table 4. Due to the reduced object spatial resolution, the run time decreases significantly. Compared with iterative algorithms, the 5 networks all require a much shorter run time. However, it can be observed from Table 1, Table 2, and Table 4 that as the size of an object increases from 121×121 to 528×528, the run time of 3DTCI-R4 becomes the least compared to the networks FC, E2E, and ADMM. Generally, the reconstruction quality using DeSCI is the best among all methods. However, 3DTCI-R4 requires the shortest run time for large objects with the second best reconstruction quality.

Optics EXPRESS



(a) GMM (b) GAP (c) DeSCI (d) FC (e) E2E (f) ADMM (g) 3DTCI (h) 3DTCIR4

Fig. 11. Experimental results of different algorithms. Four sets of experimental results are shown here. To save space, rows 1-2, 3-4, 5-6, and 7-8 are the 1st and 9th frames of the four sets of experiments, respectively.



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

Fig. 12. Reconstruction comparisons of (a) DeSCI, (b) E2E, (c) ADMM, (d) 3DTCIR4.

Optics EXPRESS

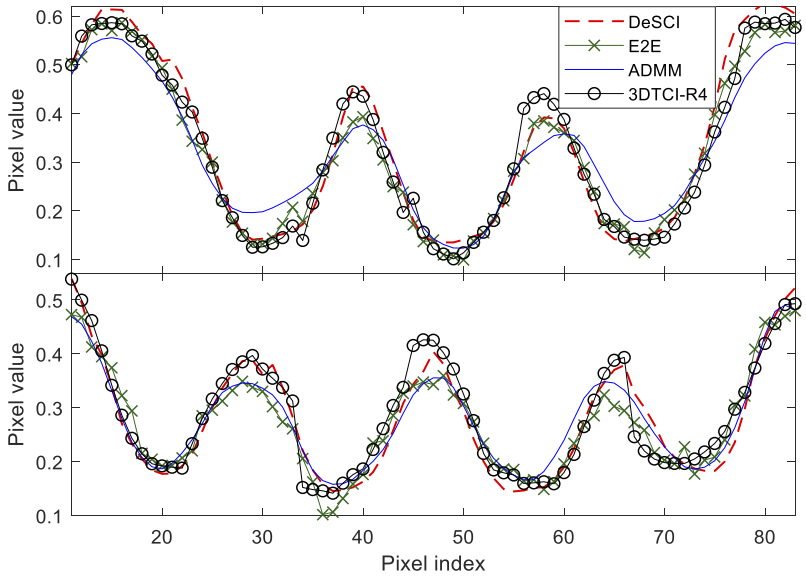
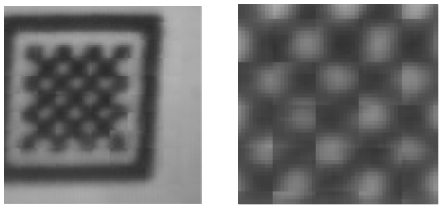


Fig. 13. Pixel values along (Up) Green dashed and (dowm) Red solid lines in Fig. 12.



(a) (b)

Fig. 14. A reconstruction and its enlarged detail obtained from a measurement frame without residual calibration error.

Table 4. The run time (s) of different algorithms in Fig. 11.

Method	GMM	GAP	DeSCI	FC	E2E	ADMM	3DTCI	3DTCI-R4
Num 4	2.289	1.468	1841	0.0012	0.0079	0.0390	0.0073	0.0269
Num 8	2.211	1.371	1823	0.0012	0.0080	0.0389	0.0072	0.0269
4 Bar Target	2.273	1.464	1852	0.0012	0.0089	0.0390	0.0071	0.0269
check board	2.154	1.320	1818	0.0012	0.0080	0.0390	0.0074	0.0268
Average	2.232	1.406	1833	0.0012	0.0082	0.0390	0.0073	0.0269

5. Conclusion

In this paper, we propose the 3DTCI CNN network, which is used to solve the TCI reconstruction problem. In this network, we combine the 3D convolutional network unit and the residual network idea to improve the reconstruction quality visually. In addition, a measurement calibration algorithm is designed so that the calibrated measurement frame can be used for reconstruction using a network trained by simulated data. Both simulation experiments and optical experiments show that the 3DTCI CNN network can reconstruct the continuous object frames very well.

Note that, our networks require that a measurement frame is divided into blocks and each block is reconstructed individually. This is mainly due to the limited computational resource. It

Optics EXPRESS

limits the network reconstruction performance, since block artifact can be observed. To reduce the artifacts, one way is to enlarge the block size. On the other hand, dividing frames into blocks has its advantage in terms of network run time. For large object size, we observe that the run time of networks processing a measurement frame as one unit, such as E2E, increases very fast. On the other hand, 3DTCI-R4 has comparable slowly increased run time.

Funding. National Natural Science Foundation of China (61675023).

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

References

1. M. A. Neifeld and J. Ke, "Optical architectures for compressive imaging," Appl. Opt. **46**(22), 5293–5303 (2007).
2. J. Ke and E. Y. Lam, "Object reconstruction in block-based compressive imaging," Opt. Express **20**(20), 22102–22117 (2012).
3. S. Lohit, K. Kulkarni, R. Kerviche, P. Turaga, and A. Ashok, "Convolutional neural networks for noniterative reconstruction of compressively sensed images," IEEE Trans. Comput. Imaging **4**(3), 326–340 (2018).
4. J. Ke and E. Y. Lam, "Image reconstruction from nonuniformly spaced samples in spectral-domain optical coherence tomography," Biomed. Opt. Express **3**(4), 741–752 (2012).
5. M. F. Duarte and R. G. Baraniuk, "Spectral compressive sensing," Applied and Computational Harmonic Analysis **35**(1), 111–129 (2013).
6. T.-H. Tsai, P. Llull, X. Yuan, L. Carin, and D. J. Brady, "Spectral-temporal compressive imaging," Opt. Lett. **40**(17), 4054–4057 (2015).
7. P. Llull, X. Yuan, X. Liao, J. Yang, D. Kittle, L. Carin, G. Sapiro, and D. J. Brady, "Temporal compressive sensing for video," in *Compressed Sensing and its Applications*, (Springer, 2015), pp. 41–74.
8. Q. Zhou, J. Ke, and E. Y. Lam, "Near-infrared temporal compressive imaging for video," Opt. Lett. **44**(7), 1702–1705 (2019).
9. J. Ke, L. Zhang, Q. Zhou, and E. Y. Lam, "Broad dual-band temporal compressive imaging with optical calibration," Opt. Express **29**(4), 5710–5729 (2021).
10. X. Yuan, D. J. Brady, and A. K. Katsaggelos, "Snapshot compressive imaging: Theory, algorithms, and applications," IEEE Signal Process. Mag. **38**(2), 65–88 (2021).
11. S. Jiao, M. Sun, Y. Gao, T. Lei, Z. Xie, and X. Yuan, "Motion estimation and quality enhancement for a single image in dynamic single-pixel imaging," Opt. Express **27**(9), 12841–12854 (2019).
12. J. Wu, L. Hu, and J. Wang, "Fast tracking and imaging of a moving object with single-pixel imaging," Opt. Express **29**(26), 42589–42598 (2021).
13. L. Zhang, J. Ke, S. Chi, X. Hao, T. Yang, and D. Cheng, "High-resolution fast mid-wave infrared compressive imaging," Opt. Lett. **46**(10), 2469–2472 (2021).
14. D. Reddy, A. Veeraraghavan, and R. Chellappa, "P2c2: Programmable pixel compressive camera for high speed imaging," in *CVPR 2011*, (IEEE, 2011), pp. 329–336.
15. Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, and S. K. Nayar, "Video from a single coded exposure photograph using a learned over-complete dictionary," in *2011 International Conference on Computer Vision*, (IEEE, 2011), pp. 287–294.
16. J. M. Bioucas-Dias and M. A. Figueiredo, "A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration," IEEE Trans. on Image Process. **16**(12), 2992–3004 (2007).
17. J. Yang, X. Yuan, X. Liao, P. Llull, D. J. Brady, G. Sapiro, and L. Carin, "Video compressive sensing using gaussian mixture models," IEEE Trans. on Image Process. **23**(11), 4863–4878 (2014).
18. X. Yuan, "Generalized alternating projection based total variation minimization for compressive sensing," in *2016 IEEE International Conference on Image Processing (ICIP)*, (IEEE, 2016), pp. 2539–2543.
19. Y. Liu, X. Yuan, J. Suo, D. J. Brady, and Q. Dai, "Rank minimization for snapshot compressive imaging," IEEE Trans. Pattern Anal. Mach. Intell. **41**(12), 2990–3006 (2018).
20. M. Iliadis, L. Spinoulas, and A. K. Katsaggelos, "Deep fully-connected networks for video compressive sensing," Digital Signal Processing **72**, 9–18 (2018).
21. J. Ma, X.-Y. Liu, Z. Shou, and X. Yuan, "Deep tensor admm-net for snapshot compressive imaging," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (IEEE Computer Society, 2019), pp. 10223–10232.
22. M. Qiao, Z. Meng, J. Ma, and X. Yuan, "Deep learning for video compressive sensing," APL Photonics **5**(3), 030801 (2020).
23. Q. Zhou, J. Ke, and E. Y. Lam, "Dual-waveband temporal compressive imaging," in *Computational Optical Sensing and Imaging*, (Optical Society of America, 2019), pp. CTu2A–8.
24. J. Pont Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 davis challenge on video object segmentation, arXiv preprint arXiv:1704.00675 (2017).