NORTHWESTERN UNIVERSITY

Synergy of Physics and Learning-based Models in Computational Imaging and Display

A DISSERTATION

SUBMITTED TO THE COMMITTEE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Computer Science

By

Zihao Wang

EVANSTON, ILLINOIS

August 2020

ABSTRACT

Synergy of Physics and Learning-based Models
in Computational Imaging and Display

Zihao Wang

Computational imaging (CI) is a class of imaging systems that optimize both the opto-electronic hardware and computing software to achieve task-specific improvements. Machine/deep learning models have proven effective in drawing statistical priors from adequate datasets. Yet when designing computational models for CI problems, physics-based models derived from the image formation process (IFP) can be well incorporated into learning-based architectures. In this thesis, we propose a group of synergistic models (synergy between physics-based and learning-based models) and apply such models in several CI tasks. The core idea is to derive differentiable imaging models to approximate the IFP, enabling automatic differentiation and integration into learning-based models. We demonstrate two synergistic models with the use of differentiable imaging models. The first synergistic model combines a differentiable model with residual learning for high frame-rate video frame synthesis based on event cameras. The second one integrates a light transport model with an autoencoder for 3D holographic display design.

Additionally, we demonstrate two other synergistic strategies without differentiable imaging models. In solving privacy preserving action recognition task using coded aperture videos, we show that extracting motion features derived from the IFP can improve the performance of deep classifiers. In an on-chip holographic microscopy task, to achieve space-time super resolution, we use sparsely-coded bi-level dictionary for hologram super resolution followed by a phase retrieval algorithm for 3D localization.

Acknowledgements

Throughout the five years I have spent at Northwestern University, I had the pleasure working with multiple charismatic individuals, whose help and inputs were essential for the completion of my research tasks. Pursuing a Ph.D. degree proved to be much harder than I had initially expected. Overpassing the obstacles that came up would not have been possible without the help of my colleagues and academic advisors.

I would like to sincerely thank all members of my dissertation committee for their unlimited support. Each one of them has contributed in different ways for the completion of different parts of the work presented herein.

I express my sincerest gratitude to my academic advisor Professor Oliver Cossairt for trusting me and offering me the opportunity to pursue the doctoral degree at Northwestern. He has been a continuous source of motivation and inspiration and his innovative research ideas have introduced me to research topics I never knew existed. My research with him has allowed me to view the world in a different way and opened new pathways for future success in my career. He has also been very supportive in both a professional and personal level helping me in many difficult situations that arose during my studies.

I am grateful to Professor Aggelos Katsaggelos, who has been closely collaborating with the Computational Photography Laboratory (CPL) for years. He has helped me a lot with learning and understanding optimization and machine learning techniques which constituted a large part

of my research. Many of the papers we have published would not have been achieved without his constant instructive comments and numerous hours he spent in meetings.

I would like to thank Dr. Dikpal Reddy, Dr. Sing Bing Kang, Dr. Xiaokai Li and Professor Boxin Shi for accepting and hosting me as a research intern during my PhD career.

Dr. Reddy mentored me in 2017 at a start-up company called Light Labs Inc. in Palo Alto, CA, where I was exposed to several practical problems in industry and encouraged to leverage the knowledge I have learned to solve them.

In the summer of 2018, I had a wonderful research experience with Dr. Sing Bing Kang at Microsoft Research (MSR), where I was given the opportunity to investigate computational cameras from a privacy preserving perspective. I enjoyed that experience as I was learning new things everyday. I thank Dr. Francesco Pittaluga, who was also a research intern, for providing me rich tips in deep learning and image processing. I thank Dr. Vibhav Vineet for collaborating with me after I left MSR.

I was fortunate to have interned at Apple Inc. working on a project to develop the next generation AR displays. My mentor, Dr. Xiaokai Li, had provided useful industrial experience and advices in guiding me through the internship. I have also received various sources of help from many colleages, including Dr. Yu Wei, Dr. Hao (Ian) Chen (formerly intern at Apple), Dr. Lingshan Li (formerly intern at Apple), and Dr. Mehmet N. Ağaoğlu.

I would like to thank Professor Boxin Shi at Peking University for hosting me as a visiting research intern in Beijing and Shenzhen of China. It was an exciting experience working on event cameras, together with Peiqi Duan. The collaboration has continues even after I returned to Northwestern.

I would like to thank Professor Jack Tumblin for being part of my dissertation committee. Professor Tumblin is always creative and can come up with the wildest research ideas. I have known Professor Tumblin through a number of classes I attended at Northwestern and he has been a constant source of inspiration.

I would like to thank Dr. Nathan Matsuda for being part of my dissertation committee. He is a delightful individual who always tries to help and make you feel better with his positive attitude.

I would like to thank my collaborators, including Professor Lei Tian at Boston University, Professor Roarke Horstmeyer at Duke University, Donghun Ryu (formerly at California Institute of Technology) for providing various sources of help and support.

I would like to thank the lab members of CPL, including Dr. Marina Alterman, Dr. Xiang Huang, Dr. Kuan He, Fengqiang Li, Chia-kai Yeh, Sushobhan Ghosh, Weixin Jiang, Professor Florian Willowmitzer, Florian Schieffers, and Lionel Fisk, Prasan Shedligeri, Hamid Hasani. They have provided me with constant support and joy. I cannot thank more to their company.

Finally, I take great pleasure in acknowledging the support of my family and friends throughout my doctoral studies. They have been always lifting my spirits making my demanding work easier.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

Computational imaging (CI) is a class of imaging systems that optimize both the opto-electronic hardware and computing software to achieve task-specific improvement. A conceptual visualization is shown in Fig. 1.1. Generally, the imaging hardware includes the optical elements and electronic sensors to transform the real-world signals into digital forms. The raw data collected from the hardware system serves as the input for computational algorithms for executing pertinent tasks, which can be diversely defined. A CI system can not only be designed to produce high quality photographs and videos, but also to retrieve information and/or perform cognitive interpretation.

Computation has emerged as an integral role in the design of modern imaging and display systems. In essense, CI research has been largely focused on the study of image formation. Interestingly, a large volume of computational models have been proposed in solving existing as well as novel tasks. Very recently, we have witnessed a shifting in designing computational models, from physics-based towards learning-based. For instance, suppose we are tasked to

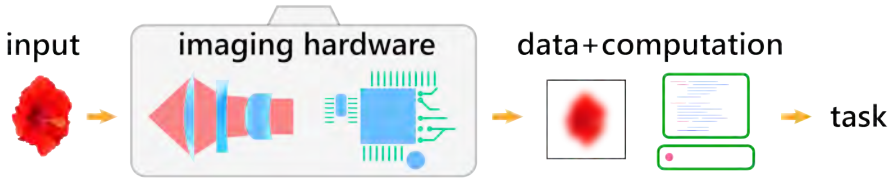


Figure 1.1. Computational imaging systems include the design of optical elements, electronic sensors, as well as computation. The raw data collected from the hardware system is leveraged by computation to achieve the final task.

solve the classic visual problem of image denoising. Researchers in 2007 [**9**] would take a close look at the in-camera image signal processing (ISP) pipeline and derive an analytical relationship for different sources of noise and the noisy image. The noise model, which contains a few parameters but are closely related to the physical imaging process, is then used to perform image denoising. As a comparison, recent image denoising models such as DnCNN in 2017 [**10**] do not have explicit modeling of the ISP pipeline, rather modeling noisy image and the residual image in an end-to-end fashion with many layers of convolutional filters. The weights of the filters are optimized by enormous amount of simulated image pairs. Such models have departed from ISP knowledge, yet are able to solve the image denoising task (additive Gaussian noise) with effectiveness and efficiency. Although both models are tasked to solve the same image denoising task, the two models have different design logic and architectures. In this thesis, we refer to the models that arrive at analytical solutions from the image formation process (IFP) as physics-based models. Meanwhile, we refer to the models that draw statistical priors from data as learning-based models. In terms of the number of parameters, physics-based models usually contain much fewer parameters than learning-based models. Learning-based models, however, due to their large number of parameters, require large amounts of high quality training datasets and may have generalization and domain adaptation issues while physics-based models do not. The goal of this thesis is to explore and develop synergistic models to take the union of the advantages from both the physics-based models and learning-based models.

One key idea being explored in this dissertation is to derive differentiable models for the IFP. Ene-to-end differentiable models have been the core feature of learning-based models as such models can be optimized by back propagation following the chain rule. By incorporating differentiable imaging models in conjunction with the already-differentiable learning-based models,

one can employ automatic differentiation techniques to propagate the loss information through both parts. Since part of the differentiable models is based on IFP, the relevant weights and parameters have physical counterparts/interpretation. The pipeline for this optimization strategy is shown in Fig. 1.2a. The loss error is propagated by the chain rule to update the weights ω in the imaging model $f(x, \omega)$. η stands for the learning rate. One benefit of such models is that part of the weights ω has physical counterparts. By optimizing the weights in the imaging model in an end-to-end fashion, one can make physical interpretation and fabrication. Such models are also task-specific. This is because the model weights are optimized by the annotated data x and y, and therefore, towards improving the corresponding task performance. Examples along this line are [**11, 12, 13, 14, 15, 16, 17**]. These works are common in that, although performing different tasks ranging from action recognition to depth estimation, they all have physical components such as phase plate pattern or pixel-wise exposure coding as part of the imaging models and learned by optimizing the corresponding task performance. A variant of the task-specific differentiable models is shown in Fig. 1.2b. In such types of models, the weights are pre-defined and fixed during training while the input is being updated at each epoch. This type of models can be interpreted as differentiable inverse solver designed for inverse problems. Examples along this line includes [**18, 19**].

1.1. Overview of this Dissertation

In this dissertation, we explore and demonstrate synergistic models. Such models can be viewed as the combination between physics-based models and learning-based models. We show that synergistic models have benefits in overcoming certain drawbacks from individuals. In the

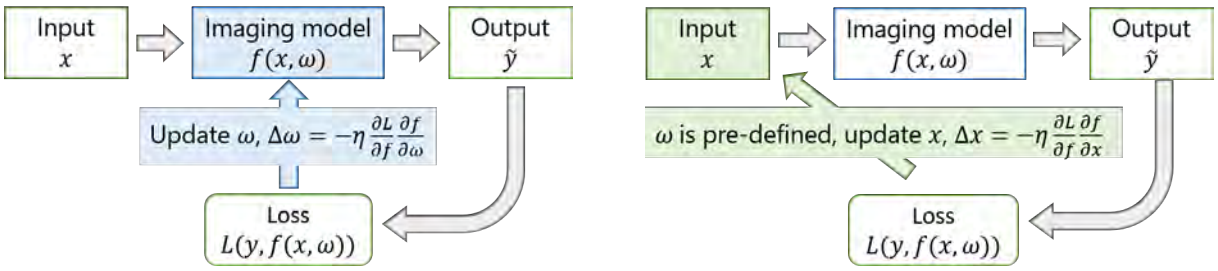


Figure 1.2. Task-specific differentiable imaging. The input signal x is passed through a differentiable imaging model (a) In the backpropagation step, the weights of the imaging model are being updated. (b) The weights are fixed while the input signal is updated.

meantime, synergistic models also have disadvantages. We will demonstrate several synergistic models applied in different imaging systems and tasks.

We demonstrate two synergistic models with the use of differentiable imaging models. The first synergistic model combines a differentiable model with residual learning for high frame-rate video frame synthesis. This is detailed in Chapter 2. The second one integrates a light transport model with an autoencoder for 3D holographic display design. This is detailed in Chapter 3.

Additionally, we demonstrate two other synergistic strategies without differentiable imaging models. Chapter 4 discussed the topic of holographic imaging. In an on-chip holographic microscopy task, to achieve space-time super resolution, we use dictionary learning for hologram super resolution followed by a phase retireval algorithm for 3D localization. In Chapter 5, we demonstrate the limit of deep classifiers in classifying unconventional image data, *i.e.*, lens-free coded aperture images. In solving privacy preserving action recognition task using coded aperture videos, we show that extracting motion features derived from the image formation can improve the performance of deep classifiers.

Chapter 6 demonstrates our recent development in the synergy between event cameras and RGB cameras. This has been a continued research of Chapter 2. Although not in the scope of synergistic models, we believe RGB-event vision is an interesting direction to inspire future works.

CHAPTER 2

Event-driven Video Frame Synthesis

In this chapter, we propose a synergistic model for solving the problem of high frame-rate video frame synthesis. We approach this problem by fusing a hybrid set of inputs, *i.e.*, a regular frame-rate intensity video and a stream of high speed neuromorphic events. We first analyze the temporal sensing process of the hybrid camera DAVIS, and propose a differentiable forward model that models the degeration from a latent high frame-rate video tensor to a low frame-rate frame observation tensor and event frame tensor. Our differentiable model enables iterative optimization of the latent video tensor via automatic differentiation, which propagates the gradients of a loss function defined on the observation data and updates the target video signal. This part of the model is referred as the physics-based model. Second, we concatenate the physics-based model with a residual network and develop a deep learning strategy to enhance the results from the first step, which we refer as a residual "denoising" process. Our trained "denoiser" is beyond Gaussian denoisers and shows properties such as contrast enhancement and motion awareness. We show that our framework is capable of handling challenging scenes including both fast motion and strong occlusions.

2.1. Introduction

Conventional video cameras capture intensity signals at fixed speed and output signals frame by frame. However, this capture convention is motion agnostic. When the motion in the scene is significantly faster than the capturing speed, the motion is usually under-sampled, resulting

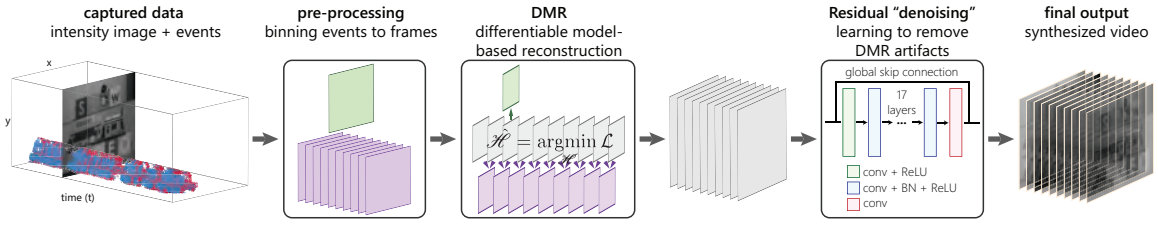


Figure 2.1. We propose a fusion framework of intensity image(s) and events for high frame-rate video synthesis. Our synthesis process includes a differentiable model-based reconstruction and a residual "denoising" process.

in motion blur or large discrepancies between consecutive frames, depending on the shutter speed (exposure time). One direct solution to capture fast motion is to use high speed cameras, in exchange with increased hardware complexity, degraded spatial resolution and/or reduced signal-to-noise ratio. Moreover, high speed moments usually happen instantaneously in-between regular-speed context. Consequently, either we end up collecting long sequences of frames with a great amount of redundancy, or the high-speed moment is missed before we realize to turn on the "slow-motion" mode.

We argue that high speed motion can be acquired and synthesized effectively by augmenting a regular-speed camera with a bio-inspired event camera [**20, 21**]. Compared to conventional frame-based sensors, event pixels *independently* detect logarithmic brighness variation over time and output "events" with four attributes: 2D pixel address, polarity (*e.g.*, "1": brightness increase; "0": brightness decrease) and timestamp ($\sim 10\mu s$ latency). This new sensing modality has salient advantages over frame-based cameras: 1) the asynchronism of event pixels results in sub-millisecond temporal resolution, much higher than regular-speed cameras ($\sim$ 30 FPS); 2) since each pixel responds only to intensity changes, the temporal redundancy and power consumption can be significantly reduced; 3) sensing intensity changes in logarithmic scale

enlarges dynamic range to over 120 dB[1]. However, event-based cameras have increased noise level over low frame-rate cameras. And the bipolar form of output does not represent the exact temporal gradients, introducing challenges for high frame-rate video reconstruction from event-based cameras alone.

In this chapter, we propose a high frame-rate video synthesis framework using a combination of regular-speed intensity frame(s) and neighboring event streams, as shown in Fig. 2.1. Compared to intensity-only or event-only TVFS algorithms, our work takes advantages from both ends, *i.e.*, high-speed information from events and high contrast spatial features from intensity frame(s). Our contributions are listed below:

(1) We introduce a differentiable fusion model which is able to model various temporal settings. We consider three fundamental cases, *i.e.*, interpolation, prediction and motion deblur, which can serve as building blocks for other complex settings. The problem can be solved by automatic differentiation that does not involve training. We refer to this process as Differentiable Model-based Reconstruction (DMR).

(2) We introduce a novel event binning strategy and compare it against conventional stacking-based binning strategy [**22, 7, 8, 23**]. Our binning preserves the temporal information of events necessary for high frame-rate video reconstruction. Additionally, we perform statistical evaluation for our binning strategy on the existing dataset [**24**].

(3) We introduce a deep learning strategy for further improving the DMR results. We model the DMR artifacts as additive "noise" and perform "denoising" via deep residual learning. During training, we augment the samples by randomizing *all* the parameters of the DMR. We show preliminary results that the trained residual denoiser (RD) has

[1]Typical dynamic range of a conventional camera is 90 dB

properties including contrast enhancement and motion awareness, which is beyond a Gaussian denoiser.

2.1.1. Related work

Multimodal sensor fusion Fusion among different types of sensing modalities for improved quality and functionality is an interesting topic. A related problem to ours is to spatially upsample functional sensors, *e.g.*, depth or hyperspectral sensors, with a high resolution guide image. The fusion problem can be formulated as joint image filtering via bilateral [**25**], multi-lateral filters [**26**] or Convolutional Neural Network (CNN) based approach [**27**]. For high speed video sensing, a fusion strategy can be employed between high speed video cameras (low spatial resolution) and high spatial resolution still cameras (low speed) [**28, 29, 30, 31, 32**].

Our paper investigates the temporal upsampling problem. While previous approaches investigate in the framework of compressive sensing [**33, 34, 35, 36, 37, 38, 39**], we formulate our work as fusing event streams with intensity images to obtain a temporally dense video. Compared to existing literature [**1**] which integrates events per pixel across time, our differentiable model utilizes "tanh" functions as event activation units and imposes sparsity constraints on both spatial and temporal domain.

Event-based image and video reconstruction Converting event streams (binary) to multiple-valued intensity frames is a challenging task, yet has been shown beneficial to downstream visual tasks [**8**]. Existing strategies for image reconstruction include dictionary learning [**7**], manifold regularization [**40**], optical flow [**22**], exponential integration [**2, 1**], conditional Generative Adversarial Networks (GAN) [**41**] and Recurrent Neural Network (RNN) [**8**]. Compared

to existing algorithms, our work unifies different temporal frame synthesis settings, including interpolation, extrapolation (prediction) and motion deblur (reconstructing a video from a motion-blurred image).

Non-event-based video frame synthesis 1) Interpolation: Early work on video frame interpolation has focused on establishing block-wise [**42**] and/or pixel-wise [**43, 44**] correspondences between available frames. Improved performance has been achieved via coarse-to-fine estimation [**45**], texture decomposition [**46**], and deep neural networks (DNN) [**47**]. Recent DNN-based approaches include deep voxel flow [**48**], separable convolution [**3**], flow computation and interpolation CNN [**49**]. 2) Prediction: Recent work on future frame prediction has proposed to use adversarial nets [**50**], temporal consistency losses [**51**] and layered cross convolution networks [**52**]. 3) Motion deblur: Recent work on resolving a sharp video/image from blurry image(s) has leveraged adversarial loss [**53**], gated fusion network [**54**], ordering-invariant loss [**55**], *etc.*.

2.2. Approach

2.2.1. Image formation

Assume there exists a high frame-rate video denoted by tensor $\mathscr{H} \in \mathbb{R}^{h \times w \times d}$, $d > 1$[2]. The forward sensing process results in two observational tensors, *i.e.*, the intensity frame tensor $\mathscr{F}$ and event frame tensor $\mathscr{E}$. Our goal is to recover tensor $\mathscr{H}$ based on the observation of intensity and event data.

[2] $\mathscr{H}$ is indexed on time axis starting from 1. Color channel is omitted here.

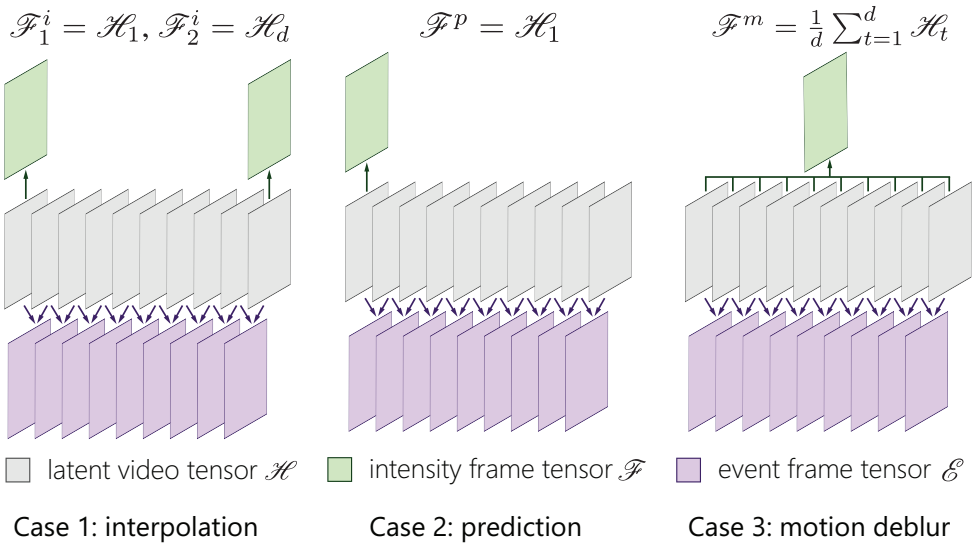


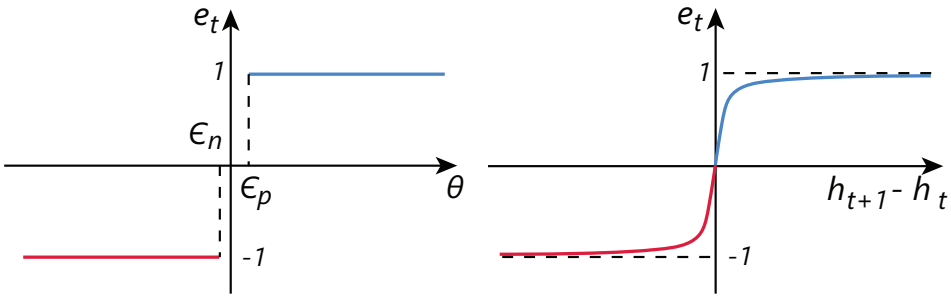
Figure 2.2. Forward models considered in this chapter. Case 1: interpolation from two observed intensity frames and event frames. Case 2: prediction from one observed intensity frame at the beginning and event frames. Case 3: Motion video from a single observed intensity frame and event frames.

Intensity frame tensor. We consider three sensing cases, *i.e.* 1) interpolation from the first and last frames of $\mathscr{H}$; 2) prediction based on the first frame of $\mathscr{H}$ and 3) motion deblur, in which case the intensity tensor is the summation over time. This can be visualized in Fig. 2.2.

Event frame tensor. As previously introduced, a pixel fires a binary output/event if the log-intensity changes beyond a threshold (positive or negative). This thresholding model can be viewed in Fig. 2.3a. Mathematically, the event firing process can be expressed as,

$$e_t = \begin{cases} 1 & \theta > \epsilon_p \\ -1 & \theta < -\epsilon_n \\ 0 & otherwise \end{cases}, \tag{2.1}$$

where $\theta = \log(I_t + b) - \log(I_0 + b)$. If $e_t = 0$, no events are generated. In order to approximate this event firing process, we model each event frame as a function of the adjacent frames from



(a) the event firing process (b) our proposed activation unit

Figure 2.3. Comparison of the event firing process and our proposed differentiable model. h_t denotes a pixel of $\mathscr{H}_t$.

the high frame-rate tensor $\mathscr{H}$, *i.e.*,

$$\mathscr{E}_t = \tanh\Big\{\alpha[\mathscr{H}_{t+1} - \mathscr{H}_t]\Big\}, \tag{2.2}$$

where α is a tuning parameter to adjust the slope of the activation curve. This function can be viewed in Fig. 2.3b. Based on this formulation, a video tensor with d temporal frames correspond to $d-1$ event frames.

2.2.2. Differentiable model-based reconstruction

The DMR is performed by minimizing a weighted combination of several loss functions. The objective function is formed as,

$$\hat{\mathscr{H}} = \underset{\mathscr{H}}{\operatorname{argmin}}\, \mathcal{L}_{pix}(\mathscr{H}, \mathcal{F}, \mathcal{E}) + \mathcal{L}_{TV}(\mathscr{H}) \tag{2.3}$$

Pixel loss. The pixel loss includes per-pixel difference loss against intensity and event pixels in ℓ_1 norm, *i.e.*,

$$\begin{aligned}\mathcal{L}_{pix}(\mathscr{H}, \mathcal{F}, \mathcal{E}) = \mathbb{E}_{fpix}[\|\mathcal{F} - \mathcal{A}(\mathscr{H})\|_1] \\ + \lambda_e \mathbb{E}_{epix}[\|\mathcal{E} - \mathcal{B}(\mathscr{H})\|_1],\end{aligned} \tag{2.4}$$

over the entire available data range. $\mathcal{F}$ and $\mathcal{E}$ denote the captured intensity and event data, respectively. $\mathcal{A}$ and $\mathcal{B}$ denote the forward sensing models described in Fig. 2.2 and Eq. 2.2. $\mathbb{E}_x$ represents expectation with respect to the observed pixels/events.

Sparsity loss. We employ total variation (TV) sparsity in the spatial and temporal dimensions of the high-res tensor $\mathscr{H}$. The TV sparsity loss is defined as:

$$\mathcal{L}_{TV}(\mathscr{H}) = \lambda_{xy} \mathbb{E}_{hpix}\left[\left\|\dot{\mathscr{H}}_{xy}\right\|_1\right] + \lambda_t \mathbb{E}_{hpix}\left[\left\|\dot{\mathscr{H}}_t\right\|_1\right], \tag{2.5}$$

where $\dot{\mathscr{H}}_{xy} = \frac{\partial \mathscr{H}}{\partial x} + \frac{\partial \mathscr{H}}{\partial y}$ and $\dot{\mathscr{H}}_t = \frac{\partial \mathscr{H}}{\partial t}$. We later denote $\mathcal{L}_{TV_{xy}} = \mathbb{E}_{hpix}[\left\|\dot{\mathscr{H}}_{xy}\right\|_1]$ and $\mathcal{L}_{TV_t} = \mathbb{E}_{hpix}[\left\|\dot{\mathscr{H}}_t\right\|_1]$. $\mathcal{L}_{TV_{xy}}$ can be viewed as a denoising term for intensity tensor, and $\mathcal{L}_{TV_t}$ can be viewed as an event denoising term. A comparison of the performance for each loss function is shown in Fig. 2.4. The figure shows a synthetic case for single-frame interpolation. We use three frames, resulting in two event frames (Eq. 2.1). Combining the spatial and temporal TV losses resullts in better performance.

Implementation. We use stochastic gradient descent to optimize Eq. 2.3 so as to reconstruct the latent high-res tensor. Our algorithm is implemented in TensorFlow. We use Adam optimizer. The learning rate varies depending on the tensor size as well as related parameters. Empirically, we recommend 0.002 as initial value. We recommend to schedule the learning rate to decrease $5\times$ every 200 epochs. The momenta $\beta_1 = 0.9, \beta_2 = 0.99$. For the case of

(a) $\mathcal{L}_{pix}$ (b) $\mathcal{L}_{pix} + \lambda_t \mathcal{L}_{TV_t}$ (c) $\mathcal{L}_{pix} + \lambda_{xy} \mathcal{L}_{TV_{xy}}$ (d) $\mathcal{L}_{pix} + \mathcal{L}_{TV}$ (e) ground truth

Figure 2.4. Comparison of different loss functions (simulated single-frame interpolation). $\mathcal{L}_{TV} = \lambda_t \mathcal{L}_{TV_t} + \lambda_{xy} \mathcal{L}_{TV_{xy}}$.

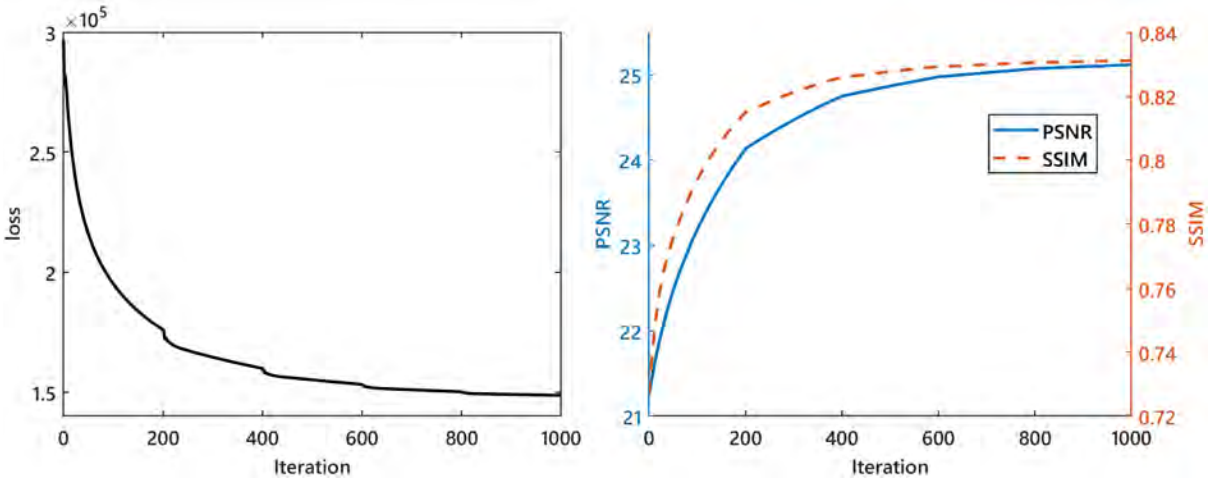


Figure 2.5. Loss values and accuracy (PSNR and SSIM) during DMR optimization.

interpolation, we initialize the high-res tensor $\mathscr{H}$ by linearly blending the two available low-res frames. For prediction and motion deblur, we initialize the high-res tensor using the available single low-res frame. An example of the optimization progress can be viewed in Fig. 2.5. As the loss decreases, both PSNR and SSIM increase and gradually converge.

2.2.3. Binning events into event frames

Our event sensing model requires binning events into frames. The ideal binning strategy would be "one frame per event". However, this binning strategy is unnecessarily expensive. For

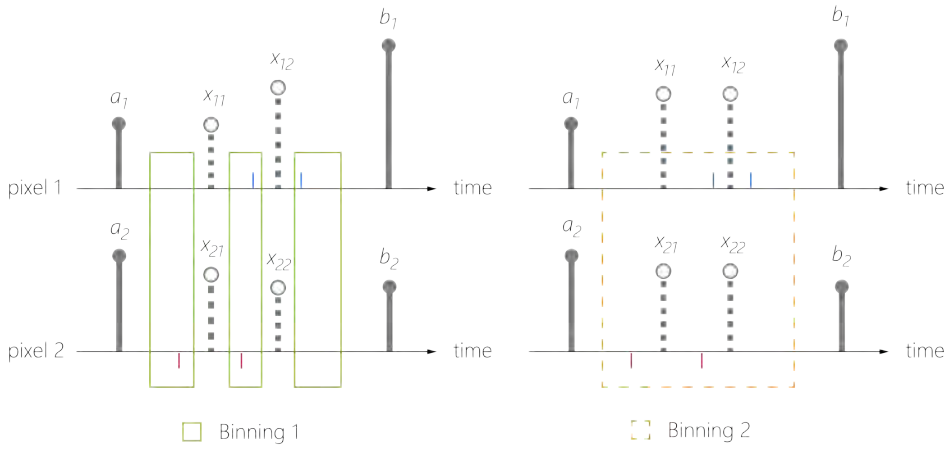


Figure 2.6. Comparison of two binning strategies.

example, the events between two consecutive frames (22 FPS in [**24**]) may vary from thousands to tens of thousands, resulting in computational challenges and redundancy. However, events happening at different locations but at very close timestamps can be processed in the same event frame. Therefore, we design and compare two binning strategies:

Binning 1 (proposed): For an incoming event, if its spatial location already has an event in the current event frame, then cast it into a new event frame; otherwise, this incoming event will stay in the current event frame. In this case, each event frame should only have three values, *i.e.*, {-1, 0, 1}.

Binning 2: Similar to several previous work [**22, 7, 8, 41**], where events are stacked/integrated over a time window, we allow each event frame to have more than three values. However, since the "tanh" function in Eq. 2.2 only outputs values between -1 and 1, we modify our event sensing model to have a summation operation over several sub-event frames. Mathematically, $\mathscr{E}_{b2} = \sum_t \mathscr{E}_t$. We use a toy example to analyze the performance of the two binning strategies, shown in Fig. 2.6. Assume there are two intensity pixels at different locations. During a certain

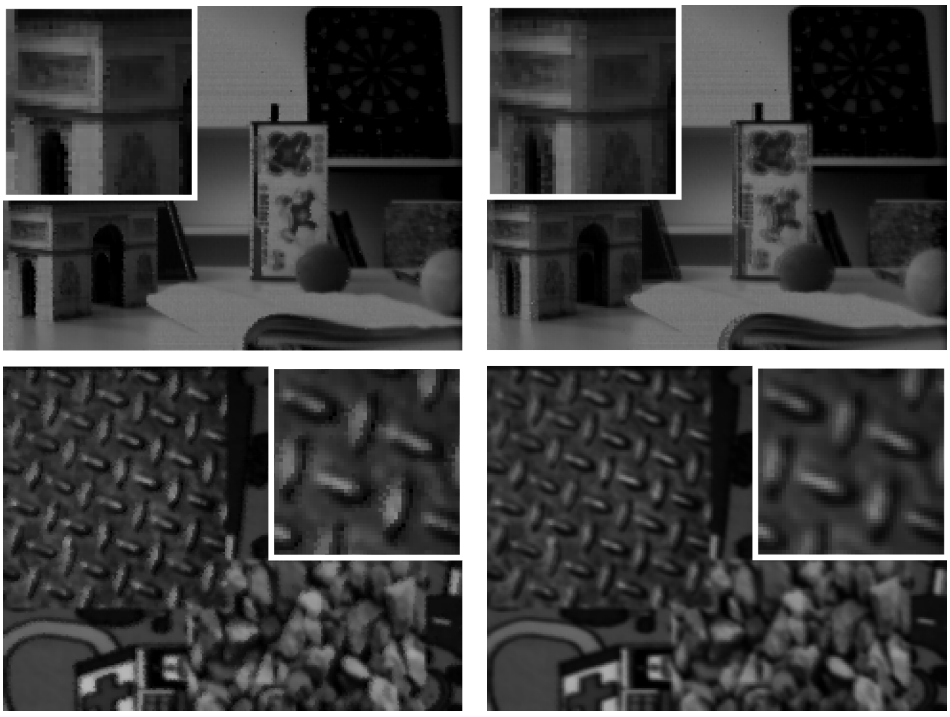
amount of time, each intensity pixel outputs two intensity values, *i.e.*, a_1 and b_1 from Pixel 1 and a_2 and b_2 from Pixel 2. Assume in the same time window four events are fired from two event pixels. (Assume the locations of event pixels and intensity pixels match perfectly.) According to Binning 1, the four events result in 3 event frames. Therefore, two intermediate frames $[x_{11}, x_{21}]$ and $[x_{12}, x_{22}]$ can be interpolated accordingly. Binning 1 makes sufficient use of the temporal order of events, resulting in 6 constraints:

$$\begin{cases} \sigma(x_{11} - a_1) = 0 \\ \sigma(x_{12} - x_{11}) = 1 \\ \sigma(b_1 - x_{12}) = 1 \\ \sigma(x_{21} - a_2) = -1 \\ \sigma(x_{22} - x_{21}) = -1 \\ \sigma(b_2 - x_{22}) = 0, \end{cases} \tag{2.6}$$

where we use $\sigma(\cdot)$ to denote the event sensing model $tanh\{\alpha(\cdot)\}$. Binning 2 integrates sub-event frames together. Therefore, it does not preserve the temporal order of events, resulting in ambiguity. In Eq. 2.7, the first equation has at least three solutions, *i.e.* $\{0, 1, 1\}$, $\{1, 0, 1\}$, $\{1, 1, 0\}$ corresponding to each "tanh" function respectively. This ambiguity is challenging to be solved by stochastic gradient descent.

$$\begin{cases} \sigma(x_{11} - a_1) + \sigma(x_{12} - x_{11}) + \sigma(b_1 - x_{12}) = 2 \\ \sigma(x_{21} - a_2) + \sigma(x_{22} - x_{21}) + \sigma(b_2 - x_{22}) = -2 \end{cases} \tag{2.7}$$

We show DMR results for a frame interpolation case using DAVIS dataset [**24**] in Fig. 2.7. We



(a) Binning 1 (b) Binning 2

Figure 2.7. Comparison of two binning strategies applied to frame interpolation using the DAVIS dataset.

use two consecutive intensity frames and the events in-between. In Row 1 ("slider_depth"), 9 event frames are binned from over 7, 700 events using Binning 1. Row 2 ("simulation_3_planes") has 19 event frames from over 40, 000 events. For Binning 2, we match the sub-event frame number with Binning 1 so as to compare the performance. Frame #2 is shown. Our results show that Binning 1 preserves sharp spatial structures. We will use Binning 1 in subsequent experiments.

2.3. Statistics on real event streams (Binning 1)

We examine several event streams captured in real scenarios using our Binning 1 strategy. The results are shown in Fig. 2.8. We plot three metrics: 1) event density, defined as (# of

events) / (full resolution) $\times$ 100% per event frame; 2) event speed, defined as (event density) / (event frame duration); and 3) event frame duration, defined as the elapsed time from the first to the last event in the same frame. We observe that the event frame duration results in less variation than the event density and speed. An empirical mean of the event frame duration is $\sim$ 2ms, corresponding to $\sim$ 500FPS and $\sim$ 16$\times$ temporal upsampling from 30FPS (regular frame rate).

2.3.1. Learning a residual denoiser

DMR is an iterative reconstruction approach based on a differentiable model, which does not involve training. The benefit of DMR is that it can handle a variety of fusion settings (interpolation, prediction, deblur, *etc.*) and is independent of optimizers. Although DMR does not involve training, it requires case-specific parameter tuning. Moreover, we observe that the DMR results may have visual artifacts. This is due to the ill-posedness of the fusion problem and different noise levels between the two sensing modalities.

In order to address these issues, we model the artifacts outcome of DMR as additive "noise" and propose a "denoising" process to remove the artifacts. Inspired by ResNet [**56**] and DnCNN [**10**], we employ the residual learning scheme and train a residual denoiser (RD). Rather than training the denoiser from various levels of artificial noise, we design to train the network from the outcome of DMR. Mathematically, the residual $\mathscr{R}$ is expressed as,

$$\mathscr{R} = \hat{\mathscr{H}} - \mathscr{H}_g, \tag{2.8}$$

where $\hat{\mathscr{H}}$ represents the reconstructed frame from DMR, and $\mathscr{H}_g$ represents the ground truth frame. We use a residual block similar to [**57**], which has a $\{$conv + ReLU$\}$ and a $\{$conv$\}$ layer

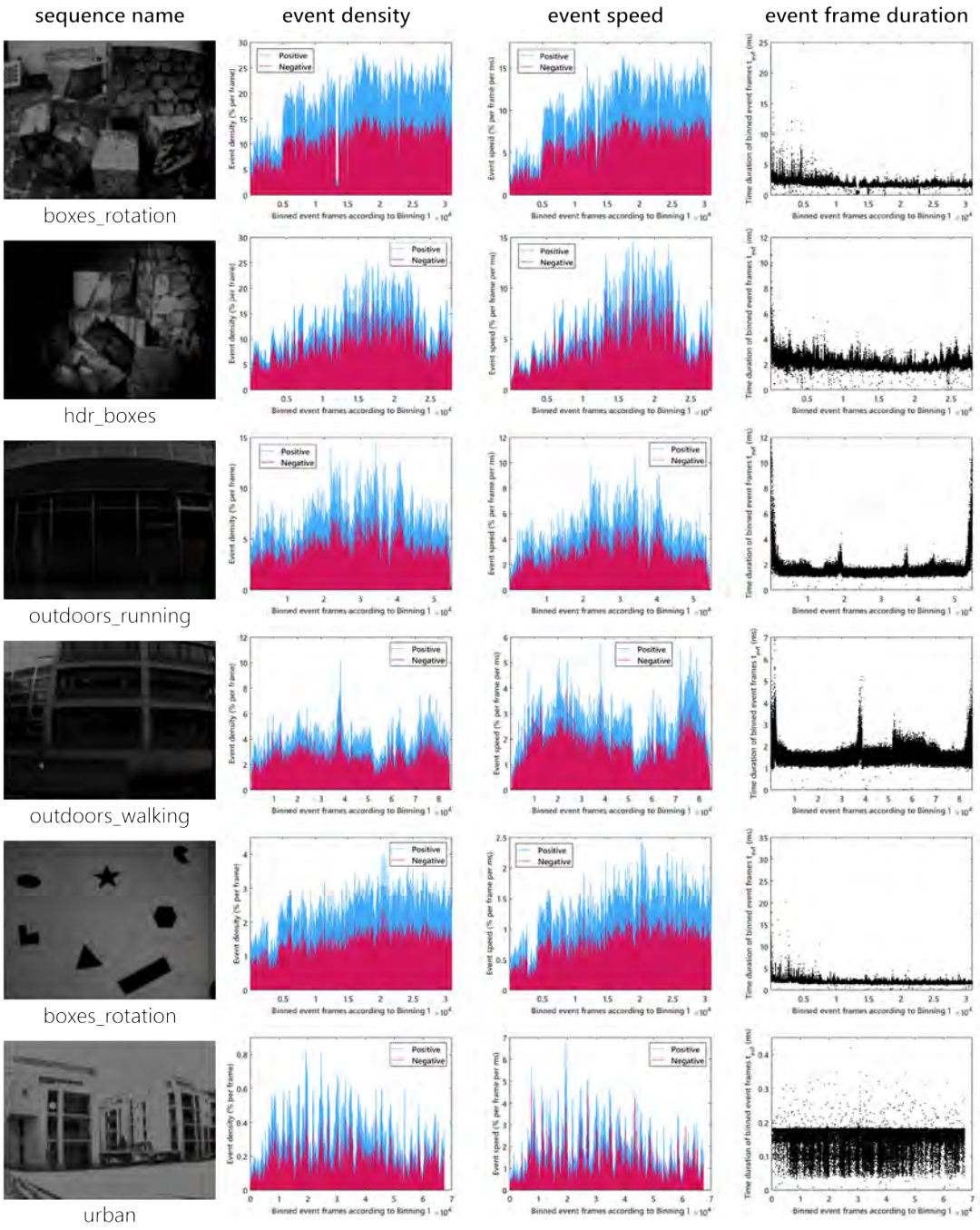


Figure 2.8. Statistics for using Binning 1 on real event streams.

Table 2.1. Augmentation recipe

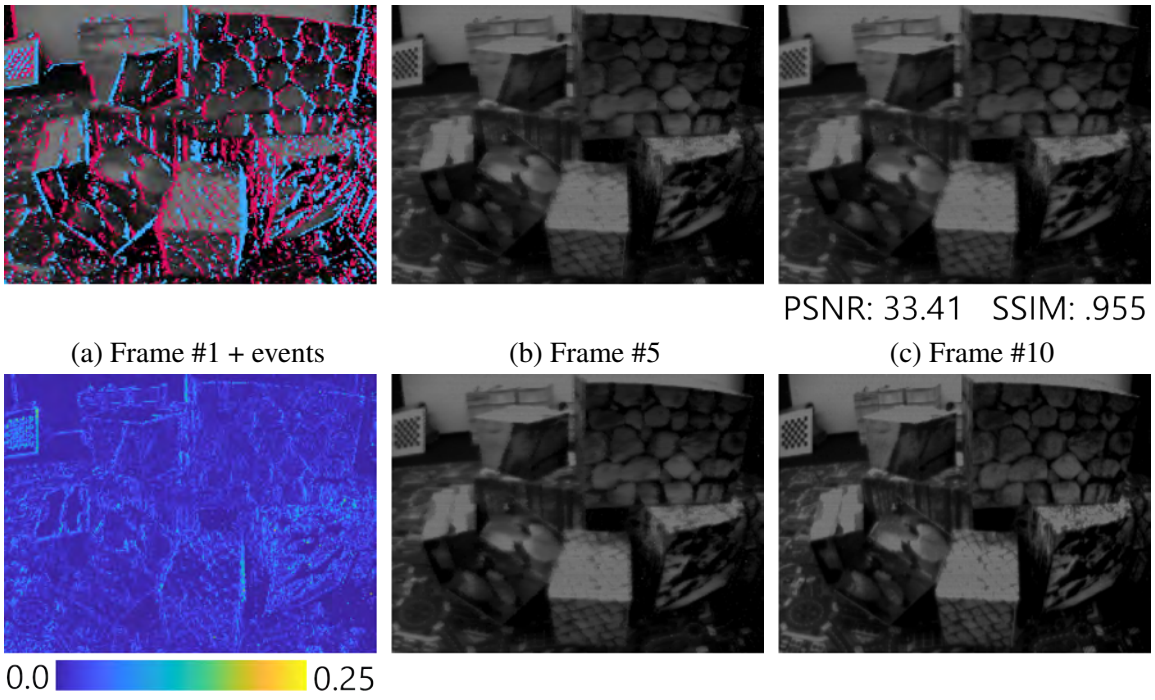
source	notation	value range
Eq. (2.1)	ϵ_p, ϵ_n	(0, 0.05)
Eq. (2.2)	α	(8, 20)
Eq. (2.4)	λ_e	(0.1, 0.5)
Eq. (2.5)	λ_{xy}	(0.3, 0.8)
Eq. (2.5)	λ_t	(0.2, 0.6)
	event percentage	(0%, 20%)
	PBR learning rate	(0.001, 0.009)
	PBR epoch(s)	(1, 350)

at the beginning and end, with 17 intermediate layers of $\{$conv + BN + ReLU$\}$. The kernel size is 3×3 with stride of 1. The loss function for our denoiser is the mean squared error of $\hat{\mathscr{H}}$ and $\mathscr{R}$. During training, we augment data by randomizing the configuration parameters (including the running epochs) in DMR, summarized in Table 2.1. The goal of this augmentation is 1) to prevent overfitting; 2) to enforce learning of our DMR process; 3) to alleviate effects due to non-optimal parameter tunning. Our denoiser is single-frame, as we seek to enhance each DMR output frame iteratively *without* comprimising the variety of DMR fusion settings.

2.4. Experiment results

We design several experiments to show the effectiveness of our framework.

- For DMR, we evaluate the three cases (interpolation, prediction and motion deblur) described in Fig. 2.2 on the DAVIS dataset [**24**], and compare against state-of-the-art event-based algorithms, *i.e.*, Complementary Filter [**1**] and Event-based Double Integral [**2**].
- For RD, we first discuss how to use the trained RD. We compare two strategies, *i.e.*, 1) to use RD after every DMR iteration; 2) to use RD only when DMR is converged. We



(a) Frame #1 + events (b) Frame #5 (c) Frame #10

(d) Error map #10 (e) Frame #16 (f) Frame #21

Figure 2.9. Frame interpolation. The start and end frames, as well as in-between events, are used as input. Frame #10 is compared against the ground truth middle frame.

then evaluate the effectiveness of trained RD by comparing with Gaussian denoisers, *e.g.*, DnCNN [**10**] and FFDNet [**57**].

- Finally, we compare our results with a non-event-based frame interpolation algorithm, SepConv [**3**].

2.4.1. Results for DMR

Interpolation. We first show interpolation results in Fig. 2.9. We use three consecutive frames from [**24**], withholding the middle frame. The intermediate events bin into 20 event frames. The ground truth middle frame is the closest to Frame #10.

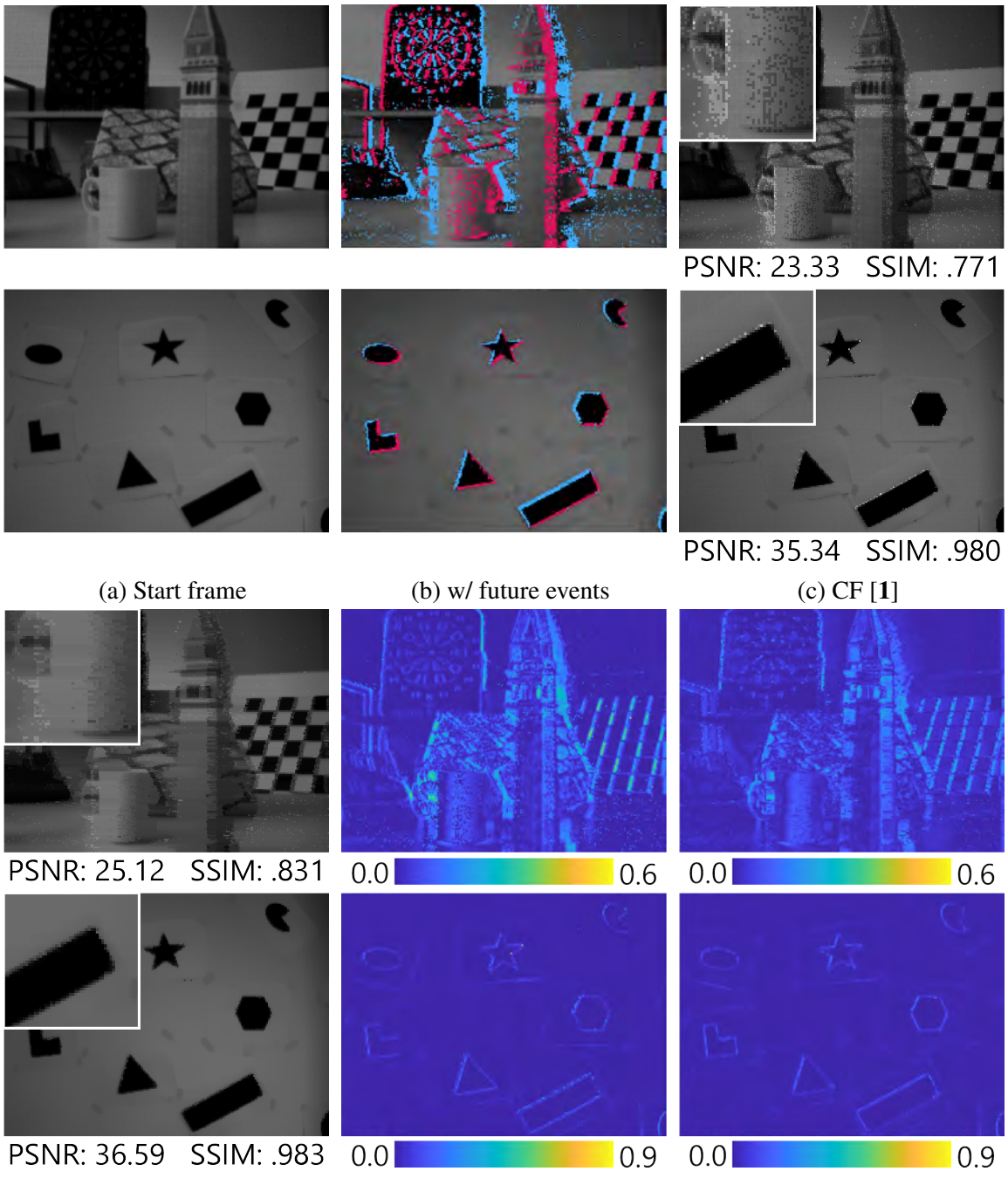
Prediction. We next show frame prediction results, corresponding to Case 2 in Fig. 2.2. We withhold the end frame of two consecutive frames and seek to predict it using the start frame and "future" events. The results are shown in Fig. 2.10. Compared to CF [**1**], our results are less noisy and closer to the ground truth.

Motion deblur. Corresponding to Case 3 in Fig. 2.2, we compare our DMR results with state-of-the-art, Event-based Double Integral (EDI) [**2**], shown in Fig. 6.16. Compared to EDI, our results preserves sharp edges while alleviating event noise.

2.4.2. Results for RD

Data preparation. We use publicly available high-speed (240 FPS) video dataset, the Need for Speed dataset [**58**]. The reason we choose this dataset is because it has rich motion categories and content (100 videos with 380K frames) which involves both camera and scene/object motion. As introduced in Subsection 2.3.1, our RD is trained on the output of DMR process. As a proof of concept, we simulate solving a single-frame prediction problem, *i.e.* given two consecutive video frames, we first simulate the latent event frame. Next, a DMR is performed to predict the end frame.

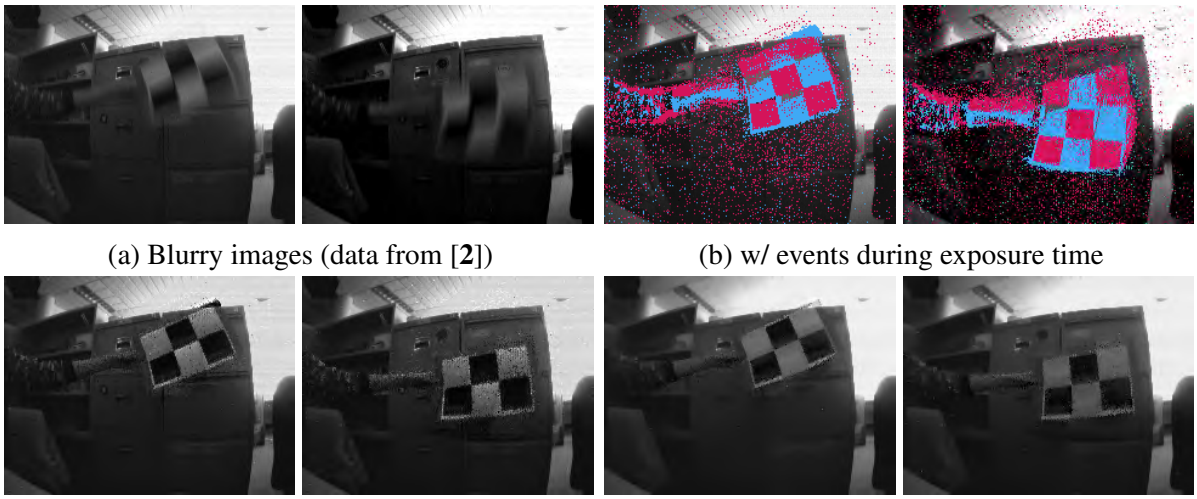
Training and testing. We randomly split the dataset into 89 training classes and 11 testing classes. For augmentation purpose, we perform a random temporal flip and a spatial crop with size 40×40. The sample clip will then experience event frame simulaltion and DMR using a random setting according to Table 2.1. Note that we enforce generated event frames to contain less than 20% of events. This is according to the statistical analysis of the DAVIS dataset we have performed in Fig. 2.8. We generate 100K image pairs of size 40×40 pixels; 80% of the



(a) Start frame (b) w/ future events (c) CF [**1**]

(d) Ours (DMR) (e) Error map of CF (f) Error map of ours

Figure 2.10. Frame prediction. Given a start frame (a) and the future events (b) happened after (a), we predict the end frame (ground truth omitted). Our results using DMR alone outperforms existing algorithm, Complementary Filters (CF) [**1**].



(a) Blurry images (data from [**2**])

(b) w/ events during exposure time

(c) EDI [**2**]

(d) Ours (DMR)

Figure 2.11. Motion deblur. A motion blurred image (a) and the events during exposure time (b) are used to reconstruct a high frame-rate video. Compare to (c) EDI [**2**], our results (d) preserves spatial features with less noise.

sample dataset are randomly chosen as training samples and the rest 20% are used for validation. We use a batch size of 128, which results in 2K batches per epoch. We use mini-batch stochastic gradient descent with an Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$). Note that the same optimizer has been used in the DMR process, but the optimization of DMR does not involve training. The learning rate is scheduled as 1×10^{-3} for the initial 30 epochs, then 1×10^{-4} for the following 30 epochs and 5×10^{-5} afterwards. We use an NVIDIA TITAN X GPU for parallelization. Each epoch takes approximately 6 minutes training on our machine. We train our network for 150 epochs. Since our model is fully convolutional, the number of parameters is independent of the image size. This enables us to train on small patches (40×40) and test on the whole image.

Plug & play *vs.* one-time denoising. Since we train our denoiser to establish a mapping function between DMR and its residual towards the ground truth, the first experiment we investigated is how/when to use this denoiser. We compare two frameworks, *i.e.*, the plug & play

Table 2.2. Plug & play *vs.* one-time denoising using RD.

clip name	plug &play	one-time
Motorcycle	28.07 / .951	**29.11/.965**
Car race	24.53 / .883	**24.89/.895**
Football Player	29.94 / .935	**32.30/.978**

[**59**] and the one-time denoising. The plug & play framework decouples the forward physical model and the denoising prior using the ADMM technique [**60**]. For one time denoising, we apply the residual denoiser once after the DMR has converged. One-time denoising is considered because it is considerably faster than plug & play. Our experimental results show that one-time denoising performs similar or even better than plug & play, shown in Table 2.2. We reason that this is related to our training process and the initialization of the high-res tensor. Our differentiable model involves a temporal transition process from an existing frame to a future frame. We initialize the high-res tensor with the reference intensity frame. In each DMR iteration, the reconstruction process produces artifacts that are similar to the degradations in the initialized image. However, our denoiser is trained to "recognize" this degradation and remove these artifacts. Therefore, our denoiser is most useful and efficient when applied after the DMR has converged. Examples are shown in Fig. 2.12.

Comparison with Gaussian denoisers. Since we decouple the problem as DMR and RD process, it is interesting to see whether a general denoiser can complete this task. We select several video clips from the testing classes and compare our results with two other denoisers, DnCNN [**10**] and FFDNet [**57**]. DnCNN is an end-to-end trainable deep CNN for image denoising with different Gaussian noise levels, *e.g.*, [0, 55]. During our testing of DnCNN we found that the pre-trained weights do not perform well. We retrained the network using the Need for Speed dataset with Gaussian noise. The FFDNet is a later variant of DnCNN with the inclusion of pre-

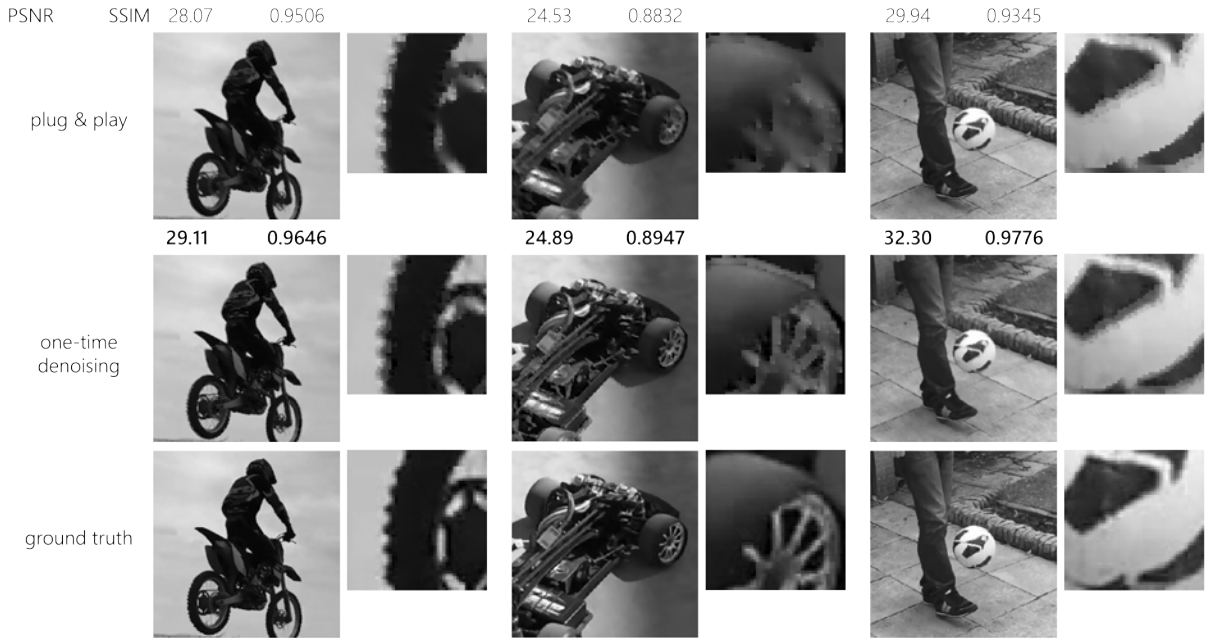


Figure 2.12. Plug & play vs. one-time denoising.

Table 2.3. Performance comparison for different denoisers.

clip name	metric	DMR	DnCNN	FFDNet	Ours
airplane	PSNR	30.91	31.10	30.92	**31.38**
	SSIM	0.975	0.982	0.976	**0.982**
basketball	PSNR	23.55	24.05	23.47	**24.06**
	SSIM	0.963	0.971	0.964	**0.972**
soccer	PSNR	29.96	31.08	30.13	**31.29**
	SSIM	0.961	0.974	0.962	**0.975**
billiard	PSNR	36.46	35.42	**36.48**	36.46
	SSIM	0.982	0.986	0.983	**0.987**
ping pong	PSNR	32.46	32.26	**32.50**	32.24
	SSIM	0.974	0.978	0.975	**0.979**

and post-processing. During our tuning of the FFDNet, we found that smaller noise levels (a tunable parameter for using the model) result in better denoising performance in terms of PSNR and SSIM metrics. For each testing image, we present the best tuned FFDNet result (noise level less than 10) and compare with our proposed denoiser. The results are summarized in Table 2.3.

(a) Ours (DMR) (b) DnCNN [**10**] (c) FFDNet [**57**] (d) Ours (RD) (e) ground truth

Figure 2.13. Comparison of denoising performance. Our learned Residual Denoiser (RD) reconstructs the intermediate frame (1-frame interpolation case) with fewer motion artifacts.

2.4.3. Comparison to non-event-based approach

We compared our results for performing multi-frame interpolation with a state-of-the-art approach, SepConv [**3**]. We present results comparing 3-frame interpolation in Fig. 2.14. We convert our grayscale testing images to 3 channels (RGB) before applying the SepConv interpolation algorithm. Although the results from SepConv provide better visual experience, they have salient artifacts around large motion regions. Note that performing intensity only frame interpolation produces significant artifacts in the presence of severe occlusions. On the other hand, our event-driven frame interpolation is able to successfully recover image details in occluded regions of interpolated frames. For a quantitive comparison, the SepConv method has an average SSIM of 0.9566 and PSNR of 29.79. Ours have average SSIM of 0.9741 and PSNR of 37.64.

2.5. Concluding remarks

In this paper, we have introduced a novel high frame-rate video synthesis framework by fusing intensity frames with event streams, taking advantages from both ends. Our framework includes two key steps, *i.e.*, DMR and RD. Our DMR is free of training and is capable to unify different fusion settings between the two sensing modalities, which was not considered in previous work such as [**2, 1**]. We have shown in real data that our DMR performs better than existing algorithms. However, DMR requires tuning parameters, which have large variance across various settings. This was one of the reasons we propose to train an RD. Our strategy is to incorporate a range of DMR parameter settings so as to expose the network with various DMR results, including both the optimal and non-optimal ones. By learning the corresponding residual, our simulation results have shown that a RD can be trained to effectively remove

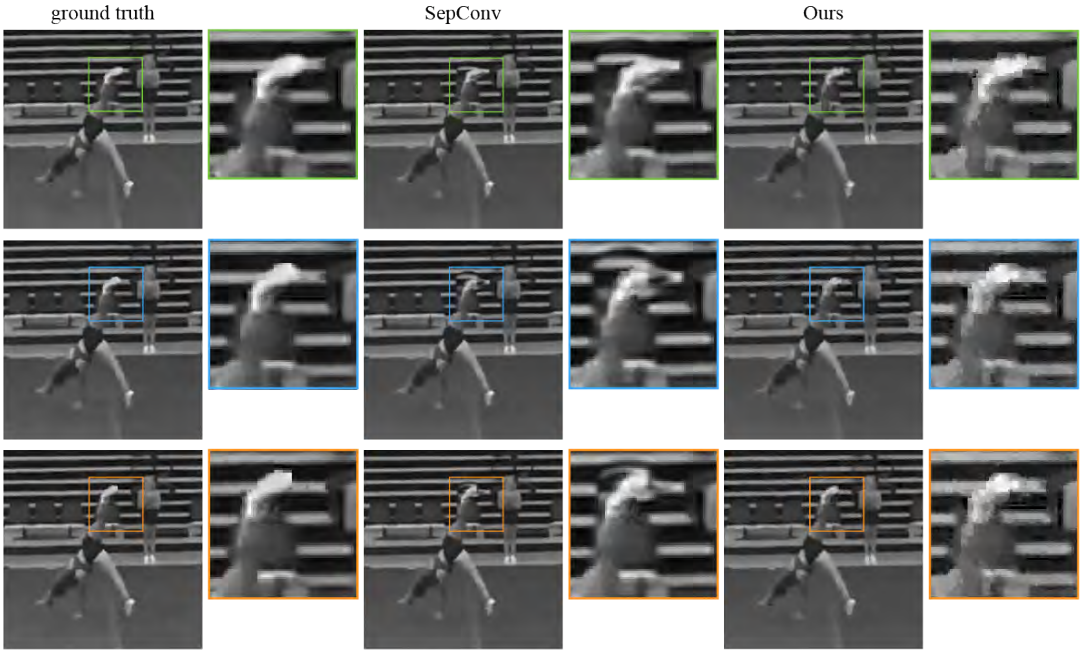


Figure 2.14. Multi-frame interpolation results, compared with SepConv [**3**]. Shown are frames #2, #3 and #4. Note that the intensity-only based frame interpolation method (SepConv) produces considerable motion artifacts around occluded areas, while our event-driven frame interpolation succesfully recovers image details in occluded regions.

artifacts from DMR. Currently we train an RD from single-frame prediction case. Yet it is interesting to further augment the training samples with *all* the cases, which we will investigate in the future. Applying the RD to real data faces a domain gap due to the resolution (both spatial and temporal) and noise level mismatch. Currently, none of the existing DAVIS datasets contains enough sharp intensity images captured at high speed for training/fine-tuning. We will investigate event simulation using event simulator [**61**] in our future work.

CHAPTER 3

Hogel basis display

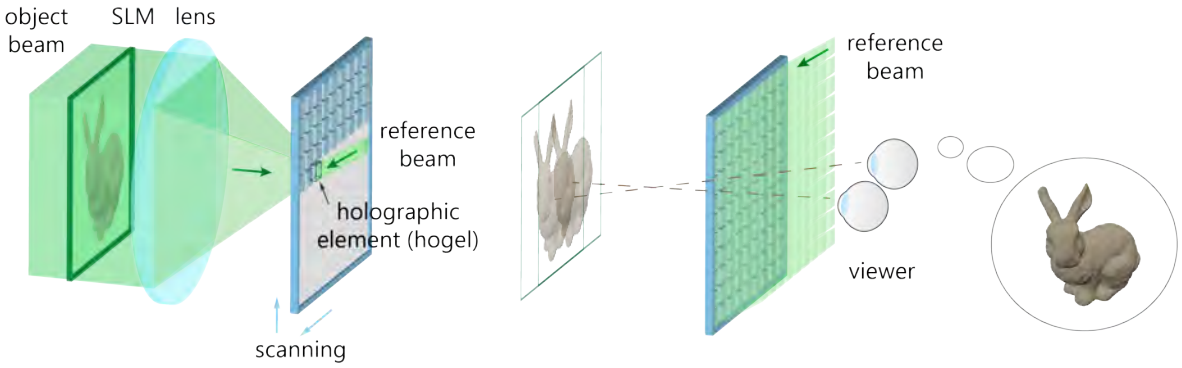
In this chapter, we continue the demonstration of synergistic modeling. We introduce a novel synergistic model for the design of 3D holographic display, namely hogel basis display (HBD). HBD is inspired by holographic stereograms (HS). HS a type of holographic display that produces high quality static imagery by recording parallax view images into an array of volume holographic pixels (hogels). HBD explores storage capability of volume holograms. The concept of hogel basis is introduced by grouping a set of hogels as super hogels (SH) and storing the compositing bases into each hogel, instead of complete images. A SH is able to display/synthesize complete viewpoint images. Programmability is added to the corresponding playback reference beam.

An autoencoder is designed for learning the hogel basis, which is essentially an image compression algorithm. The autoencoder is the learning part of our proposed synergistic model. To build the physics-based part, we numerically simulate the recording and playback processes of the hogel bases. We then integrate the differentiable hologram simulator in the autoencoder. The optimization process includes 1) learning basis decomposition of images, 2) numerically simulating the recording-playback effects for the bases. Therefore, the learned bases have potential to be recorded by hardware. Additionally, we show performance improvements for the synergistic model over the basic autoencoder. This is interesting as it indicates a regularization property of the physical model.

3.1. Introduction

Holographic stereograms. Our work is built upon the synthetic holographic stereogram (SHS) printing technique [**62, 63**], which incluses recording and playback processes of a series of volume hogels (holographic elements). The procedure can be viewed in Fig. 3.1a. The hologram plate is stationed with a scanning motor with 2-DoF mobility. At each hogel position, the object beam is illuminated through a spatial light modulator (SLM). The SLM is used to modulate the incident light beam such that a desired light field pattern is programmed. Each SHS position corresponds to a 2D perspective view of the 3D scene. The SLM pattern is then focused down to the hogel position (nearly a point). On the other side of the hogel projects a collimated reference beam. The reference beam can be projected with an oblique angle. The recording process has a time perioud for exposure, allowing the hologram material to undergo photochemical reactions such that its refractive index is proportional to the interference fringes [**64**]. In the playback process, the best viewing condition is to have the same reference beam illuminated onto the hologram with the same geometry. Since the recording reference beams do not have encodings or variations, it is also possible to use other light sources to view the stereogram. When viewing, the viewers need to see through the panel, as shown in 3.1b. Effectively each eye observes different scene perspectives that allows the viewers to have 3D viewing experience.

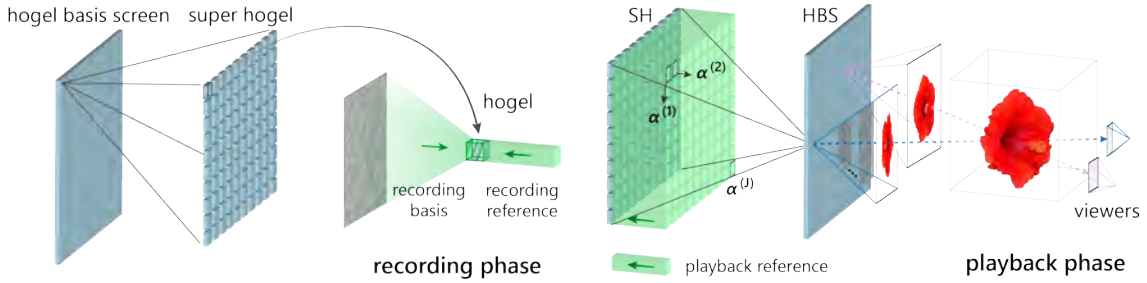
Overview of hogel basis display. The design of hogel basis display is based on SHS. The recording and playback processes are shown in Fig. 3.2. Instead of directly projecting a complete view image onto each hogel, we project an optimized basis. A few hogels are grouped together as a super hogel (SH). An SH is equavalent to a conventional hogel in the SHS setting. Therefore, during playback, the viewers should look through the hogel basis screen (HBS) as



(a) Recording a holographic stereogram.

(b) Viewing the holographic stereogram.

Figure 3.1. The recording and viewing of a holographic stereogram.



(a) Recording phase.

(b) Playback phase.

Figure 3.2. Overview of hogel basis display (hardware).

conventional SHS does. However, the playback reference is programmed by another SLM device (omitted in Fig. 3.2b). Each SLM pixel will program a single hogel with a complex value. For the m-the SH, the 2D view image is represented as:

$$I_{\mathrm{SH}_m} = |\sum_{j=1}^{J} \alpha^{(m,j)} u_b^{(j)}|^2, \tag{3.1}$$

where I_{SH_m} denotes the m-th SH image intensity, $\alpha^{(m,j)}$ denotes the j-th coefficients within the m-th SH and $u_b^{(j)}$ denotes the j-th basis image. Both α and u_b are complex-valued. The image I_{SH_m} is a complex linear representation of J bases weighted by the coefficients.

Here, we list a proposed quantification of HBS. Assume an SLM has 1000×1000 programmable pixels per frame. An SH consists of 10×10 hogels, totally 100 bases as well as programmable coefficients for each view. An HBS is therefore composed of 100×100 SH, equivalently 100×100 viewing perspective images. We use 100×100 SLM pattern to record one hogel. Therefore, during playback each hogel will project 100×100 directional rays (related to the playback optics). In sum, we use 1000×1000 programmable SLM pixels to program the playback reference beam, and is able to produce $100 \times 100 \times 100 \times 100$ directional rays. In the next section, we will describe the optimization for the hogel bases u_b.

3.2. Hogel basis autoencoder

We propose a deep neural network architecture named hogel basis autoencoder to optimize the bases. The designed neural network is shown in Fig. 3.3. This network has a deep encoder stage and a shallow autoencoder stage. The encoder stages takes in as input a single view of the light field and outputs the complex basis coefficients. The input image is of size 100x100 and the output of the encoder is a complex vector of length 100. This basis coefficient vector is the compressed representation of the input image. The decoder consists of the hogel bases in the form of network weights. There are 100 complex hogel bases each of size 100x100. The predicted coefficients from the encoder network are used to scale the hogel bases and summed coherently to produce the predicted image.

To train the network we generate a synthetic dataset that contains high angular resolution light fields. Both the spatial and angular resolution are set as 100x100. We use Pytorch3d[1], a differentiable OpenGL based renderer library to render the high dimensional light field images.

[1] https://pytorch3d.org/

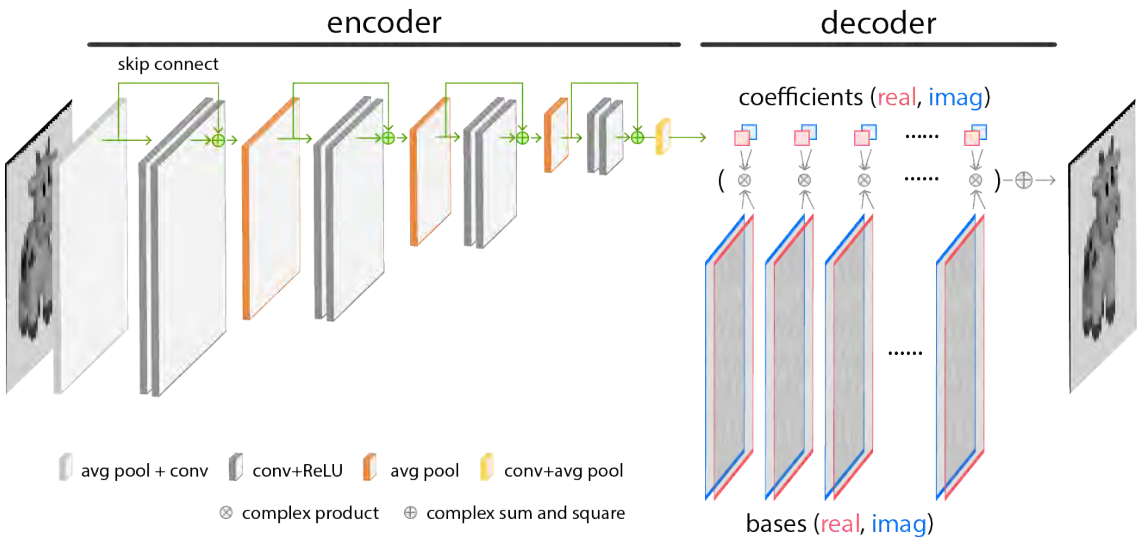


Figure 3.3. Hogel basis autoencoder (BaseNet).

We use these high dimensional light fields to learn the hogel bases which can effectively represent these images in a much lower dimension. Here, we have chosen a 100x lower dimension than the input space, thus achieving a 100x compression ratio. As shown in Fig. 3.3, the encoder consists of a series of convolutional layers with kernel size of 3×3 and strides of 1×1 and padding size of 1×1, as well as average pooling layer with filter size of 2×2. The final average pooling layer has two output branches, each leading to a vector of real/imaginary part of the coefficients. A 100×100 input will be encoded as a 100×2 coefficient tuple. The bases are initialized independently from the input images. The decoder takes the complex coefficients and the basis images and reconstructs the image by performing complex multiplication and summation. The final intensity values are regarded as the output image. The encoder network weights and the hogel bases are optimized over the entire light field training dataset to minimize the mean squared error.

In contrast to the hogel basis autoencoder, referred as BaseNet, we propose a physics-in-the-loop architecture to incorporate the physical effects, as shown in Fig. 3.4. This network, referred as HoloNet, has a different decoder as BaseNet. The original bases are first fed into a hologram simulator, which simulates the recording and playback processes for each complex-valued basis. The output bases are then combined with the coefficients to reconstruct the image. During training, both the coefficients and the bases are learnable variables. The hologram simulator is constructed as a differentiable model such that we can backpropagate the loss error to the original basis end.

In the next section, we will describe the hologram simulator.

3.3. Hologram simulator

In this section, we describe a simulation approach for volume holograms (VH) based on 3D convolution. The problem is to solve the light field distribution with VH as scattering media. Here, a VH is regarded as a hogel in the HBS. During the recording phase, a recording basis wave $u_b^{(j)}(x, y; z_b)$ and reference wave $u_r^{(j)}(x, y; z_r)$ are counter-propagating through the VH, where $j \in \{1, ..., J\}$, z_b and z_r denote the location on the z-axis respectively. We use $u_b^{(j)}(x, y)$ and $u_r^{(j)}(x, y)$ to denote the complex-valued 2D images located at two ends along z-axis of the VH. The reference wave is constrained as a ramp phase image, *i.e.*, $u_r^{(j)}(x, y) = \exp[i(k_{r,x}^{(j)}x + k_{r,y}^{(j)}y]$. We use $\mathbf{k}_r^{(j)} = [k_{r,x}^{(j)}, k_{r,y}^{(j)}, k_{r,z}^{(j)}]$ (with $|\mathbf{k}_r^{(j)}| = k_0 = \frac{2\pi n_0}{\lambda}$) to represent the reference wave vector. The light distribution in the VH satisfies the inhomogeneous Helmholtz equation:

$$(\nabla^2 + k_0^2)u(\mathbf{x}) = -v(\mathbf{x})u(\mathbf{x}), \tag{3.2}$$

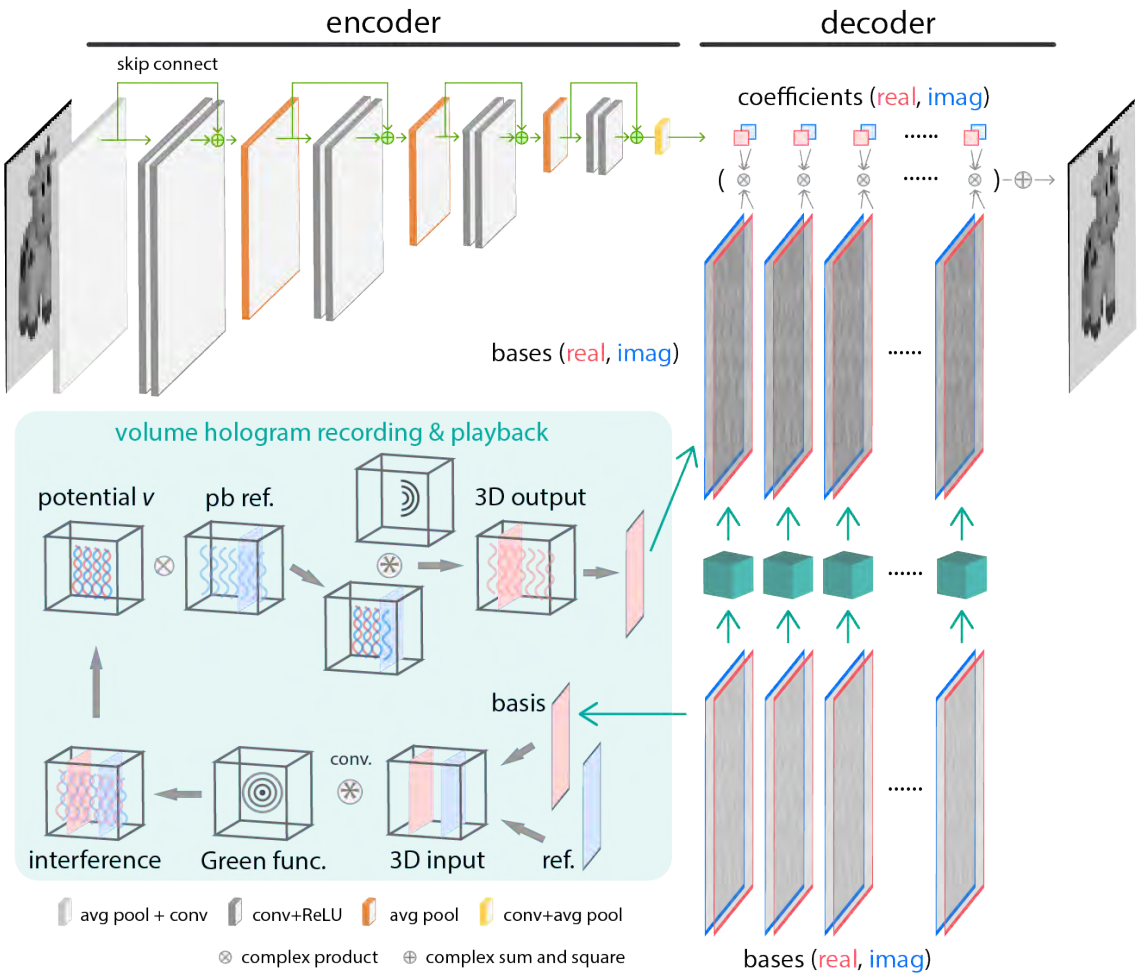


Figure 3.4. Physics-in-the-loop hogel basis autoencoder (HoloNet).

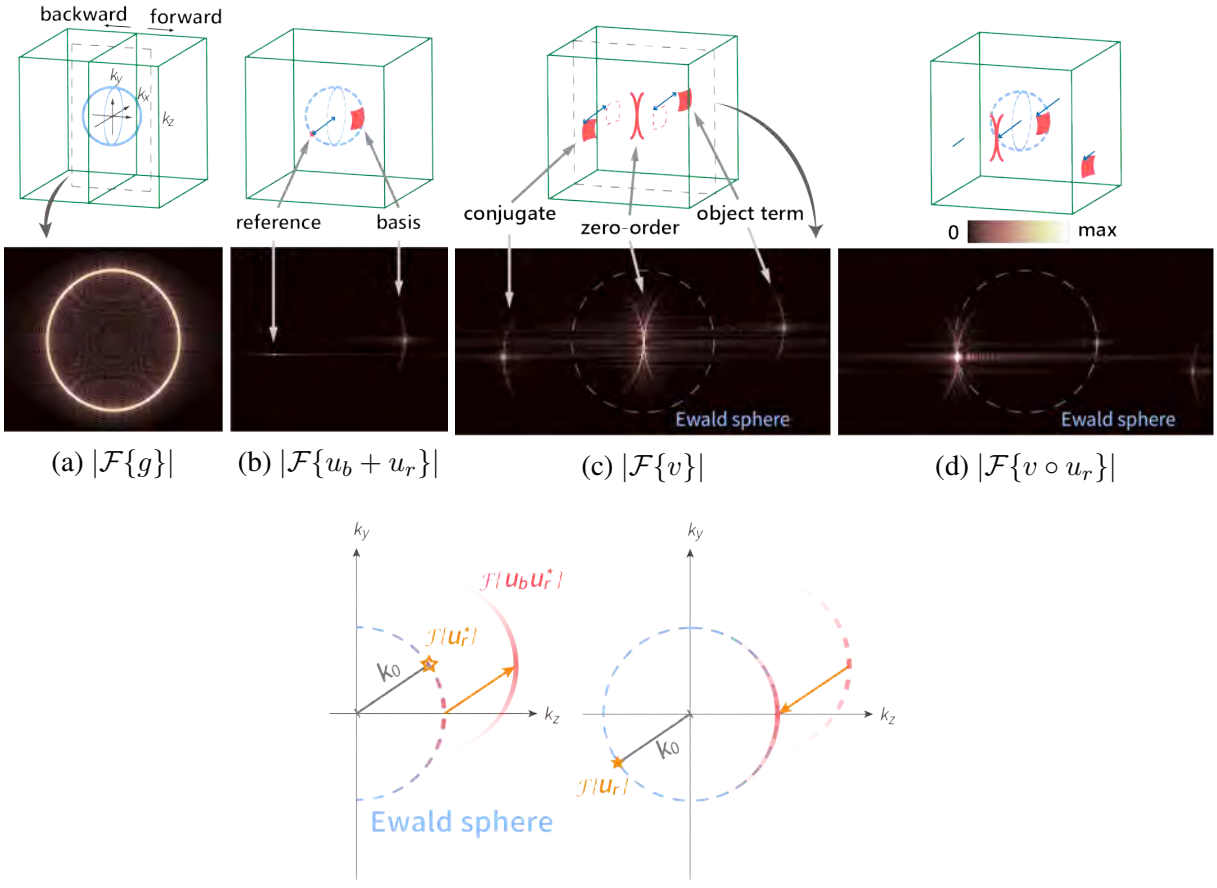
where $v(\mathbf{x})$ is the scattering potential and is defined as:

$$v(\mathbf{x}) = \frac{k_0^2}{4\pi}[(\frac{n(\mathbf{x})}{n_0})^2 - 1]. \tag{3.3}$$

Under the Born approximation, the scattered field $u(\mathbf{x})$ is approximated as a superposition of the incident wave $u_{\text{in}}(\mathbf{x})$ and a scattered component $u_s(\mathbf{x})$, *i.e.*, $u(\mathbf{x}) = u_{\text{in}}(\mathbf{x}) + u_s(\mathbf{x})$. The scattered component $u_s(\mathbf{x})$ can be modeled as a 3D convolution with the Green's function:

$$u_s(\mathbf{x}) = \mathcal{E}(v(\mathbf{x}),\ u_{\text{in}}(\mathbf{x})) = \mathcal{F}^{-1}\{\mathcal{F}[u_{\text{in}}(\mathbf{x}) \circ v(\mathbf{x})] \circ \mathcal{F}[g(\mathbf{x})]\}, \tag{3.4}$$

where $g(\mathbf{x}) = \frac{\exp(ik_0\mathbf{x})}{|\mathbf{x}|}$ is the Green function, and $\mathcal{F}$ represents 3D Fourier transform. Next, we illustrate the recording and playback process in the Fourier space, as shown in Fig. 3.5.



(a) $|\mathcal{F}\{g\}|$ (b) $|\mathcal{F}\{u_b + u_r\}|$ (c) $|\mathcal{F}\{v\}|$ (d) $|\mathcal{F}\{v \circ u_r\}|$

(e) Single recording (f) Single playback

Figure 3.5. Fourier analysis for volume diffraction holography. (a-d) show the amplitude of the Fourier transforms of (a) the Green's function, (b) the interference field $(u_b + u_r)$ in reflection hologram geometry, (c) the scattering potential v, (d) the playback field (before applying the Green's function). Bottom of (a-d) show amplitude slices in corresponding volumes.

Green function. In Fourier space, the Green function has the form of $\mathfrak{g}(\mathbf{k}) = \frac{1}{|\mathbf{k}|^2 - k_0^2}$. As the spatial frequency approaches the wave vector k_0, the value of the Green function approaches infinity. Since the 3D convolution is to equivalently multiply the incident spectrum by the Green spectra, it serves as a sampling function near k_0, *i.e.*, $|\mathbf{k} - k_0| < \epsilon$. In literature, the Green spectra is referred as the Ewald sphere, as shown in Fig. 3.5a. The forward and backward half-space represents the propagation/sampling in forward and backward directions.

Recording phase. The recording phase involves counter-propagating two waves inside the VH. We use $u_b^{(j)}(\mathbf{x})$ and $u_r^{(j)}(\mathbf{x})$ to represent the 3D distribution of the basis wave $u_b^{(j)}(x, y)$ and reference wave $u_r^{(j)}(\mathbf{x})$, by using the 3D convolution in Eq. 3.4. Since we have constrained the reference wave as a ramp phase, it only consists of one wave vector $\mathbf{k}_r^{(j)}$. In Fourier space, this signal is a point on the backward Ewald hemisphere, while the basis signal is located on the forward hemisphere. The spectra of the interference pattern is shown in Fig. 3.5b. After the j-th exposure, the refractive index variation of the VH is proportional to the intensity of the interference pattern:

$$\Delta n(\mathbf{x}) \propto |u_b(\mathbf{x}) + u_r(\mathbf{x})|^2 = \underbrace{u_b^2(\mathbf{x}) + u_r^2(\mathbf{x})}_{\text{non-interferring terms}} + u_b(\mathbf{x}) \circ u_r^*(\mathbf{x}) + u_b^*(\mathbf{x}) \circ u_r(\mathbf{x}), \tag{3.5}$$

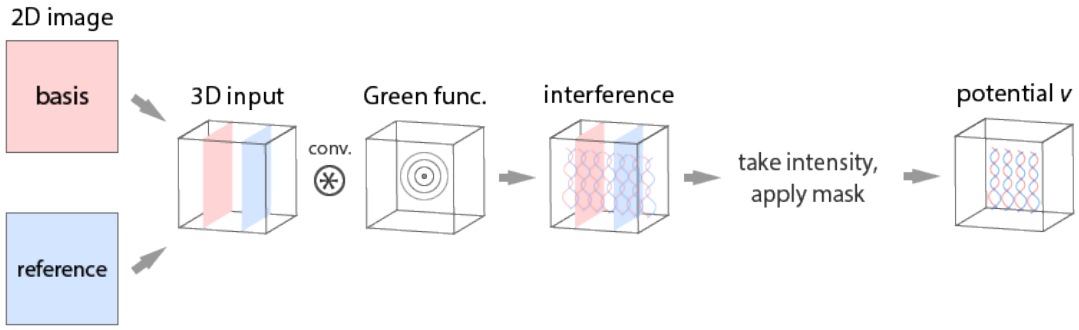
where index j is omitted. u_b^* and u_r^* are the complex conjugates of u_b and u_r. In Fourier space, the Hadamard multiplication becomes convolution. Since the reference wave and its conjugate resemble two delta signals. The convolutions operate as spatial shifts. The basis spectra and its conjugate have been shifted away from the Ewald sphere, with shift magnitudes of k_0. The

scattering potential v preserves this behavior as taking the square of Δn is performing pixel-wise scaling. The Fourier spectra of v can be seen in Fig. 3.5c. The non-interferring terms are centered around the zero frequency.

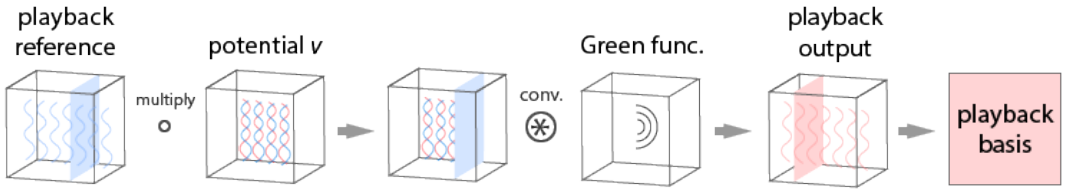
Playback phase. To playback a recorded basis wave, the same reference wave is used with the same illumination geometry. The field $\mathcal{F}\{u_r \circ v\}$ before applying the Green function in Eq. 3.4 is shown in Fig. 3.5d. The reference wave shifts the potential spectra, resulting in the term $\mathcal{F}\{u_b \circ u_r^*\}$ shifted back to the Ewald sphere. After applying the Green function, all other signals have been suppressed while the basis signal $\mathcal{F}\{u_b\}$ has been amplified. We refer to the playback 3D light field in the VH as $u_{\text{b, p}}(\mathbf{x})$, and the playback basis image located at the basis location as $u_{\text{b, p}}(x, y)$.

Numerical implementation. The simulation pipeline is illustrated in Fig. 3.6. In the recording step, the basis image and the reference image are placed in different depth layers in an empty 3D volume. The distance between the two images is the hologram thickness. The 3D volume is convolved with the Green's function. This operation is performed by taking 3D FFT of both volumes, pixel-wise multiplying the spectra, and then taking the inverse 3D FFT. The output is regarded as the interference pattern. To construct the scattering potential, we need to take the intensity of the interference and follow Eq. 3.5. Note that there is a step for applying a mask function to constrain the light interference between the hologram region (between the basis and reference image). This masking operation can be performed either in the spatial domain as shown in Fig. 3.6a, or in the FFT domain. When in the FFT domain, the basis 3D cube and the reference 3D cube are separately convolved with the forward Green's function and the backward Green's function, constraining the propagation direction, *i.e.*, counter-propagating.

To construct a forward Green's function, we apply a zero-masking operation on the spectra of the Green's function, which is the Ewald sphere as shown in Fig. 3.5a. The backward Green's function should mask out the forward hemi-space in the Fourier space. The playback phase starts with the same playback reference cube as used in the recording phase. The playback reference is pixel-wise multiplied with the potential v from the recording phase. The product is then convolved with the *forward* Green's function so as to obtain the reflection wave. Then, the same depth layer as the recording basis depth is used as the playback output image.



(a) Recording of a basis image.

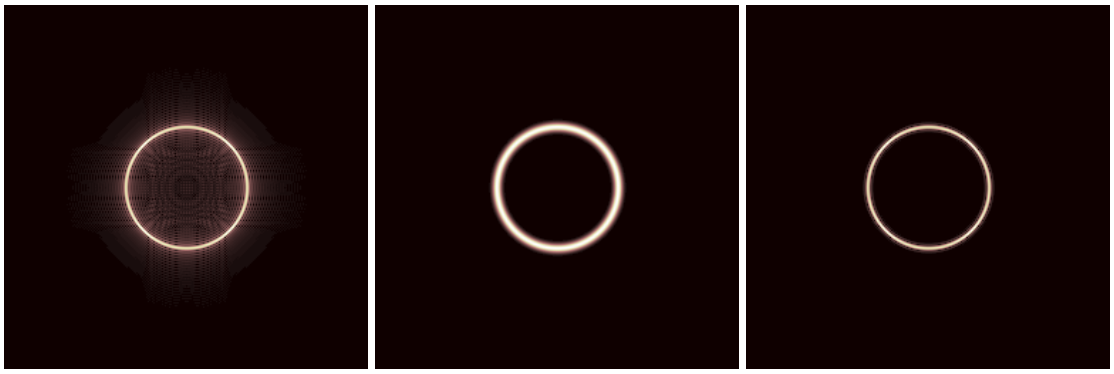


(b) Playback of the recorded basis image.

Figure 3.6. Hologram simulator algorithm pipeline.

There are singularities in the Green's function, both in the spatial ($g(x) = (ik_0x)|x|$) and the frequency ($\mathfrak{g}(\mathbf{k}) = \frac{1}{|\mathbf{k}|^2 - k_0^2}$) domain. In our simulation, we first construct the Green's function in the spatial domain and then take its FFT to simulate the Ewald sphere. To avoid dividing by zero, we sample the coordinates with even sample numbers in symmetry. The spectrum of the Green's function is shown in Fig. 3.7a. The spectrum shows aliasing artifacts due to the

undersampling of the Green's function (around x=0). We therefore apply a Gaussian mask to suppress the signal away from the wave vector, shown in Fig. 3.7. The Ewald sphere after masking is shown in Fig. 3.7c.



(a) Ewald sphere *w/o* mask. (b) Gaussian mask. (c) Masked Ewald sphere.

Figure 3.7. Suppressing the signal amplitude away from $k = k_0$ in the frequency domain for the Green's function by applying a Gaussian mask.

We implemented a normalization step for the Green's function in the frequency domain. Each k_z layer is pixel-wise normalized with the complex sum values along the k_z direction. This can be viewed in Fig. 3.8b (before normalization) and 3.8b (after normalization). Note that in this simulation the Ewald sphere looks elipsoidical rather spherical. This is because we make the unit of k_z different than k_x and k_y axes, such that the Ewald sphere covers most of the input spectrum. The right hand side shows k_x-k_y plot of the sphere. The 3D convolution results using the Green's function *w/o* and *w/* k_z normalization are shown in Fig. 3.8c and Fig. 3.8d. As can be seen in the middle colume, most of the signal of the input image has been preserved by using the normalized Green's function.

3.4. Physics-in-the-loop basis learning

In this section, we detail experiments for physics-in-the-loop basis learning. The learning archiceture is shown in Fig. 3.4. We convert the hologram simulator to a differentiable model. During each training forward pass, the initial bases are first fed into the hologram simulator to simulate the recording and playback phases. The outputs are the same number of bases. The output "playback" bases are then multiplied with the coefficients to construct the target image. For the backpropagation step, the loss is first propagated to update the "playback" bases, and then backward through the hologram simulator to update the initial bases. By doing this, the initial bases are updated in a way that incorporates the properties of the hologram simulator.

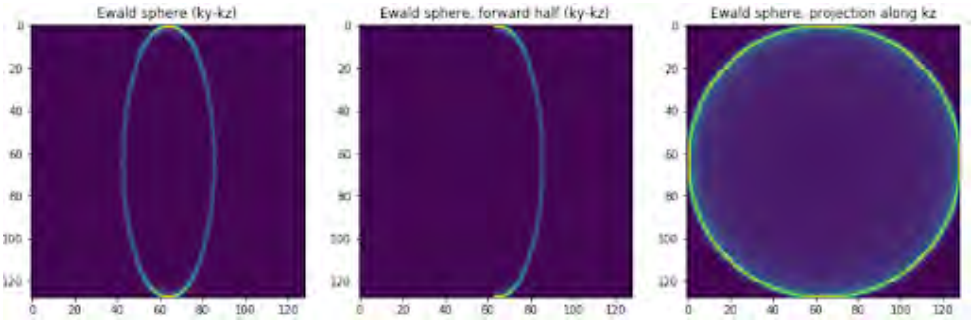
As an intermediate step, we propose GreenNet. GreenNet refers to the architecture that the initial bases are only convolved with the 3D Green's function, but not through the entire recording-playback process. This process is a portion of the full simulator and is computationally efficient. Additionally, HoloNet can initialize its bases either randomly, or from the results of GreenNet. We refer to the training strategy that HoloNet initialized by the pre-trained GreenNet bases as GreenHoloNet. The results are shown in Fig. 3.9. As can be seen, the BaseNet plateaus around 26.3dB near 300 epochs. At the same number of epochs, the GreenNet achieves better results at 26.5dB. HoloNet trained from scratch (gray curve) does not show enough progress over certain epochs. However, HoloNet trained from GreenNet bases at first drops performance, but is able to quickly improve and keeps the growing trend of the GreenNet, plateaued at 27.9dB.

The results comparing the BaseNet and the GreenHoloNet are shown in Fig. 3.10. We can observe sharper regions in the GreenHoloNet results compared to both the BaseNet and GreenNet.

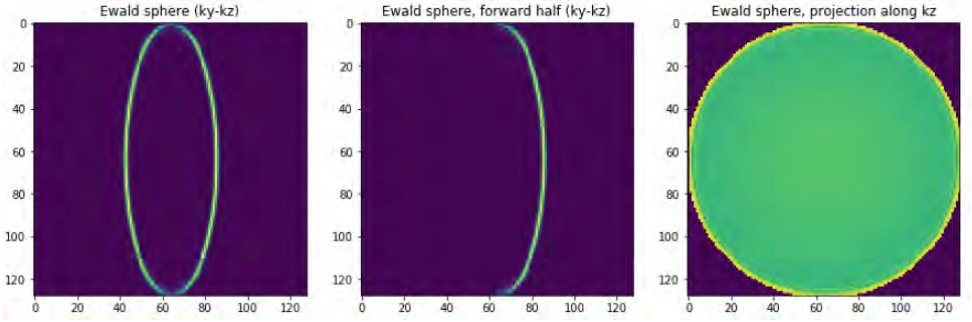
The visualization of the learned bases is shown in Fig. 3.11. Compared to BaseNet, the GreenHoloNet bases are less noisy both in the real and imaginary components.

3.5. Conclusion

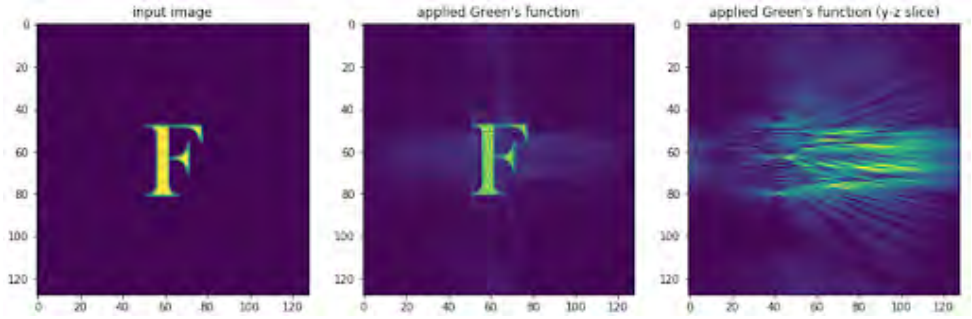
In this chapter, we have demonstrated a physics-in-the-loop deep learning architecture for optimizing the image decomposition. The physics part incorporates a volume hologram simulation process that serves as a predictor to "record a hologram for each basis and playback". By the physical simulation, the learned bases achieve better reconstruction performance than the basic architecture (BaseNet). However, this performance improvement is not achieved by directly training HoloNet. In fact, we have shown that directly learning basis using HoloNet is unable to achieve basis learning. Therefore, we use a part of the HoloNet, *i.e.*, the GreenNet, to first optimize the bases. The learned bases are then used as initialization for the HoloNet, referred as GreenHoloNet. In this time, the GreenHoloNet is able to fine-tune the bases and steadily improve the performance. The learned bases also have potential to be finally printed by hardware.



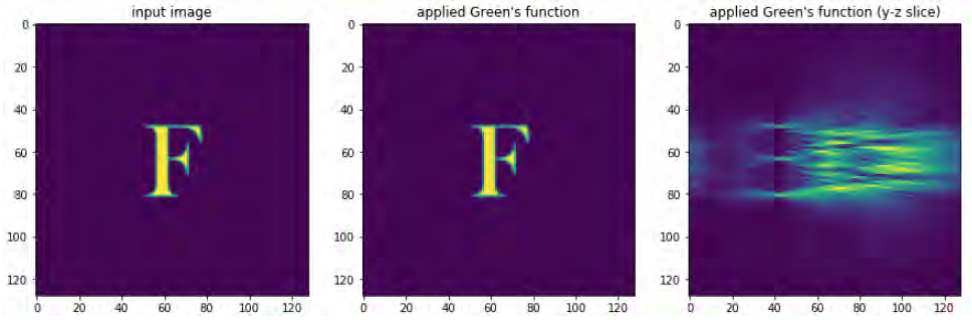
(a) Ewald sphere before k_z normalization.



(b) Ewald sphere after k_z normalization.



(c) 3D convolution using Ewald sphere *w/o* k_z normalization.



(d) 3D convolution using Ewald sphere *w/* k_z normalization.

Figure 3.8. k_z normalization.

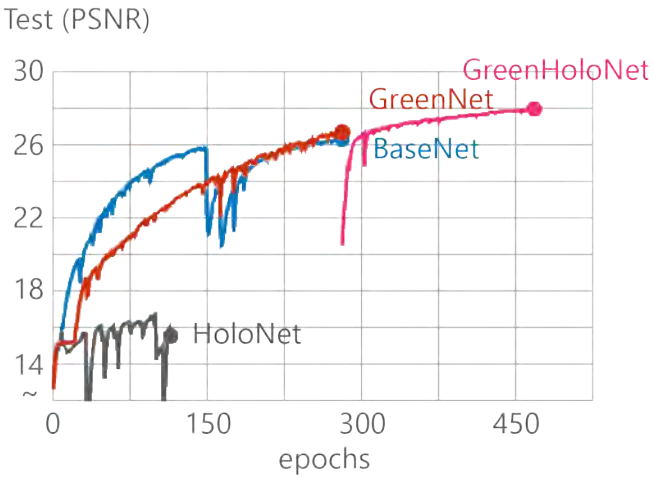


Figure 3.9. Progressive testing results for BaseNet, HoloNet, GreenNet and GreenHoloNet.

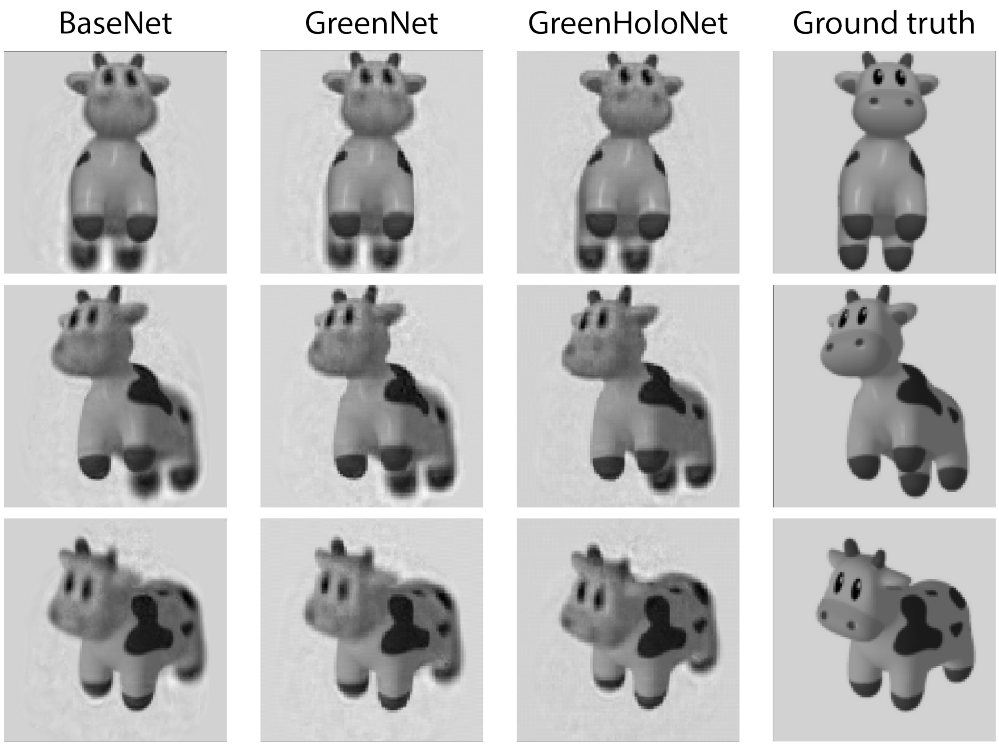


Figure 3.10. Results comparing BaseNet and GreenHoloNet.

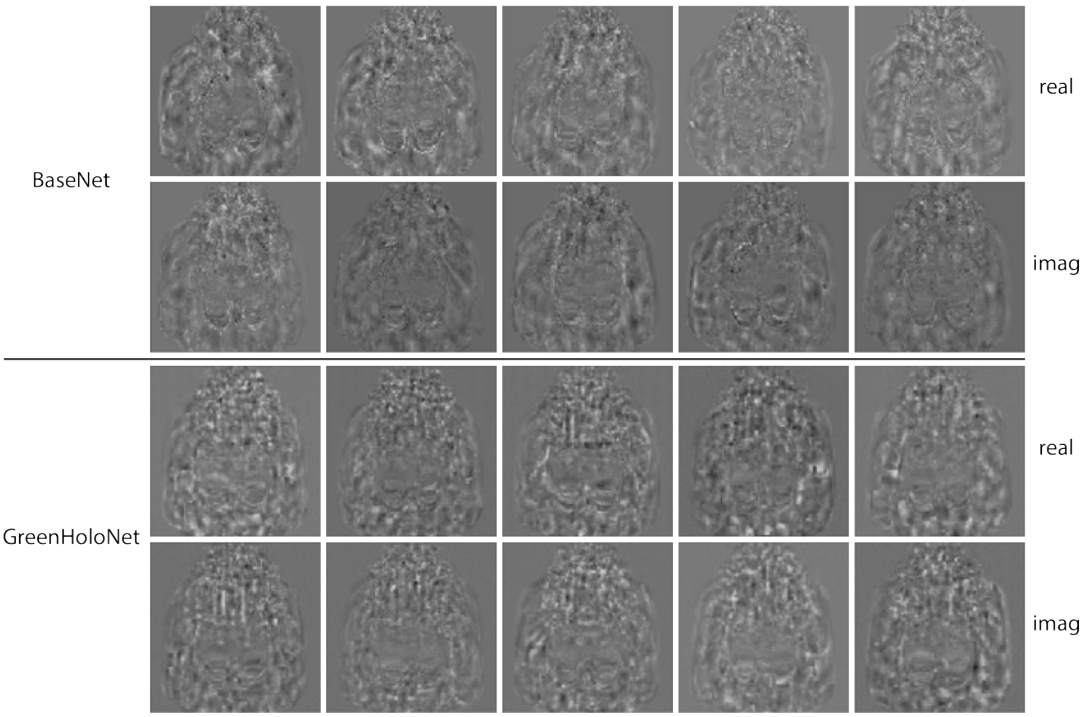


Figure 3.11. Visualization of example bases learned from BaseNet and GreenHoloNet.

CHAPTER 4

Dictionary learning-based space-time super resolution for on-chip holographic imaging

This chapter demonstrates a synergistic strategy without the use of differentiable imaging models. On-chip in-line holography (OIH) is a lens free imaging method that offers high resolution over a wide field-of-view. However, high spatial resolution sensors usually yield low temporal resolution. Compressive sampling schemes can be used for improving temporal resolution, leaving spatial resolution recovery a challenging task. In this work, we consider the problem of localizing objects in 3D at both high spatial and temporal resolutions. We present a dictionary-based phase retrieval approach for space-time super resolution on OIH videos. Here, the phase retrieval algorithm can be viewed as a physics-based solution while the dictionary can be regarded as learning-based model. By leveraging single frame super resolution schemes, we recover high resolution holograms from sub-sampled holograms. We show that this approach performs better than conventional phase retrieval methods applied directly to sub-sampled holograms. Through pixel sub-sampling, we experimentally achieve a factor of 9 increase in sensor frame rate while monitoring the in vivo movement of Euglena microorganisms.

4.1. Introduction

Lensless on-chip imaging offers the ability to simultaneously obtain a high resolution image over a wide field-of-view (FOV) in a simple optical setup. In a lensless on-chip imaging experiment, one typically places a sample within several millimeters of a digital sensor. By

illuminating the sample with a spatially coherent light source, a diffraction pattern is formed at the nearby sensor, which may be captured as a digital in-line hologram. A phase retrieval algorithm typically recovers the sample's amplitude and phase, at high fidelity, from the recorded hologram intensities [**65**].

Lensless on-chip holographic imaging has been widely used to investigate biological and chemical phenomena at the micro and/or nano scale. Recent examples include high resolution and wide field-of-view imaging of malaria-infected cells [**66**], dense pathology slides [**67**], and nanometer-scale viruses [**68**]. The issue of spatial resolution improvement was addressed and explored based on interpolation [**69, 70**]. Recent development also focuses on sparsity-based super resolution [**71**]. While these samples are primarily stationary over time, it is also possible to monitor in-vivo dynamic phenomena using lensless holographic video. On-chip examples of monitoring biophysical processes include discovering the spiral trajectories of sperm [**72**], the formation of endothelial cells into microvessels [**73**], and analyzing single-cell motility [**74**].

The total pixel count of the reconstructed images from these setups (i.e., the system space-bandwidth product) is simply set by the effective pixel count of the detector array. As detector array sizes grow into the regime of hundreds of megapixels, a limited detector array readout rate will eventually limit the rate of high-speed lensless image acquisition. A tradeoff space thus emerges between the spatial and temporal resolution of a lensless on-chip imaging experiment: either images can be acquired at either high resolution, or at high frame rates, but currently not both.

The same tradeoff space also currently impacts video capture in conventional cameras. The limited speed of sensor hardware for pixel readout, analog-to-digital conversion, and a constrained on-board memory together form a data bottleneck. To overcome this limitation, many

high speed camera sensors now offer a multitude of video frame rates at different image resolutions. A typical example is the recent Casio EX-F1 camera, which trades off image resolution and frame rate in an inversely proportional manner, offering 2.07 megapixels (MP) at 30 frames per second (fps), 0.20 MP at 300 fps, 0.08 MP at 600 fps, and 0.03 MP at 1200 fps[1]. Here, the sensor data rate faces an approximate upper bound of 65 MP per second.

A number of different coding strategies were recently proposed to overcome this data readout limit in standard video. For example, offsetting the exposure time of interleaved pixels may simultaneously provide high-speed video and high-resolution imaging [**75**]. A similar strategy may be applied to the interleaved frames from a camera array [**76**]. Alternatively, the incident light may be coded into a spatio-temporal pattern, either using a spatial light modulator [**37, 77**], global shutter [**78**] or translating mask [**36**]. Subsequently, an inversion algorithm, typically operating within a compressive sensing framework that assumes scene sparsity, can recover a high-resolution and high-speed video[**39**]. This strategy was most recently applied with a streak camera to create videos of light propagation resolved down to picosecond time scales [**79**].

Similar coding strategies may also help overcome the space-time resolution tradeoff in lensless holographic imaging. Unlike traditional video, however, the operation of a lensless holographic setup is fundamentally connected to its phase-retrieval algorithm. An ideal strategy to improve lensless image readout rates would operate in tandem with phase retrieval [**80**]. As with the compressive video recovery schemes above, phase retrieval must also assume some prior knowledge about the imaged sample to ensure accurate algorithm convergence. Examples include a known finite sample support [**81, 82**], sparsity[**71**], non-negativity or an intensity

[1]Casio cameras, `http://www.casio.com/products/archive/Digital_Cameras/High-Speed/EX-F1/`

histogram. Several recent works examine how sample sparsity permits accurate sample reconstruction from a limited number of holographic measurements [**83, 84, 39, 85, 86, 87**]. To the best of our knowledge, no work has yet examined whether prior knowledge of sample support alone may also relax required in-line holographic image readout rates, nor has demonstrated that such a modified phase retrieval process can improve the frame rate of on-chip holographic video.

In this work, we aim at overcoming the space-time resolution tradeoff with a dictionary-based image super resolution approach. Our result demonstrates a suitable performance for space-time reconstruction with a temporal increase of $9\times$ with 4×4 sub-sampling. The rest of this article is organized as follows: Section 4.2 reviews related works. In Section 3, we first introduce the principle of OIH, followed by an examination of its auto-refocusing capability. Section 3.2 performs 3D tracking of biological sample (Euglena) and addresses the problem of space-time resolution tradeoff. Section 3.3 proposes our dictionary-based super resolution scheme. Section 4 shows our simulation and reconstruction results for two dictionary-based methods. Section 5 concludes the paper.

4.2. Related works

4.2.1. Digital holography and coherent diffraction imaging

Numerical reconstructions for digital holography (DH) usually branch into tomographic and phase retrieval directions. Tomographic reconstruction enables recovering 3D information from 2D images, such as refocusing [**88**] and sectioning [**89**]. This property comes from the depth-dependent nature of the 2D coherent spread function originating from coherent light propagation. This is analogous to incoherent point spread function (PSF). Recent advancements in

DH leveraged compressed sensing (CS) theory, a mathematical framework that allows robust signal reconstruction from substantially fewer measurements [**90, 91, 92**]. The 3D reconstruction capability of DH implies a proper CS mechanism [**83, 93**]. Besides the 3D reconstruction capability, computational methods are also devised to retrieve phase information from holograms. The phase retrieval task is commonly addressed in the literature of coherent diffraction imaging (CDI), where a diffraction pattern of the object signal is sampled instead of an interference pattern with a reference beam. However, the simplicity of the in-line holographic setup actually aligns with CDI as it is implemented by enforcing the illumination arm to serve both as the reference and object beam. This is also based on the assumption that the objects do not occlude the illumination much so that the non-occluded part of the illumination can be regarded as the reference beam. Several algorithmic schemes have been proposed for solving phase retrieval problem. Among them, popular ones are alternating projection algorithms [**94, 81**]. Pioneered by the work of Gerchberg and Saxton, alternating projection methods aim to recover the complex field by iteratively imposing real-plane and Fourier-plane constraints, such as non-negativity and/or object boundary/support in real-plane and Fourier-plane magnitude, respectively. Follow-ups [**95, 82**] have been proposed to overcome the limitations such as high-noise vulnerability and pre-defined support. Another approach [**96, 97**] formulates the phase retrieval problem as a semi-definite programming (SDP) problem and solves it by matrix lifting, i.e., replacing the sought vector with a higher dimensional matrix. Recent advances in solving phase retrieval problems incorporated sparsity constraints in some known representation and have been incorporated into the previous methods [**98, 99, 100**].

4.2.2. Space-time resolution tradeoff

Research scientists have proposed many heuristic methods for tackling the space-time resolution tradeoff of modern sensors. Hybrid camera systems with cameras, featuring high spatial resolution and high temporal resolution separately, have been proposed for synthesizing high-resolution videos and motion-debluring [**101, 30, 102**]. Another part of literature in computational photography focuses on the encoding of multiplex information in the temporal domain, i.e., coded exposure. Raskar et al. [**103**] pioneered the concept of coded exposure by using a coded global shutter. Gu et al. [**104**] proposed coded rolling shutter for CMOS image sensors. Gupta et al. [**30**] proposed a similar system with motion-aware photography. Instead of trading off space-time resolution on the whole scene [**75**], they did so only at the moving regions of the scene. Reddy et al. [**37**] proposed a programmable pixel-wise compressive sensing camera based on LCoS. Liu et al.[**77**] proposed a dictionary learning approach for recovering a multi-frame video from a single coded image. The advantage of using a dictionary-based approach is that it does not require analytical motion models. Wang et al. [**39**] exploited spatial-temporal redundancy and performed 4D reconstructions (3D position with time) from a single image. In their implementation, in-line holography served as a spatial encoder and pixel-wise coded exposure was used as temporal encoder.

4.3. Phase retrieval algorithm for on-chip holographic imaging

The phase retrieval task is commonly addressed in the literature of coherent diffraction imaging (CDI), where a diffraction pattern of the object signal is sampled instead of an interference pattern with a reference beam. In view of this, the LOIH setup aligns with CDI by

enforcing the illumination arm to serve both as the reference and object beam. Several algorithmic schemes [**100, 105**] have been proposed for solving phase retrieval problem based on CDI/LOIH setup. This type of algorithms can be categorized as alternating projection methods, originally proposed by [**106, 81**], which aim to recover the complex field by iteratively imposing real-plane and Fourier-plane constraints, such as non-negativity and/or object boundary/support in real-plane and Fourier-plane magnitude, respectively.

A simple on-chip imaging setup is shown in Fig. 4.1(a). A small light source (e.g., light-emitting diode (LED)) positioned at a relatively large distance from the detector (sensor) illuminates the sample with quasi-monochromatic light. The spatial coherence of the source, defined by its distance from the sensor and the width of its active area should be sufficient to produce high-contrast diffraction fringes at the detector plane. The samples for imaging are prepared on a transparent glass. The glass is closely placed and adjusted parallel to the sensor plane. By modeling the incident light reaching the sample plane as a coherent plane wave, the modulated field right after the sample plane is a direct indication of the optical properties of the sample, i.e., the object signal $f(x, y)$ can be expressed as $f(x, y) = A(x, y) \exp^{i\phi(x,y)}$. The signal is usually assumed to be complex with amplitude A so that its phase information ϕ reflects the optical thickness of the object. The observed signal is the field on the sensor which is propagated a distance d and only the intensity information is received, i.e., $I(x, y) = |h(x, y; d)|^2$. The propagation process is well described by the Fresnel-Kirchoff diffraction theory and can be equivalently modeled as a convolution process, that is,

$$h(x, y; d) = Q_d * f, \tag{4.1}$$

with kernel

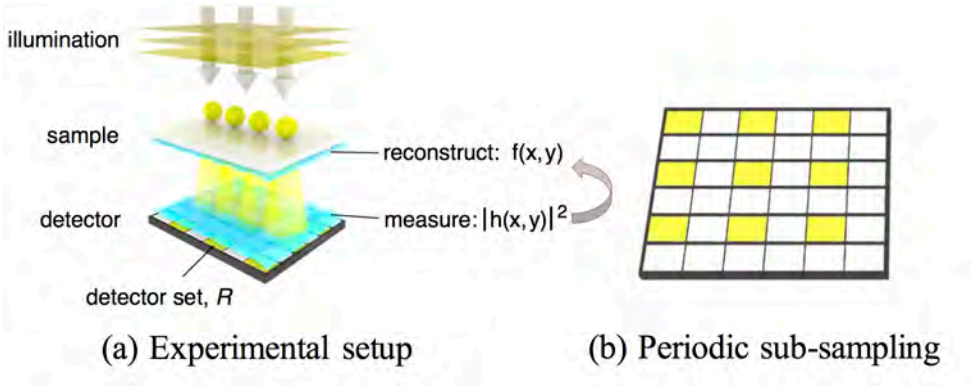


Figure 4.1. (a) Experimental setup for on-chip in-line holography. (b) Periodic sub-sampling. The pattern is static over time.

$$Q_d = -\frac{ik}{2\pi d}\exp\left[ik\sqrt{x^2+y^2+d^2}\right] \tag{4.2}$$

where $k = \frac{2\pi}{\lambda}$ and λ is the wavelength. This convolution process can be efficiently implemented in the Fourier domain, with a corresponding multiplicative kernel to be $Q_d^F = \exp\left[id\sqrt{k^2-k_x^2-k_y^2}\right]$ and a frequency-cut-off aperture $C = circ\left(\frac{\sqrt{k_x^2+k_y^2}}{k}\right)$. This method is also referred to as angular spectrum method (ASM) [**107**].

Phase retrieval algorithms usually leverage this diffraction process and seek for the recovery of the original signal through an iterative process. We start by adopting the error reduction algorithm [**82**], which iteratively projects an initial estimation of f onto two constraints in two different domains. On the sample plane, it enforces the constraint of the object's support, i.e., location and shape etc., while on the sensor plane, it replaces the intensity with the measured values. However, the exact support of the object cannot be easily obtained. A "shrink-wrap" method [**?**] method has been proposed for iteratively updating the object support by blurring it and thresholding it. The convergence of this type of alternating projection method has been examined in [**82**].

4.3.1. 3D localization from OIH

The depth-dependent nature of the propagation kernel (Eq. 4.2) brings up the digital refocusing property through a convenient back-propagation procedure [**107**]. In combination with the convenient setup of OIH, the 3D information can be easily extracted from a 2D hologram [**88**]. Note that the useful depth information of the object is also a pre-requisite for the phase retrieval (PR) algorithms based on the OIH setup. However, conventional PR algorithms usually take a visually-determined depth. We intend to design a scheme that estimates the depth automatically and integrate this into the PR algorithm, which saves the effort of visual justification. In order to achieve this goal, we examine the refocusing capability in this subsection and design an auto-refocus method for automatic determination of the object depth. We start by a simulation example of an extended object, as shown in Fig. 4.2. An extended object, with image size 256×256 pixels, is constructed within a depth range of 1.3 mm to 2.7 mm away from the imaging plane. The simulated hologram is shown in (a). A depth volume ($U \in \mathbb{R}^{256\times256\times100}$, absolute value from back propagation) is reconstructed with a refocus range from 1.0 mm to 3.0 mm. By scanning a square window of size $(2k+1) \times (2k+1)$ (e.g., $k = 2$ pixels) in the image at depth z and repeating this scanning procedure across different depths, a variance cube (V) can be computed as,

$$V[x, y, z] = \sum_{i=x-k}^{x+k} \sum_{j=y-k}^{y+k} |U[i, j, z] - \bar{U}[x, y, z]|^2, \tag{4.3}$$

where $\bar{U}$ is the average value

$$\bar{U}[x, y, z] = \frac{1}{(2k+1)^2} \sum_{i=x-k}^{x+k} \sum_{j=y-k}^{y+k} U[i, j, z] \tag{4.4}$$

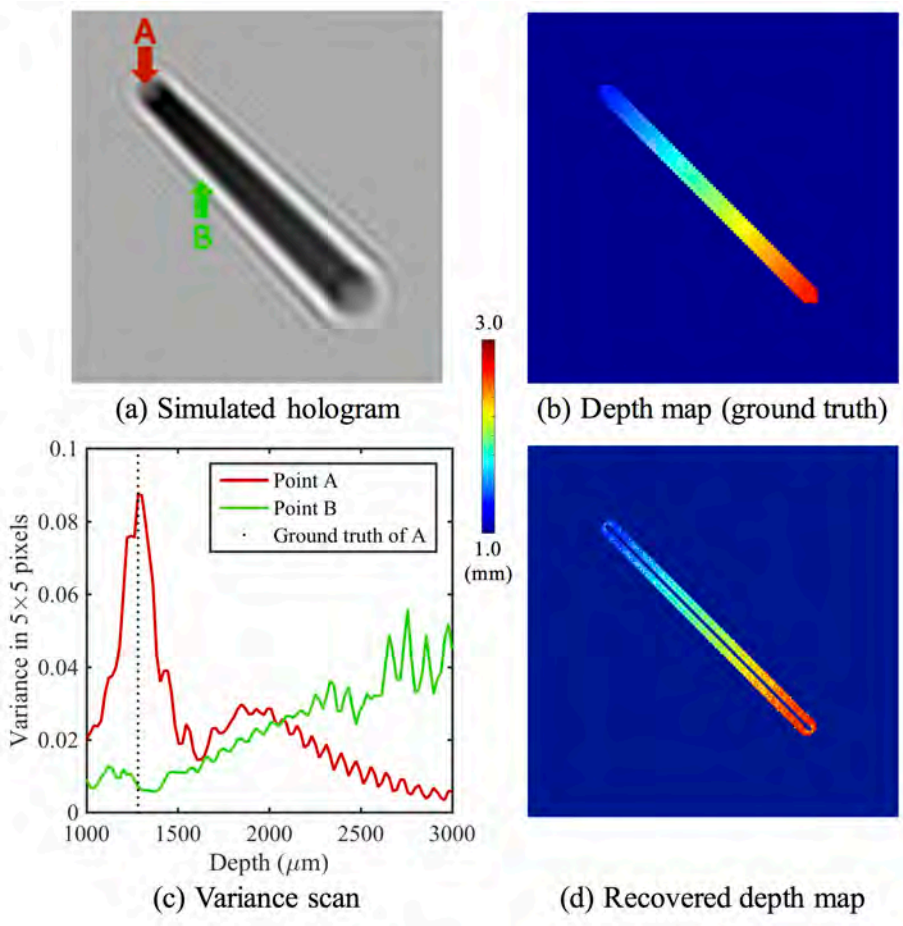


Figure 4.2. Simulation of depth recovery. (a) Simulated hologram. (b) Constructed depth map as ground truth. (c) Scan of variance on two example points pointed out in (a). (d) Recovered depth map.

Fig. 4.2 (c) plots the variance vs. depth for the two example points shown in (a). It can be seen that the maximum value of the variance form a good estimate of the depth (Point A), i.e.,

$$d_e[x, y] = \arg\max_z V[x, y, z] \tag{4.5}$$

However, points that are not of great interest, e.g., Point B, usually have spurious variance peak values and will be assigned to a wrong depth. Thus, a threshold is set to filter out low

variance peak values so that only object pixels are used for depth estimation. Fig. 4.2 (d) shows the recovered depth map from the hologram. Note that the depth at the edges of the object has been successfully recovered while the depth estimates in the inner-area are removed by thresholding because of their low peak values the variance. This effect provides an insight into the depth estimation of biological samples. Reconstructions for biological samples require a rough estimation of location, which is usually adjusted visually. We use this depth estimation technique to automatically refocus and locate the position of the object and integrate this technique into a phase retrieval algorithm. Figure 4.3 shows our reconstruction result of Blepharisma, in which (b) shows a preliminary depth estimation of the object. Here, we insert another filtering step using k-means clustering, which seeks an optimal partition of the depth values into k sets, $\mathbf{S} = \{S_1, S_2, ...S_k\}$,

$$\underset{\mathbf{S}}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{d_e \in S_i} |d_e - \precneqq_i|^2, \tag{4.6}$$

where $\precneqq_i$ is the mean values of each partition. In this case, we set $k = 3$ because the background pixels are assigned as the frond end of the depth range and the error pixels are mostly assigned near the back end of the depth range (also see Fig. 4.2 (c)). The boundaries of the object are the principle points for estimating the depth. Fig. 4.3 (c) shows the filtered depth. This leaves out the pixels that are located roughly on the object. The mean value of these effective pixels is used as the depth estimation for phase retrieval. Fig. 4.3 (d-f) present the phase retrieval result. We then use the recovered support to filter the background pixels, as shown in Fig. 4.3 (g). We focus our depth estimation only on boundary pixels, the filtered boundary area by k-means clustering is shown in Fig. 4.3 (h).

Experimentally validating the depth estimation results is difficult because of the difficulty in establishing the ground truth. However, it is still possible to examine the relative depth range.

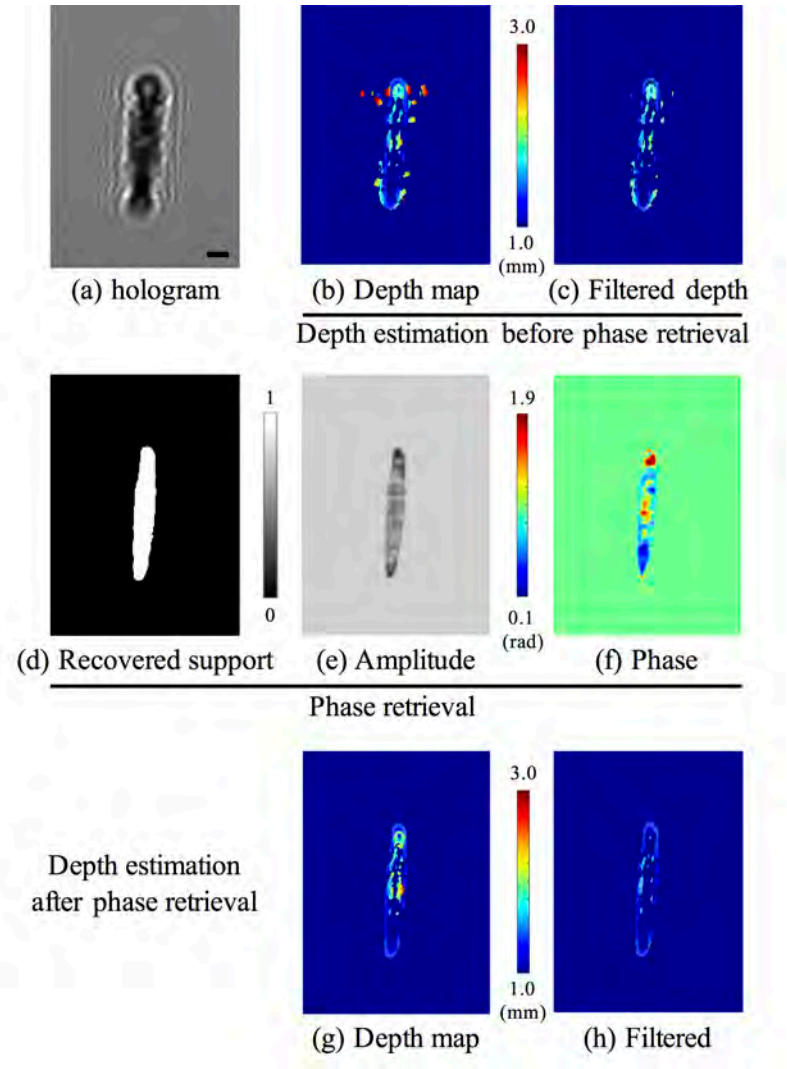


Figure 4.3. Phase retrieval improves depths recovery. (a) Hologram of blepharisma. Scale bar: $40\mu m$ (b) Depth map. (c) Filtered depth map after k-means clustering (k = 3). (d)-(f) Phase retrieval using estimated depth from (c). (d) Recovered support. (e) Recovered amplitude. (f) Recovered phase. (g) Depth map after PR. (h) Filtered depth around boundary.

We show a validation example by observing two holograms of the same Blepharisma at different time frame, as shown in Fig. 4.4 (a). The free-swimming rotation of the Blepharisma results in depth differences for the boundary, as shown in Fig. 4.4 (b). Based on this, we collect the

near-boundary pixels and fit a linear regression model for these location points. The range of the projected values is regarded as a length estimation of the Blepharisma. As is also pointed out in Fig. 4.4 (c), the two estimates are $262.4\mu m$ and $264.7\mu m$. The normal size of Blepharisma is between 75 and 300 μm. The samples are cultured for a week after receiving them. The error between the two length estimates originates from the subtle shape change of the Blepharisma itself and also the defocusing effect resulting from the phase retrieval process. It is evident that the Blepharisma on the right panel has an extended depth range while phase retrieval only performs alternating projection on two planes with fixed depths. The defocus effect influence the accuracy of the support recovery and will yield less accurate estimation for the boundary.

4.3.2. 3D tracking over time

By applying the auto-refocusing technique as described above, one is able of achieving quasi-pixel-wise depth information based on the OIH setup. This is especially useful for localizing *in vivo* biological samples and is also beneficial for developing tracking schemes. However, tracking moving objects requires high-enough sensor frame rate. Unfortunately, high spatial resolution and high temporal resolution are usually not satisfied simultaneously. For example, in Fig. 4.5 left-most, we present a tracking result based on high (full) resolution imaging. In this case, a larger size of variance-scanning window (e.g., 20×20 pixels, empirically determined for Euglena) is used to estimate the overall depth of each Euglena. However, the temporal performance is poor, only yielding 4.4 FPS. In order to achieve the goal of tracking objects with visually indistinguishable frame rate, it is non-trivial to leverage a compressive sampling technique, such as a periodic sub-sampling scheme (static over time) in Fig. 5.6(b), to increase the temporal resolution. Figure 4.5 shows an improved performance of tracking Euglena based on

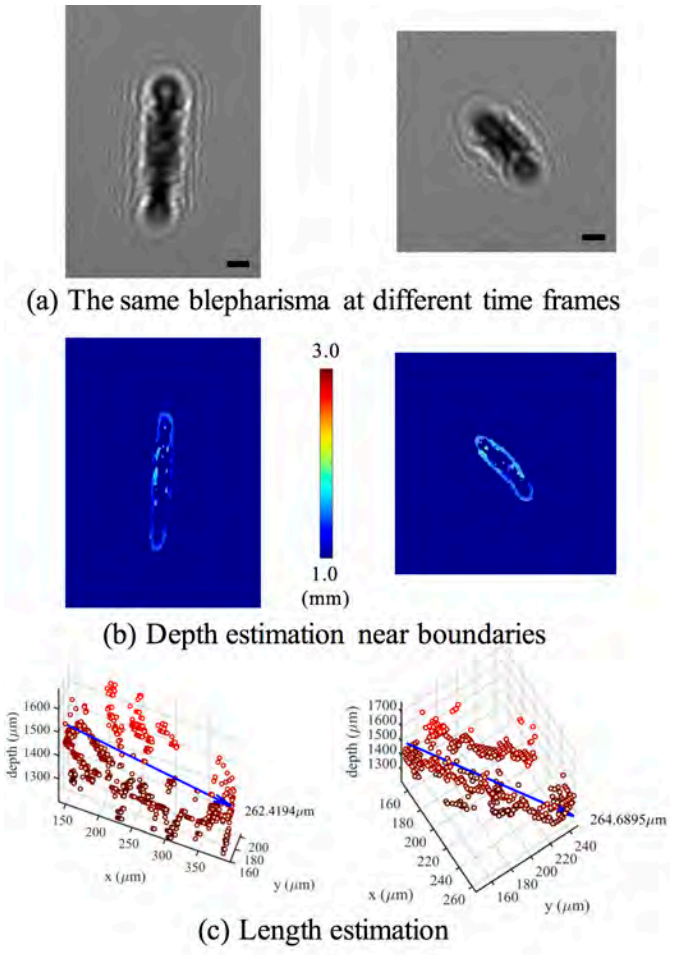


Figure 4.4. Depths (near boundary) and length estimation at different time frames. Shown are the same Blepharisma with a time interval of 0.91s.

different sub-sampling factors. However, as the sub-sampling factor increases, the same auto-refocusing scheme becomes increasingly difficult to apply as the signal turns weak and noisy and no longer preserves high frequency information which results in deteriorated refocusing accuracy. The loss of high frequency information also affects the phase retrieval algorithm because during each iteration, fewer pixels will be used for updating the amplitude at the sensor plane (Fourier domain).

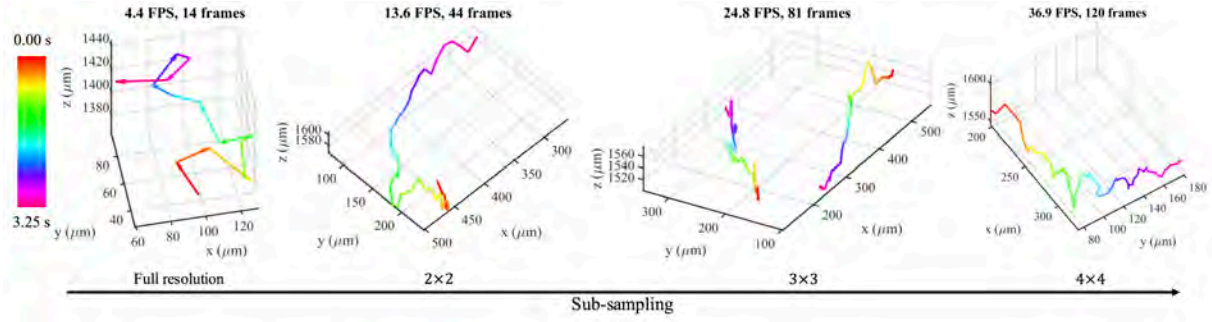


Figure 4.5. Sub-sampling technique improves temporal resolution. An example: 4D tracking of Euglena. Within the same time duration, the trajectory of motion becomes more smooth as sub-sampling factor increases.

4.3.3. Improving spatial resolution for sub-sampled holograms

Performing 3D tracking with high space-time resolution is a challenging task. Temporal resolution can be improved by sub-sampling technique, leaving spatial resolution degraded because only a subset of the full resolution image is sampled. A simple yet effective perspective to solve spatial resolution recovery problem based on OIH setup was recently proposed in [**105**]. Instead of updating the amplitude for "full resolution" (achieved by interpolation, etc.) holograms, in [**105**], an approach was proposed for updating amplitude only on the available (sampled) pixel locations, which is from a subset R of the detector. We refer to this as the sub-sampled phase retrieval (SPR) method. A version of the algorithm with our auto-refocusing scheme is described in Alg. 1. In Alg. 1, the input and output images are represented as n-element vectors. Some empirical parameter choices can be found in [**105**]. The benefit of SPR to 3D tracking is that it recovers the support on the object plane. Note that in Section 3.1, we also showed an improved depth estimation by applying the recovered support from PR.

This type of reconstruction technique, along with [**39**], can be classified as recovering object information directly from substantially fewer measurements of holograms, i.e., low resolution

Algorithm 1 Sub-sampled phase retrieval

Input: $I[n]$, $R[n]$, σ, a, Th, b, w, maxIter
$I[n]$ - intensity of captured hologram
$R[n]$ - sub-sampling boolean mask
σ - standard deviation for Gaussian filter
a - coefficient for adjusting σ every w
Th - threshold for error reduction
b - coefficient for adjusting threshold every w
w - iteration strides for updating support
maxIter - maximum iterations
Output: f[n] - recovered signal with phase information

Initialization: $h_0[n] = \sqrt{I[n]}$,

Depth estimation, d
Ref $= Q_{-d} *$ median$(h_0[n])$

for $i = 0$ to maxIter **do**
    update amplitude of $g_i[n]$ if $R[n]$ = 1
    back-propagation $f_i[n] = Q_{-d} * h_i[n]$
    if $mod(i, w) = 0$ **then**
        update support $S[n]$ (boolean)
        $bg =$ median$(|f_i[n]|)$
        $\sigma = a \cdot \sigma$, $Th = b \cdot Th$
        $s[n] =$ Gaussian$(bg - |f[n]|, \sigma)$
        $S[n] =$ normalize$(s[n]) > Th$
    end if
$f_{i+1}[n] = S[n] \cdot f_i[n] + (1 - S[n]) \cdot$ Ref
forward propagation to sensor plane
$h_{i+1}[n] = Q_d * f_i[n]$
end for
return $f_{i+1}[n]$

(LR) holograms, as is shown in Fig. 4.6, Path 2. Compared to phase retrieval methods applied to high/full-resolution (HR) holograms (Path 1), low-resolution/sub-sampled (LR) holograms can provide temporal resolution enhancement.

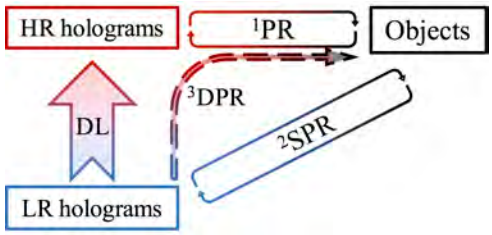


Figure 4.6. Phase retrieval classification. [1]PR: iterative phase retrieval methods based on high resolution (HR) holograms; [2]SPR: sub-sampled phase retrieval method from low resolution/sub-sampled (LR) holograms; [3]DPR: dictionary-based phase retrieval scheme. A dictionary learning (DL) method is introduced in combination with an iterative phase retrieval method in order to overcome space-time resolution tradeoff.

Apart from applying sub-sampled constraints, i.e., SPR, we propose another approach by taking a ”detour”, i.e., first recover HR holograms from LR holograms and second to perform PR on the recovered HR holograms. This approach is described in Fig. 4.6, Path 3. We propose to use a dictionary learning method for the LR-HR recovery. This is based on the consideration that HR holograms, although being acquired at a low frame rate, can serve as a resourceful dataset through a long recording time. Thus, enough features can be extracted from the HR holograms to guide the reconstruction. Another phenomenon that supports this argument is that HR holograms contain important spatial frequency information yet preserving similar visual patterns. Intuitively, holograms are ”blurry” images with fringes, i.e., diffraction pattern. Since the patterns obey the same diffraction rule, we can probably make an assumption that a unique pattern can be fully recovered/represented by a smaller amount of observations (measurements). The latent connections between missing and available observations could be well understood via machine learning/training technique. This curiosity leads to the investigation of finding a feasible computational approach for holographic imaging.

4.4. Sparse representation

The problem in sparse signal representation is to find the sparsest representation possible of a given signal vector $y \in \mathbb{R}^n$, based on an over-complete dictionary $\mathbf{\Phi} \in \mathbb{R}^{n \times m}$, with $m > n$. Each column vector of the dictionary $\phi_i \in \mathbb{R}^n,\ i = 1, ..., m.$ is referred to as an atom. Thus, the sparse representation problem becomes the following optimization problem,

$$\min_{\alpha} ||\alpha||_0 \ s.t. \ \mathbf{y} = \mathbf{\Phi}\alpha \tag{4.7}$$

This is a combinatorial optimization problem. Several variants [**108, 109**] such as convex relaxation have been proposed. In the context of single image super resolution [**110, 111**], this problem can also be formulated as bilevel optimization problem and the relationship between the LR and HR dictionaries is also optimized.

4.4.1. Tracking-based vector quantization

We first examine a classic sparse representation technique, vector quantization (VQ) [**112**], and propose a tracking-based VQ scheme to improve the spatial resolution of the tracked image patch.

Our goal is to recover high resolution images from low resolution observations acquired by $\mathbf{L} = \mathbf{RH}$, where $\mathbf{L}$ and $\mathbf{H}$ are vectorized image patches connected by a sensing matrix $\mathbf{R}$. In the case of periodic sub-sampling, with a sub-sampling factor of $k \times k$, $\mathbf{R}$ becomes a $\lfloor n/k \rfloor \times n$ dimensional matrix, with $\mathbf{R}[j, kj - k + 1] = 1,\ j = 1, ..., \lfloor n/k \rfloor$ and the rest of the elements being zeros. Here, $\lfloor \cdot \rfloor$ denotes the floor of a number. Thus, the reconstruction problem can fit into the formulation of Eq. 4.7 by applying the sensing mask $\mathbf{R}$ upon the dictionary. In this

case, each column in the dictionary ϕ_i represents an HR vector. The sought after representation is $\hat{\mathbf{H}} = \mathbf{\Phi}\alpha$.

We seek to solve the optimization problem based on a constructed dictionary $\mathbf{\Phi}$ with large image patches and finding the "nearest neighbor" in the dictionary based on VQ. This is inspired by a typical type of applications in holographic imaging, i.e., object recognition and tracking with high throughput performance [**72, 113**]. Biological samples such as protozoa are usually small in sizes, e.g., around 20×20 pixels ($2.2\mu m$ pixel pitch) for Euglena, and produce similar holographic patterns. During our characterization of sensor noise, we found that the signal-to-noise ratio (SNR) is reduced as the sub-sampling factor increases. This inspires us to develop a dictionary based on large patch sizes that can cover the entire hologram and is capable of tolerating noise. Another reason to explore this path is that high frequency fringes determine the resolution of the reconstructed object. As is also pointed out in the lower-right panel of Fig. 4.7, the distance from object center to the place that high frequency fringes disappear is about 80 pixels in a typical on-chip setup, i.e., objects are located 1 mm to 3 mm away from the sensor. This leads us to the problem of high-frequency representation based on optimized dictionaries because in order to learn a fully-optimized dictionary, a small number of atom elements is required.

As can be seen in Fig. 4.7, the local descriptors (the locations of key points are framed) for a holographic image are detected and filtered. Extracting local descriptor is a low-level computer vision task and has been well studies and applied for various applications. We used Scale-Invariant Feature Transform (SIFT) [**114**] for feature extraction. Fig. 4.7 also shows a filtering process in order to avoid extracting useless key points such as static points over time and points that are not describing holograms. In the figure, the key points that have the same

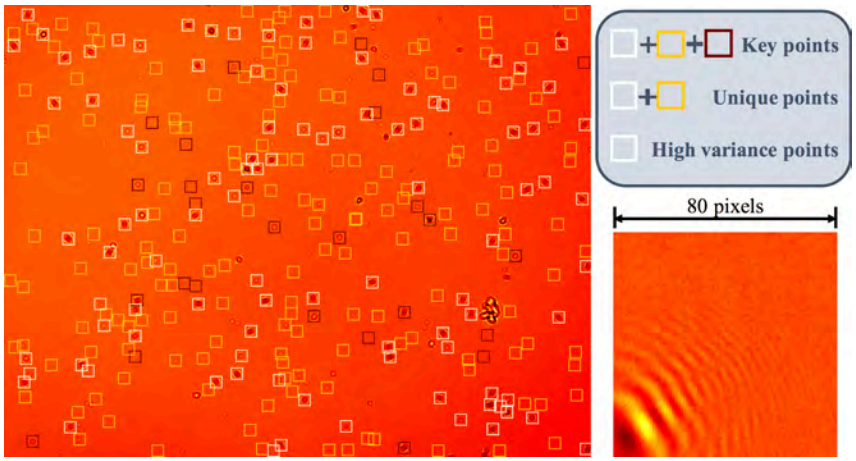


Figure 4.7. Constructing an over-complete dictionary. Left: One frame of full resolution video (Euglena). Detected key points are marked in square frames. First, unique key points between adjacent frames are picked; second, low variance points are filtered out. Lower right: *a quarter* of one atom. The high frequency information can be extended as far as 80 pixels away from key point centers. Thus, 160×160 pixels should be cropped so as to preserve high frequency information of the hologram.

location between adjacent time frames are filtered out. A second step is to apply a variance threshold within a 100×100 window around each key point's location.

In this case, the highly-redundant dictionary is used along with the tracking scheme. We aim to focus on recovering a tracked small hologram patch ($\lfloor 160/k \rfloor \times \lfloor 160/k \rfloor$) from an LR (sub-sampling factor of $k \times k$) captured image. Thus, we focus our testing case only on sample-centered image patches. This extraction requirement can be achieved by applying a filter on the "scale" of each key point. We construct the testing data set from a set of video frames used separately from the set for constructing the over-complete dictionary. Both videos are recorded during a single experiment. We build a testing set of 800 images. Each image patch has size of 160×160 pixels. We use the orthogonal matching pursuit (OMP) algorithm for seeking the sparse representation [**108**]. A pipeline is shown in Fig. 4.8 (b). According to our

experience, the coefficient drops very fast after the first sought coefficient. This is because the atoms in the dictionary are not strictly orthogonal to each other. However, since the dictionary is constructed through careful filtering procedure, we still assume the optimum solution is in the dictionary space. Based on this consideration, instead of applying a strict sparsity one constraint, we applied a stopping criterion when the smallest coefficient to be $\alpha_{min} < 10^{-6}$. As the number of atoms in the dictionary increases, the representation capability improves and becomes stable over sub-sampling factors. This indicates that in this constructed dictionary, the solution yields to a local optimum. This also shows that the local optimum does not change over different sub-sampling factors. However, the representation capability is still constrained in limited recovery performance. As shown in the error map in Fig. 4.8 (b), although the main body of the hologram has been represented, the error is not randomly distributed. This suggests that the latent connection between the LR and HR dictionaries shall be optimized.

4.4.2. Sparse coding via learned bilevel dictionary

In order to avoid the under-optimized connection between LR image and HR dictionary, we use a sparse coding scheme, bilevel coupled dictionary[**111, 110**] to jointly optimize an HR dictionary and its corresponding LR dictionary, as well as the sparse representation coefficients. The optimization problem can be described as below,

$$\min_{\mathbf{D}} \sum_{i=1}^{N} ||\mathbf{x}_i - \mathbf{D}\alpha_i||_2^2 + \lambda||\alpha_i||_i \tag{4.8}$$

$$\min_{\mathbf{D}_l, \mathbf{D}_h} \sum_{i=1}^{N} \frac{1}{2}(||\mathbf{H}_i - \mathbf{D}_h\alpha_i||_2^2 + ||\mathbf{L}_i - \mathbf{D}_l\alpha_i||_2^2) + \lambda||\alpha_i||_i \tag{4.9}$$

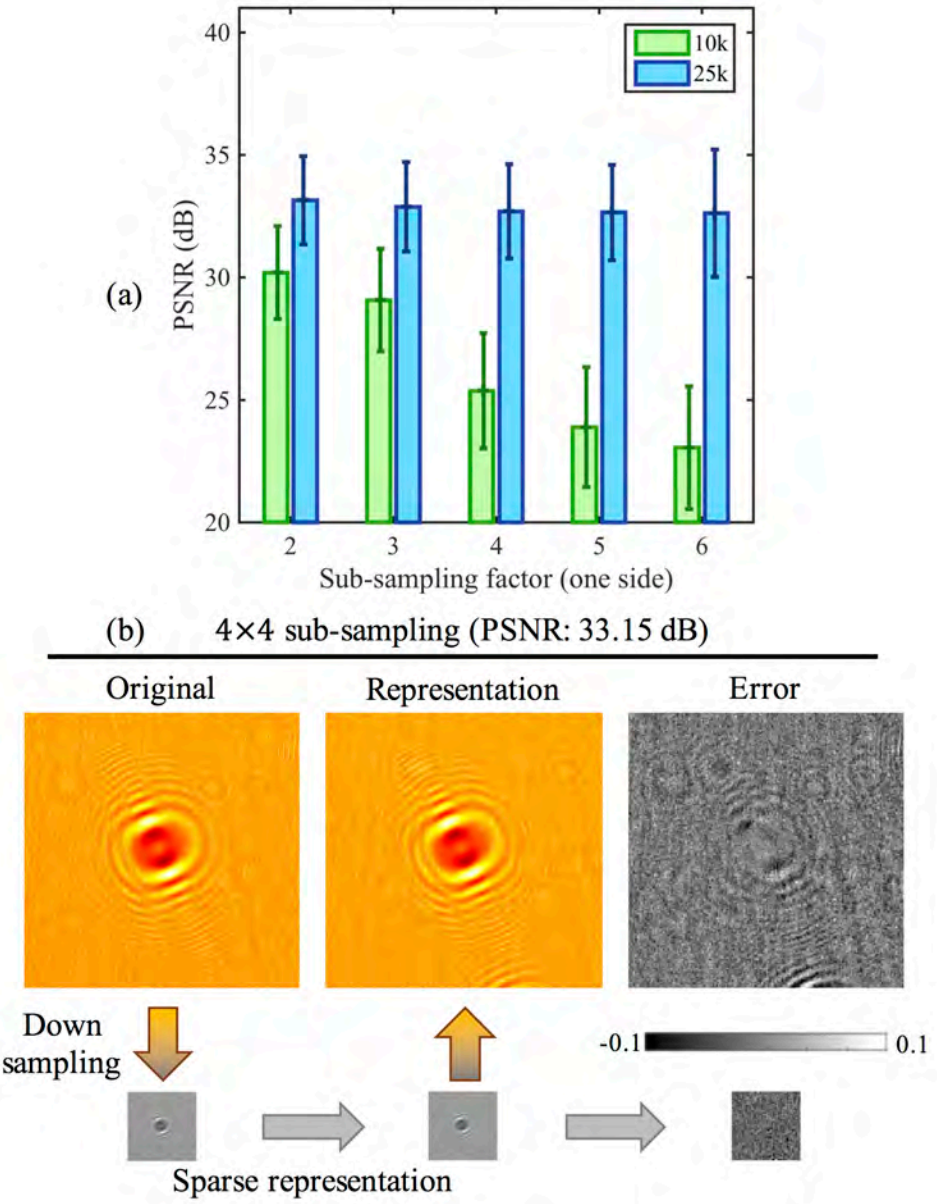


Figure 4.8. Performance of the constructed over-complete dictionary. (a) Comparison of the performance for different atom number. (b) A representation example: 4×4 sub-sampling. The range of the original image and representation image is from 0 to 1.

	2×2	3×3	4×4
5×5 pixels	38.78	N.A.	32.10†
7×7 pixels	39.04	35.67	32.33
9×9 pixels	39.13	36.72	33.13

Table 4.1. Comparison of different patch sizes at different sub-sampling factors. PSNR values are shown in decibel unit. †: tested by applying the 2×2 dictionaries twice.

$$\begin{aligned} &\min_{\mathbf{D}_l, \mathbf{D}_h} \sum_{i=1}^{N} \frac{1}{2} ||\mathbf{H}_i - \mathbf{D}_h \alpha_i^H||_2^2 \\ &\textit{s.t. } \alpha_i^H = \underset{\alpha_i^L}{\operatorname{argmin}} \frac{1}{2} ||\mathbf{L}_i - \mathbf{D}_l \alpha_i^L||_2^2 + \lambda ||\alpha_i^L||_1 \\ &||\mathbf{D}_h(:,k)||_2 \leq 1, ||\mathbf{D}_l(:,k)||_2 \leq 1, k = 1, ..., m \end{aligned} \tag{4.10}$$

Each element of the HR and LR dictionaries obeys the constraint: $||\mathbf{D}_h(:,k)||_2 \leq 1, ||\mathbf{D}_l(:,k)||_2 \leq 1, k = 1, ..., m$. The two dictionaries are coupled so as to share the same sparse coefficient $\alpha_i^H = \alpha_i^L$.

We built the coupled dictionary separately for each sub-sampling factor. In the training phase, we used the previous training video set for acquiring HR image patches (10^6). The LR image patches are obtained by numerically sub-sampling. The number for dictionary atom is 512. In the testing phase, the same 800 image patches were used. Empirically, we found that larger image patch sizes improve the reconstruction performance, as is also shown in Table4.1. However, larger image patch sizes require more atoms and larger number of training patches.

A comparison of different reconstruction algorithms is shown in Fig. 4.9. In this case, the parameters, e.g., distance of propagation, are fixed based on the full-resolution auto-refocusing scheme. The parameters in the phase retrieval algorithm are roughly the same after tuning.

In order to perform a fair comparison, the central 60×60 pixels are cropped from the originally reconstructed 160×160 image and the PSNR and SSIM values are computed based on the cropped images. It can be seen that the dictionary-based methods perform better than sub-sampled phase retrieval algorithm. For the two dictionaries we used, VQ dictionary provides a constantly stable performance. This has been discussed in the previous section. However, the clear contrast of the reconstruction result does not have higher PSNR or SSIM values compared to bilevel dictionary. This indicates the inaccuracy and limitation of an over-complete dictionary.

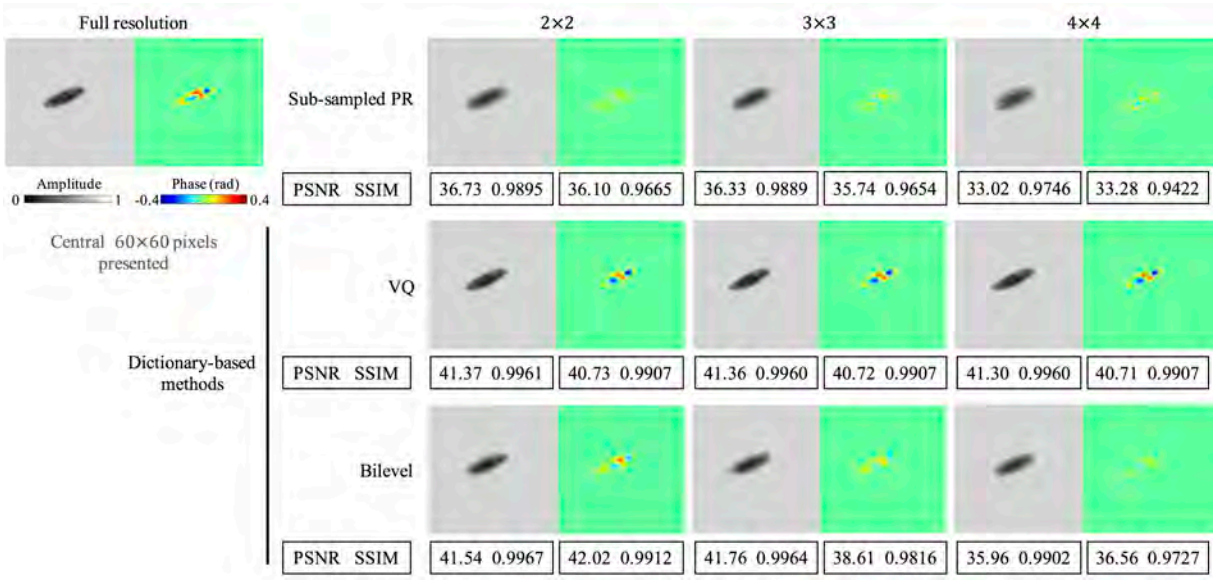


Figure 4.9. A side-by-side comparison between three super resolution algorithms applied to phase retrieval. PSNR is in decibel unit.

Further, we present a reconstruction result based on experimental data, as shown in Fig. 4.10.

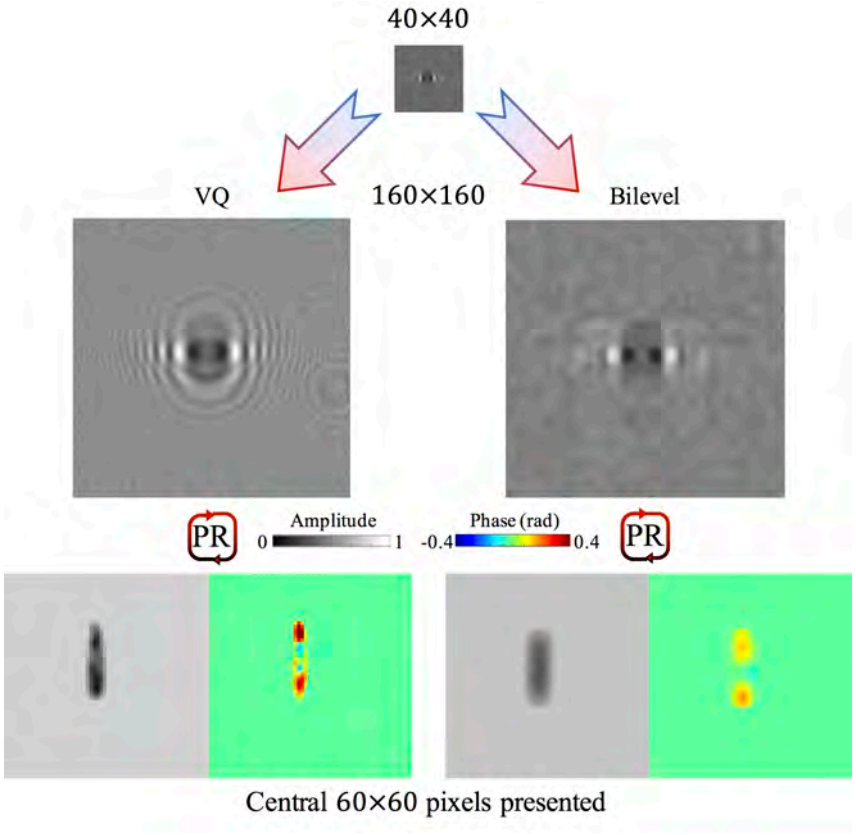


Figure 4.10. Reconstruction. Experimental data acquired from 4×4 sub-sampling.

4.5. Discussion

In this paper, we investigate the imaging performance of OIH and explore computational methods to overcome space-time resolution tradeoff. Our experiments are conducted based on a simple on-chip imaging setup (an LED is used as the light source and a CMOS sensor as the imager). However, this lens-free setup still provides high-resolution ($2.2\mu m$ in our case) and wide field-of-view. We explore the refocusing schemes for imaging extended objects. We designed an auto-refocusing scheme for estimating depth information of the objects. This is useful for integrating into conventional phase retrieval algorithms, especially for high throughput imaging tasks.

We address the space-time resolution tradeoff while performing a tracking task. As compared to previous phase retrieval method (SPR), we proposed two dictionary-based methods for overcoming the spatial resolution loss in the sub-sampling context. Our first method is to build an over-complete dictionary. We show that an over-complete dictionary has a stable performance over different sub-sampling factors. However, we also show that the reconstruction is not globally optimum. The requirement for building an over-complete dictionary is that it has to be used in combination with tracking schemes. This method is not applicable as a general single-frame super resolution method. Another issue is the image patch size. We build the dictionary based on an atom length of 25600. However, this method requires the patch size to cover the entire range of a single hologram. For significantly large objects, such as Blepharisma, this method is not recommended for use. A learned dictionary is optimized via bilevel sparse coding technique. Our result shows an improvement from the previous method. In both cases, the space-time super resolution goal has been achieved.

Our current dictionary-based phase retrieval method is trained and tested on the same scenarios. It is more attractive to develop a general holographic dictionary that is capable of performing super resolution given an arbitrary hologram.

CHAPTER 5

Lens-free Coded Aperture Imaging for Privacy Preserving Action Recognition

In this chapter, we demonstrate a synergistic model without the use of differentiable imaging models. This synergistic model is designed for the privacy preserving action recognition using lens-free coded aperture cameras. We show that deep classifiers face difficulty in directly classifying unconventional data, such as lens-free coded aperture images. Instead, we demonstrate a synergistic strategy by deriving motion features from the coded aperture imaging model, and show that physics-based motion model can improve the performance of deep classifiers.

The risk of unauthorized remote access of streaming video from networked cameras underlines the need for stronger privacy safeguards. We propose a lens-free coded aperture camera system for human action recognition that is privacy-preserving. While coded aperture systems exist, we believe ours is the first system designed for action recognition without the need for image restoration as an intermediate step. Action recognition is done using a deep network that takes in as input, non-invertible motion features between pairs of frames computed using phase correlation and log-polar transformation. Phase correlation encodes translation while the log polar transformation encodes in-plane rotation and scaling. We show that the translation features are independent of the coded aperture design, as long as its spectral response within the bandwidth has no zeros. Stacking motion features computed on frames at multiple different strides in the video can improve accuracy. Preliminary results on simulated data based on a

subset of the UCF and NTU datasets are promising. We also describe our prototype lens-free coded aperture camera system, and results for real captured videos are mixed.

5.1. Introduction

Cameras as monitoring systems inside and outside the home or business is an important area of growth. However, as cameras that are connected online are prone to hacking, with images and videos illegally acquired potentially resulting in loss of privacy and breach of security.

In this paper, we describe initial work on a novel privacy-preserving action recognition system. Our system enhances the preservation of privacy from capture to executing visual tasks, as shown in Figure 5.1. By using a lensless coded aperture (CA) camera, which places only a coded aperture in front of an image sensor, the resulting CA image would be visually unrecognizable and are difficult to restore with high fidelity. Instead of decoding the image as a preprocessing step, which is ill-posed and requires expensive computation if the mask is non-separable, we extract motion features (translation, rotation, and scaling) using the Fourier-Mellin transform and use them as inputs to a deep neural network.

We show that the translation features are invariant to the coded aperture (2D mask pattern) design, as long as its Fourier transform is broadband (*i.e.*, no zeros in the spectral magnitude). Specifically, the term "invariance" refers to the fact that the translational features are only dependent on the type of motion in the scene, not on the choice of the coded aperture design. To promote the invariance property for all features, we design a training mechanism which arbitrarily changes masks for each sample batch and observe performance improvements when testing with a new random mask.

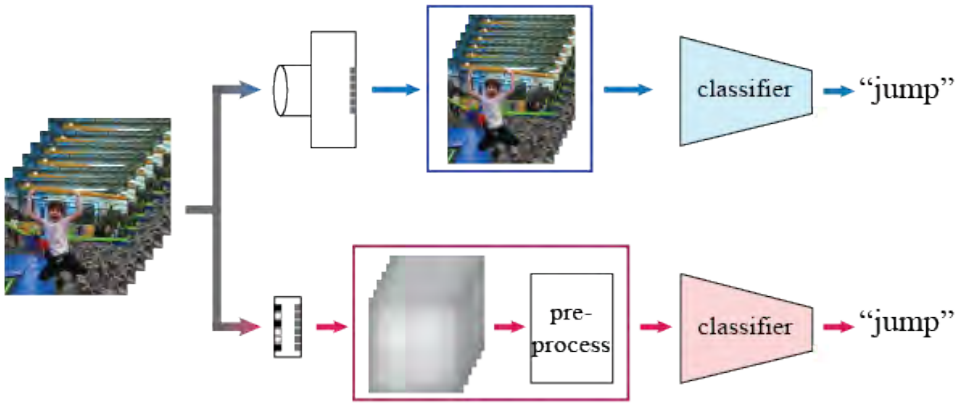


Figure 5.1. **Comparison of action recognition systems**. The conventional system (top) may be vulnerable to a privacy attack by an adversary. Our lensless coded aperture camera system (bottom) preserves privacy by making the video incomprehensible while allowing action recognition.

The "mask-invariant" feature is important for two reasons: (1) training can be done without reliance on a specific coded aperture design, and (2) from a commercial perspective, no two random cameras are likely to have the same coded aperture design, which makes image restoration virtually impossible through reverse engineering.

From a privacy perspective, the CA camera acts as the first layer of privacy protection, as CA images are visually incomprehensible. Our motion features provides a second layer of privacy. These features are based on phase correlation between pairs of video frames, which whitens signal in Fourier space and only leaves motion signal intact. Please note that from here on, we use the terms "coded aperture" and "mask" interchangeably.

5.2. Related work

Our work is multi-disciplinary. The relevant areas are compressive sensing, optics and sensors, coded apertures and action recognition. Here, we briefly survey each area.

5.2.1. Reconstruction-free visual inference

Executing visual tasks without reconstructing the original visual data is an interesting direction for data collected not in the form of visual images/videos as reconstruction problems are usually ill-posed and computationally expensive. One reconstruction heavy scenario is Compressive Sensing (CS), where the measurements are far fewer than required by Shannon-Nyquist requirement [**92**]. Tasks that can be solved by directly processing CS data include optical flow [**115**], dynamic textures [**116**], face recognition [**117, 118**], and action recognition [**119**], *etc.*. Our work considers a similar problem to [**119**], *i.e.*, performing action recognition without reconstructing images. In the smashed filters approach, every frame of the scene is compressively sensed by optically correlating random patterns with the frame to obtain CS measurements. Therefore, the approach requires multiple sequential frame capture and a DMD array (which is costly and has fragile moving parts). Our approach uses a single coded aperture camera. Reconstruction-free methods do not reveal the appearance of the scene and can therefore safeguard privacy in sensitive environments.

5.2.2. Privacy-preserving optics and cameras

Optics and imaging sensors. There are imaging sensors and modalities whose direct output is not visually recognizable. This achieves the purpose of privacy preservation at the optics/sensor level. A popular approach for preserving privacy is by defocusing [**120**]. Alternative optical solution is to put optical elements in front of sensors, *e.g.*, cylindrical lens [**121**], diffraction gratings [**122**], or diffusers [**123**] in front of the sensor. Recovery of these images requires careful calibration of the imaging system and adequate computation.

Firmware. Sensor firmware can be modified to protect privacy before or during the sensing process. For example, in PrivacyCam [**124**], regions of interest are first identified based on background subtraction before being encrypted using AES. Other implementations involve embedding watermarks into the captured data [**125, 126**].

Coded apertures. Coded aperture imaging originates from the field of astronomical X-ray and gamma-ray imaging in the 1960s [**127, 128, 129**]. By extending pinholes to cameras with masks consisting of designed patterns, coded apertures has been used for eliminating issues imposed by lenses and has found novel applications in extending depth-of-field [**130, 131**], extracting scene depth and light fields [**132, 133, 134**], and miniaturizing camera architectures [**135, 136**]. Unlike conventional RGB images, lensless coded aperture images obfuscates visual features familiar to human. Our work is inspired by this distinctive effect. We explore the feasibility of using coded aperture data to execute visual tasks such as action recognition, for the purpose of preserving privacy.

5.2.3. Privacy-preserving action recognition

Action recognition is a long-standing computer vision task with wide applications in video surveillance, autonomous vehicles and real-time patient monitoring. Early approaches use handcrafted motion features, *e.g.*, HOG/HOF [**137**] and dense trajectories [**138**]. Recent works utilize two input streams for appearance and motion [**139**] and 3D CNN architectures [**140**] to learn spatio-temporal features [**141**]. State-of-the-art approaches for video-based action recognition require both appearance and optical flow based motion features. These systems are training on large video datasets, *e.g.*, ImageNet and Kinetics.

Privacy-preserving action recognition is becoming important due to the risk of privacy breaches in surveillance systems in sensitive areas such as healthcare. Approaches that use multiple extremely low resolution cameras have been explored [**142, 143**]. Recently, Ren *et al.* used adversarial training to anonymize human faces in videos, without affecting action recognition performance [**144**]. Furthermore, adversarial learning has been explored to jointly optimize privacy attributes and utility objectives [**145, 146, 147**].

5.3. Image formation for coded aperture camera

We consider a lens-free coded aperture imaging architecture, where a planar coded aperture (mask) is placed in front of an imaging sensor. The encoding mask can be considered as an array of pinholes located at various lateral locations. The acquired image $\boldsymbol{d}$ can be numerically modeled as a convolution between the object image $\boldsymbol{o}$ and the point spread function (PSF) $\boldsymbol{a}$, *i.e.*,

$$\boldsymbol{d} = \boldsymbol{o} * \boldsymbol{a} + \boldsymbol{e}, \tag{5.1}$$

with $\boldsymbol{e}$ being noise. The convolution is applicable if the mask is far enough from the sensor, such that each sensor pixel is able to see the entire mask pattern. If the mask-sensor distance is small (as in the case of FlatCam [**136**]), the mask design should consist of a smaller pattern replicated in a 2D array. The size of the smaller pattern should be such that each sensor pixel sees a version of it locally. Then the output can be considered a result of convolution.

We first implement the convolution based on FFT, which we refer as the *without* boundary effect (BE) version. However, we observe that real CA images have boundary effect. We then incorporate boundary effect by zero-padding both image and mask. The FFT-based convolution remains the same. We then crop to the original size after convolution. This would generate

simulated CA frames that are more consistent with ones captured with a real camera. However, this procedure is significantly more computationally expensive. In experiments, we use the *without* BE version for analysis of the motion features and optimizing feature representation as DNN input, and both versions are used for final testing.

5.4. Extraction of motion features

In this section, we describe how we compute features for action recognition *without* having to first restore the images from a lenless coded aperture camera. We refer to them as *TRS (translation, rotation, scale) features*. They are computed from pairs of frames captured at different moments in time.

5.4.1. Translational (T) features

Phase correlation was used first for global image registration [**148**] and then for motion/flow estimation [**149, 150**]. Compared to other motion estimation methods [**151**], phase correlation has the advantages of being computational efficient and invariant to illumination changes and moving shadows. Additionally, from a privacy point-of-view, operating in the frequency domain, rather than the original domain, provides a natural opportunity for executing visual tasks without retrieving the original data. We show how phase correlation can be used to characterize motion in coded aperture observations without knowing the mask design.

Assume there exists a translation between two video frames:

$$\boldsymbol{o}_1(\mathbf{p}) = \boldsymbol{o}_2(\mathbf{p} + \Delta\mathbf{p}), \tag{5.2}$$

where $\mathbf{p} = [x, y]^T$ and $\Delta\mathbf{p} = [\Delta x, \Delta y]^T$ are the spatial coordinates and displacement, respectively.

In frequency domain, translation gives rise to a phase shift:

$$\mathcal{O}_1(\,^\circ) = \phi(\Delta\mathbf{p})\mathcal{O}_2(\,^\circ), \tag{5.3}$$

where $\nu = [\xi, \eta]^T$ and $\phi(\Delta\mathbf{p}) = \exp^{i2\pi(\xi\Delta x + \eta\Delta y)}$. ξ and η are the frequency coordinates in Fourier space. $\mathcal{O}_1$ and $\mathcal{O}_2$ represent Fourier spectra of $\boldsymbol{o}_1$ and $\boldsymbol{o}_2$. By computing the cross-power spectrum and taking an inverse Fourier transform, the translation yields a delta signal:

$$\mathcal{C}_o(\xi, \eta) = \frac{\mathcal{O}_1^* \cdot \mathcal{O}_2}{|\mathcal{O}_1^* \cdot \mathcal{O}_2|} = \phi^* \frac{\mathcal{O}_2^* \cdot \mathcal{O}_2}{|\mathcal{O}_2^* \cdot \mathcal{O}_2|} = \phi(-\Delta\mathbf{p}), \tag{5.4}$$

$$\boldsymbol{c}(\mathbf{p}) = \delta(\mathbf{p} + \Delta\mathbf{p}). \tag{5.5}$$

The translation can be located by finding the peak signal; this feature is the basis of the original work [**148**], assuming a single global translation. Multiple translations result in an ensemble of delta functions. Note that these two equations are critical to our technique, as they show that computing translation is *independent of the coded aperture design*, as long as they have a broadband spectrum. Instead of finding the peak signal, we make full use of the computed image, treating it as a translation map (T-map).

5.4.2. T features independent of coded apertures

The convolutional transformation that generates a CA image encodes local motion in the original video to global motion in the resulting CA video. This makes the localization of the motion very challenging without restoration. However, we demonstrate that the global translation can

still be retrieved using phase correlation, and is *independent* of the mask design, as long as they have broadband spectrum. Following Eqs. (5.1) and (5.3), a translation relationship ($\Delta\mathbf{p}$) also exists:

$$\mathcal{D}_1(\nu) = \mathcal{O}_1 \cdot \mathcal{A} = \phi\mathcal{O}_2(\nu) \cdot \mathcal{A} = \phi\mathcal{D}_2(\nu), \tag{5.6}$$

where $\mathcal{A}$ denotes the Fourier spectrum of mask $\boldsymbol{a}$. The cross-power spectrum is then

$$\mathcal{C}_d(\nu) = \frac{\mathcal{D}_1^* \cdot \mathcal{D}_2}{|\mathcal{D}_1^* \cdot \mathcal{D}_2|} = \phi^* \frac{\mathcal{O}_2^* \cdot \mathcal{A}^* \cdot \mathcal{A} \cdot \mathcal{O}_2}{|\mathcal{O}_2^* \cdot \mathcal{A}^* \cdot \mathcal{A} \cdot \mathcal{O}_2|} \simeq \mathcal{C}_o. \tag{5.7}$$

Note that phase correlation has a magnitude normalization procedure while computing the cross-power spectrum. This step can effectively whiten the spectrum so as to eliminate global changes in appearance. This property provides an additional layer of privacy protection. In our implementation, we add a small number ϵ in the denominator of Eq. (5.7) to prevent division by zero. Regardless, the object spectrum will be unstable if $\mathcal{A}$ has near-zero elements.

5.4.3. Translation size, object size and noise

We perform numerical evaluation of the translation features. In order to quantitatively analyze the motion feature, we synthesize the motion based on background (21) and human pose (16) images collected from the Internet. Since our goal is towards indoor scenarios, the background images are all indoor scenes. We collect various types of single human pose images, with and without object interactions. All the pose images are masked according to the pose shape. Examples are shown in Figure 5.2. In each simulation case, a pose image is placed on top of the background image. In the computed translation image, two dominating spike signals are observed. One is located at the image center, representing the background static signal. The

Figure 5.2. Examples of background and human pose images.

other one is the translation signal of interest. We use the ratio between the translation signal and the background signal as metric. The signal-background ratio (SBR) is evaluated using two parameters, *i.e.*, translation pixel number and the size of the moving object. The results are shown in Table 5.1. For relatively large object size, the average SBR increases as the translation step increases. However, the 5% case observes decreasing trend. Horizontally, as the object size increases, the SBR increases as well. Note that some pose images (2-5) exceed background image size at {15%, 20%} moving at {6, 8, 10} pixels and are not included.

We next characterize the performance across different noise level. We repeat the experiment in Table 5.1 with different levels of Gaussian noise. As expected, the translation signal peak

	5%	10%	15%	20%
2	.978 / .268	1.32 / .323	1.61 / .453	1.82 / .577
4	.979 / .267	1.35 / .331	1.68 / .473	1.89 / .593
6	.976 / .266	1.34 / .336	1.68 / .486	1.88 / .656
8	.988 / .265	1.36 / .339	1.70 / .508	1.90 / .693
10	.979 / .263	1.37 / .344	1.71 / .522	1.93 / .734

Table 5.1. Averaged Signal Background Ratio (SBR). Row: translation size in pixels. Column: object size in percentage. Format: *without* BE / *with* BE.

decreases rapidly as noise increases. *However, we observe that the SBR increases as noise increases*. This indicates that the noise deteriorates both the translation signal and the background signal. But the background signal experience more severe degradation. The results are summarized in Figure 5.3. For visualization purpose, we use log scale for the two axes. The original values are normalized with respect to the zero-noise case.

5.4.4. Coded aperture design

We focus on 2D intensity binary mask patterns as they enable practical implementations. As shown in Figure 5.4, the randomness in the mask pattern, which result in broadband spectra, preserves the T features compared to the T map computed from RGB frames. Figure 5.4 show representative masks that are considered. The pseudorandom mask (mask 1) provides a relatively uniform magnitude distribution. The separable mask (mask 2) based on maximum length sequence (MLS) have much stronger frequency response along the horizontal and vertical axes. Mask 3 is a round aperture and has undesirable dropoffs at higher frequencies. We use pseudorandom masks in our evaluation.

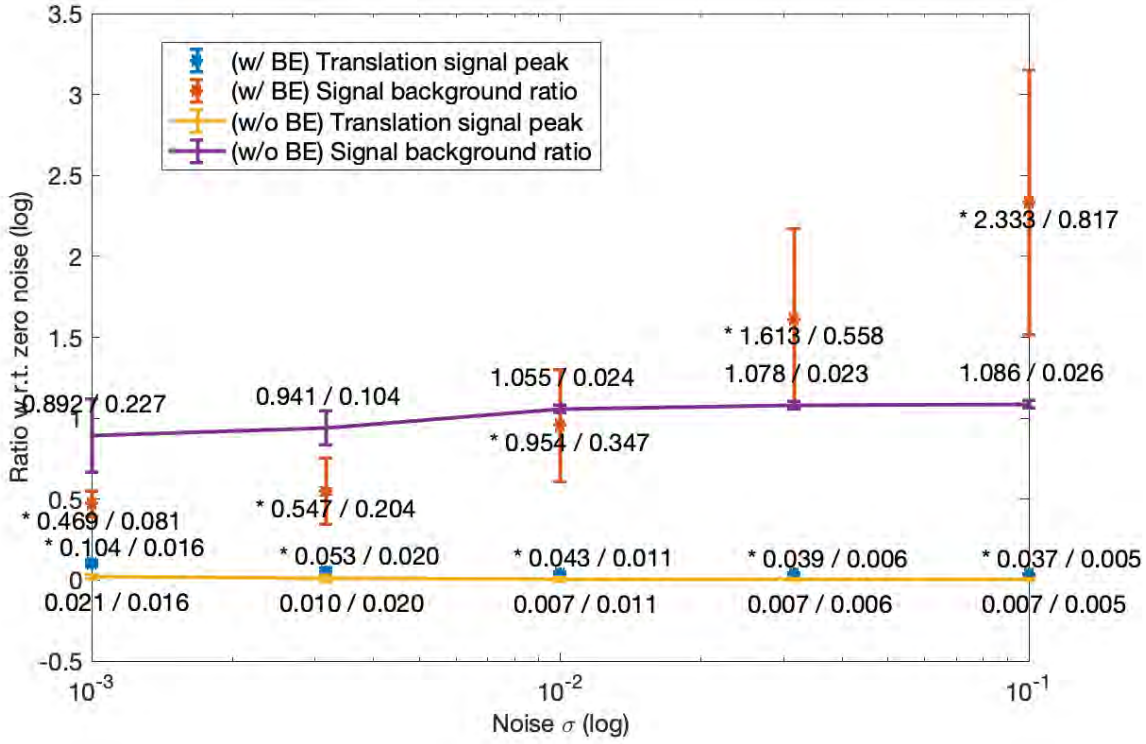


Figure 5.3. Noise characterization. Values are normalized with respect to the zero-noise case. Data format: average value / standard deviation.

Note that since these masks are spatially as large as the image and non-separable in x and y (except row 1), high fidelity image restoration would be difficult and computationally-expensive [**130**]. We did not implement a restoration algorithm for these reasons.

We will show later that using only T features is less effective for action recognition (Figure 5.5). We investigate two extensions of the T features, namely rotation and scale features, and multiple strides.

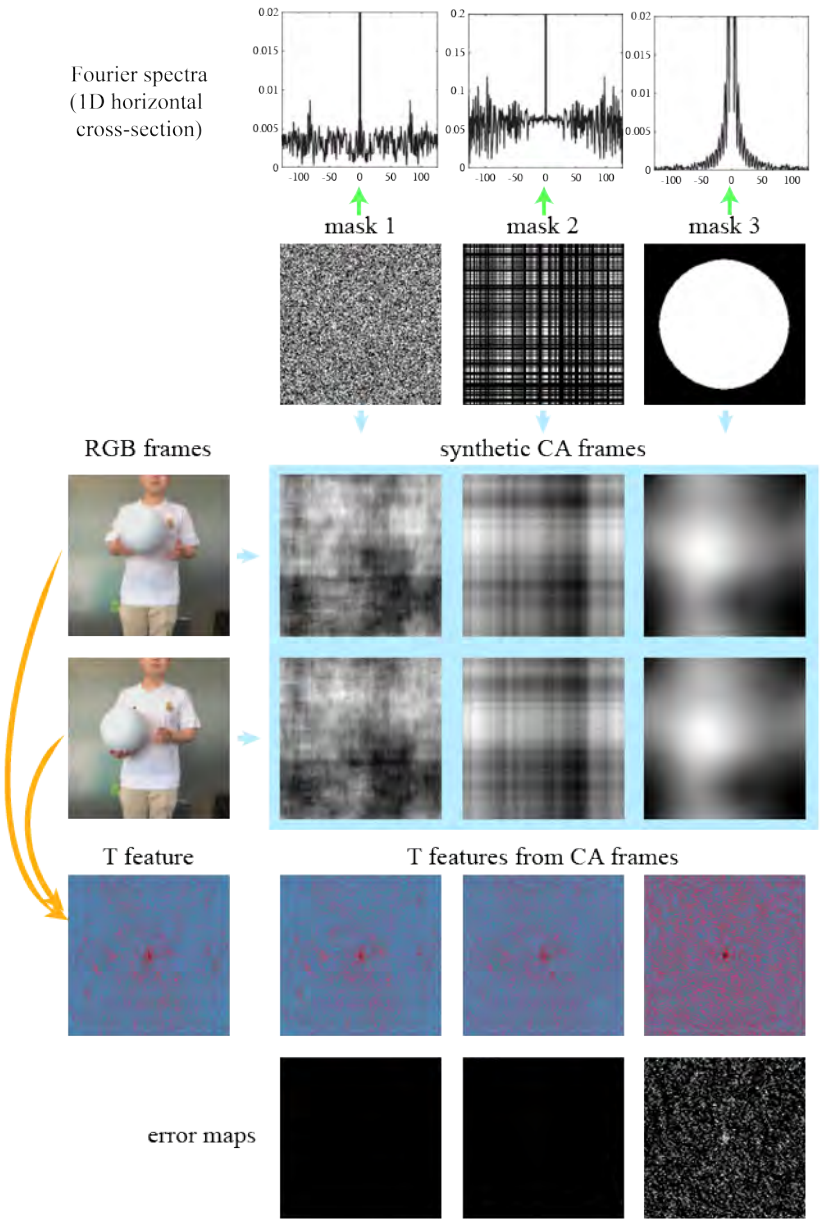


Figure 5.4. T features from different CA observations. 3 different mask patterns (all 50% clear) are investigated (Row 2). Row 1 shows the cross-section of Fourier spectra. Rows 3 and 4 show example RGB images and their corresponding synthetic CA frames (*withoout* BE). For clarity, the intensity of CA frames is rescaled to (0, 1), original contrast is approximately $1.007 : 1$; T feature maps are normalized and γ corrected ($\gamma = 0.4$). Row 5: T feature maps based on Eq. (5.7). Row 6: error maps, with the "ground truth" being the T map for RGB frames. $\epsilon = 10^{-3}$.

5.4.5. Rotation and scale features in log-polar space

Given global translation, rotation, and scaling, we have $\boldsymbol{o}_1(\mathbf{p}) = \boldsymbol{o}_2(s\mathbf{R}\mathbf{p} + \Delta\mathbf{p})$, where s is a scaling factor and $\mathbf{R}$ is a rotation matrix with angle $\Delta\theta$. Translation $\Delta\mathbf{p}$ can be eliminated by taking the magnitude of the Fourier spectrum,

$$|\mathcal{O}_1(\nu)| = |\mathcal{O}_2(s\mathbf{R}^\circ)|. \tag{5.8}$$

If we treat the Fourier spectra as images and transform them into log-polar representations, i.e., $\mathbf{p} = [x, y]^T \Rightarrow \mathbf{q} = [\log(\rho), \theta]^T$, rotation and scaling become additive shifts on the two axes

$$|\mathcal{O}_1(\mathbf{q})| = |\mathcal{O}_2(\mathbf{q} + \Delta\mathbf{q})|. \tag{5.9}$$

This enables us to use phase correlation once again to locate rotation and scale. Note that the mask invariant property is not preserved in RS space. This is because the mask spectrum contributes to a strong static signal to the observed images. However, we later show that the mask-invariant property for RS features can be realized by training with varying random masks.

5.4.6. Multi-stride TRS (MS-TRS)

We make a further extension to compute TRS features based on multiple strides in each video clip. This is to account for varying speeds of motion. For a video clip with length l, the TRS features in stride s are computed by:

$$T_i^{(s)}, RS_i^{(s)} = \mathcal{TRS}\{\boldsymbol{d}_{i\times s}, \boldsymbol{d}_{i\times s+s}\}, \tag{5.10}$$

where $i \in \{0, 1, ..., \lfloor \frac{l-s}{s} \rfloor + 1\}$ denotes all the possible consecutive indices within length l. For example, if a video clip of length 13 is given, the resulting $s2$ TRS features have 12 channels, 6 for T, and 6 for RS. In our case, we compare evaluation results for strides of 2, 3, 4, 6, with clip lengths of 13 and 19.

5.5. Experimental results on simulated data

We now report the results for the following experiments:

- We compare the performance of our method based on CA videos with a baseline that uses regular videos.
- We evaluate the performance of our method when the proposed T, TRS, and MS-TRS features are used.
- We compare the effect of using the same versus different or varying masks on training and validation data.
- We also compare the effect of using different MS-TRS configurations. This experiment is used to select an appropriate configuration for the final evaluation.
- We report results for the best MS-TRS configuration.

We first describe the datasets and protocols used.

Datasets. We have evaluated our approach on the UCF-101 [**152**] and NTU [**153**] datasets. UCF-101 [**152**] contains 101 action classes with 13k videos. In our initial evaluation, we focus on indoor settings (more important from a privacy standpoint). Therefore, we created four different subsets from the 101 classes by selecting actions relevant to indoors.

- UCF-05: Writing on board, Wall pushups, blowing candles, pushups, mopping floor;

- UCF-body (09): Hula hoop, mopping floor, baby crawling, body weight squat, jumping jack, wall push up, punch, push ups and lunges;
- UCF-subtle (13): Apply eye makeup, apply lipsticks, blow dry hair, blowing candles, brushing teeth, cutting in kitchen, mixing batter, typing, writing on board, hair cut, head assage, shaving beard, knitting;
- UCF-indoor (22): combination of UCF-body and UCF-subtle.

We also use the NTU [**153**] dataset which contains videos of indoor actions. We choose this dataset as it collects data using stationary cameras (we handle only static background for now). From our initial evaluation, we found that our proposed approach is better suited for more significant body motions. Because of this, we choose ten classes (with a mix of whole and partial body motions) for our final testing. Eight classes come from the NTU dataset and two classes are from the UCF dataset.

5.5.1. Protocol

Definitions. We use letters s and l to denote the stride and length of a video. For example, $s1, l4$ denotes four consecutive video frames. The number of input channels depends on the training mode.

Training and Validation. We use the first official train/test split from the UCF dataset and randomly select 20% of the training set for validation. Both the training and validation data is expanded using data augmentation to prevent over-fitting. The data augmentation process is as follows.

- gray clips: Each video frame is loaded in as grayscale image at a resolution between 224 and 256. The aspect ratio is fixed at (240×320). The clip is then vertically flipped with 50% chance. A $(224 \times 224 \times l)$ clip is then cropped and used as input.
- CA clips: Each CA clip first experiences the same augmentation step as gray clips. The CA simulation is computed at the resolution of 256×256 and rescaled back to 224×224. We simulate CA observations by computing element-wise multiplication in Fourier space between the Fourier transforms of the image and the mask kernel. We did not implement boundary effect for computation consideration. The diffraction effect is not accounted for as we observe minimal impact on the TRS features. Another reason is that simulating PSF for non-separable masks by matrix multiplication [**130**] is expensive.
- T features: The T features are generated from CA clips at the resolution of 256×256. The central 224×224 area is cropped as input. An l-frame CA clip results in $(l-1)$ T channels.
- TRS/MS-TRS features: In the TRS setting, the T features follow the same cropping. For RS, the R-axis uses center cropping while the S-axis is downsized to 224. An l-frame CA clip results in $2l$ channels, with l T channels and l RS channels stacked together. For MS-TRS, the resulting channels depend on the selected strides.

We use a batch size of 16 or 32. Each epoch, for both training and validation, prepares samples randomly from approximately 20% of all the possible frame combinations. 50 Epochs are used in our evaluation experiments. The percentage of accurate samples is reported. When reporting, we compute the running average accuracy of 5 epochs for better visualization.

Testing. During testing, we resampled each video at 3 spatial scales ($\mu \times \mu$ pixels, with $\mu = 224, 256, 300$) and 5 temporal starting frames evenly distributed across the video length. For example, using MS-TRS-$s346$-$l19$ configuration, a video with 100 frames will be used to generate five clips, starting at frames 1, 21, 41, 61, and 81, with each clip being 19 frames long. Each clip will be used to compute MS-TRS at three spatial scales. The final score for each video is computed by averaging the scores of the 15 clips.

Others. We use the VGG-16 CNN architecture, which contains approximately 134 million parameters. Adam optimizer is used with learning rate 0.0001, $\beta_1 = 0.9, \beta_2 = 0.999$. Since the CA observation is computed on-the-fly, we can change the underlying masks used in each batch. In this paper, we use "m1/m1" to refer to the setting where training and validation using the same fixed mask and "m1/m2" to refer to when training and validation uses two different masks. Finally, "dm1/dm2" denotes the setting where training and validation is done using variable masks. A pseudo-random binary mask is randomly generated for each batch. Note that the mask is fixed for all frames of a single video.

5.5.2. Initial evaluation (without BE)

The goal of our initial evaluation is to validate our proposed training framework, as well as to find the optimal feature representation. Such experiments are implemented using CA simulations *without* BE, as accounting for boundary effect is computationally more expensive.

Baselines. We first train one network on the original videos and three networks on the simulated CA videos as our four baselines. See the results in Table 5.2. The top-1 classification accuracy of 95% (row 1) for the original videos is our upper bound of what we can expect. The

	training	validation
gray video	99.56 (99.86)	94.39 (95.91)
CA (m1/m1)	79.06 (92.65)	63.21 (86.96)
CA (m1/m2)	94.66 (95.17)	27.95 (40.55)
CA (dm1/dm2)	34.93 (36.61)	27.23 (36.96)

Table 5.2. Baseline comparison for UCF-05. Here, for the CA cases, training and validation are done directly on CA videos. The numbers are: average accuracy % of the last 5 epochs (maximum accuracy %). All clips have length 3.

performance of the baselines trained directly on CA videos (rows 2 to 4), will serve as our lower bounds. We expect our proposed features, which involve computation based on CA, to perform better than CA. The CA baselines show instability even when training and validation phases have the same mask. The network corresponding to the second row suffers from overfitting. Changing training masks for each batch does not improve the performance.

Variable masks during training. Our goal is to maximize the robustness of the designed features to the mask patterns. In order to achieve this, we change the training and validation masks by randomly generating a pseudo-random mask during each batch. We compare this dynamic training mechanism with two other modalities, *i.e.*, (1) training and validation using the same mask (m1/m1) and (2) training and validation using two different masks, no mask variation during training (m1/m2). The results are presented in Figure 5.5.

For T features, the validation accuracy plateaus at about 60%. Dynamic training with variable masks does not improve the accuracy. This supports the fact that T features are invariant to the choice of masks.

For TRS and MS-TRS features, using the same stride and length of the clips, the performance improves to around 70% for m1/m1. However, since the RS features are not mask-invariant, validation using a different mask does not have the same accuracy. Varying the

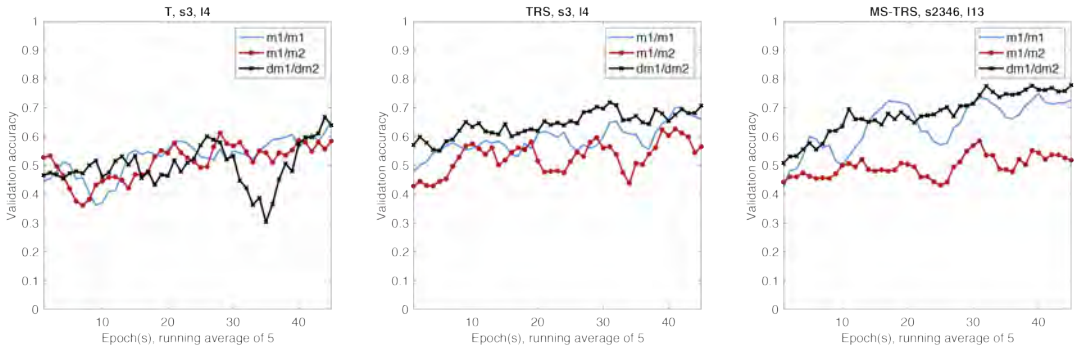


Figure 5.5. Comparison of validation accuracy for UCF-05, with training and validation: using the same mask (m1/m1), using two different masks (m1/m2), and based on a random mask per batch and a different random mask for validation (dm1/dm2). Note: s3 = stride of 3, s2346 = strides of 2, 3, 4, and 6.

masks during training does not improve the performance compared to training using the same mask. This is an interesting effect as, theoretically, the RS features do not have the same mask-invariant property. This drawback appears to be mitigated by changing the masks during training. This, in turn, enables us to test using an arbitrary mask. MS-TRS trained with varying mask achieves the highest validation accuracy 77.8%.

Strides and clip length. In the case of TRS, we found that increasing the strides and clip lengths can improve the performance. The results are summarized in Table 5.3. In this case, the same mask was used during training and validation (m1/m1).

We evaluated different combinations of MS-TRS features. The training and validation for MS-TRS is under dm1/dm2 mode. The results are summarized in Table 5.4. For the same video length, using larger strides improves validation accuracy. For the same stride setting, *e.g.*, $s346$, processing more video frames improves performance. However, using longer stride and longer video, such as *i.e.* $s46, l19$, suffers from overfitting. The combination $s2346, l19$ is not evaluated as generating the 44-channel input on-the-fly becomes computationally expensive.

		training	validation
$ch6$	$s1$	97.23	82.33
	$s2$	97.44	82.49
	$s4$	98.66	85.16
$s2$	$ch4$	97.76	81.78
	$ch6$	97.44	82.49
	$ch10$	98.87	85.56

Table 5.3. Comparing performance of different strides and lengths of video, for TRS, m1/m1 on the UCF-05 dataset. The numbers are maximum accuracy percentages within the first 50 epochs. ch denotes the number of input channels.

	input shape	training	validation
$s2346, l13$	(224, 224, 30)	96.67	83.59
$s346, l13$	(224, 224, 18)	93.69	83.66
$s46, l13$	(224, 224, 10)	92.94	86.59
$s346, l19$	(224, 224, 26)	96.00	86.26
$s46, l19$	(224, 224, 14)	89.91	79.23

Table 5.4. Comparison of training and validation performances for MS-TRS, dm1/dm2 for UCF-05. Numbers are max accuracy percentage within the first 50 epochs.

	UCF-body	UCF-subtle	UCF-indoor
$s346, l13$	88.4 / 81.2	84.9 / 73.2	84.8 / 70.8
$s346, l19$	**90.5 / 83.4**	**86.1 / 76.4**	**88.6 / 72.8**
$s46, l13$	89.9 / 79.1	80.9 / 66.5	83.8 / 66.3

Table 5.5. Training and validation accuracies on different UCF subsets for networks trained on different MS-TRS configurations. UCF-body, UCF-subtle and UCF-indoor has 9, 13 and 22 classes respectively.

More action classes. We selected three MS-TRS settings from Table 5.4 and then trained networks for three larger datasets. These datasets are also subsets of UCF-101 actions focused on indoor settings and include body motions and subtle motions which primarily involve hand & face. The evaluation results are shown in Table 5.5.

5.5.3. Testing results

Based on the experiments on the UCF subset datasets, we selected *i.e.*, MS-TRS-$s346$-$l19$ as the best feature representation. Next, we computed MS-TRS-$s346$-$l19$ features on the 10-class combined dataset of NTU and UCF to examine the feasibility of our representation for daily activities. We used about one-sixth of the NTU videos for the eight classes for training to ensure we have a similar number of training examples as for the two UCF classes. In training phase, each class consists of 100 videos with more than 10K frames. We use a different data augmentation scheme for the NTU dataset. Each NTU video is loaded at random height resolution between 460 and 520. The aspect ratio is fixed at $1080 : 1920 = 9 : 16$.

The central 240×320 region (same as the UCF classes) is cropped and used to compute CA and MS-TRS. For testing, each NTU video is loaded at 522×928 resolution. The central 256×256 video is cropped and used to compute CA and MS-TRS at different scales as described in the testing protocol.

For synthetic CA testing, the overall top-1 accuracy is 60.1% *without* BE and 35.5% *with* BE. The top-1, 2, 3 accuracies for each class is reported in Table 5.6. The results indicate a large variation across classes. Our trained model is able to correctly recognize body motions such as hopping and staggering but is less accurate at differentiating between subtle hand motions such as clapping and hand waving.

5.6. Experimental results on real data

5.6.1. Prototype

To validate our ideas, we built an imaging system as shown in Figure 5.6. Our system consists of a monochrome board-level imaging sensor (XIMEA MQ042, 2048×2048) and a spatial

	class	top-1	top-2	top-3
1	hopping	97.1 / 97.1	100 / 97.1	100 / 100
2	staggering	94.3 / 65.7	97.1 / 91.4	100 / 100
3	jumping up	91.4 / 0.00	97.1 / 71.4	97.1 / 88.6
4	JJ †	81.1 / 16.2	91.9 / 83.8	100 / 91.9
5	BWS †	76.7 / 33.3	86.7 / 73.3	93.3 / 90.0
6	standing up	57.1 / 20.0	88.6 / 40.0	94.3 / 54.3
7	sitting down	51.4 / 11.4	82.9 / 22.9	100 / 31.4
8	throw	31.4 / 20.0	57.1 / 48.6	68.6 / 80.0
9	clapping	11.4 / 20.0	14.3 / 68.6	31.4 / 77.1
10	hand waving	5.70 / 71.4	14.3 / 88.6	20.0 / 88.6
	average	60.1 / 35.5	73.4 / 68.6	80.8 / 80.2

Table 5.6. Testing results for combined NTU and UCF 10 classes dataset. Data format: accuracy % *without* BE / *with* BE. BWS: body weight squats; JJ: jumping jack. † indicates the class comes from UCF dataset, others are from NTU dataset. Ranking according to top-1 accuracy *without* BE.

light modulator or SLM (LC2012, 1024×768) sandwiched between two polarizing filters. The distance between the sensor and SLM is approximately 6mm. The pixel size for the XIMEA camera is 5.5um while that for the SLM is 36um. A long-pass filter is required to remove light frequencies that have low extinction factors with the SLM-filter combo. In addition, we use a cover with a square opening ($12\text{mm} \times 12\text{mm}$) to cut out stray oblique rays and reduce inter-reflection on the side walls between the SLM and sensor.

5.6.2. Testing results

We collect several CA videos using our prototype system and test using both models *with* and *without* BE. These models are trained on a subset of NTU and UCF data as discussed in Section 5.5.3. Each testing video consists of 100 consecutively captured frames. Quantitative results are shown in Table 5.7. We observe that "body weight squats" is a dominating class, and has been correctly classified. Other classes such as "jumping jack" and "standing up" are only correctly

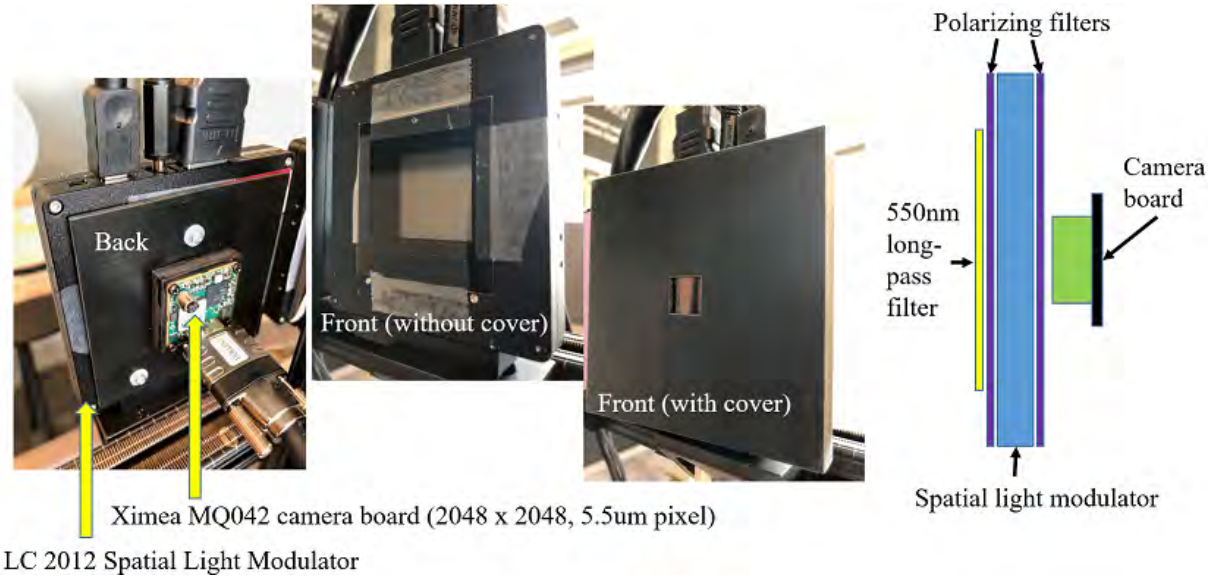


Figure 5.6. Prototype consisting of monochrome camera XIMEA MQ042 and spatial light modulator LC2012.

classified in the top-2 and top-3 choices. "Sitting down" and "hand waving" have not been correctly classified. Examples of our successful and failed videos are shown in Fig. 5.7.

We hypothesize that a possible reason for such failure could be due to the fact that failure videos are much darker than the successful videos. Further, we note that the models used for testing the prototype data has been trained only on NTU and UCF data. The model has not seen a single sample from the real prototype system. We believe this could have caused domain gap between the prototype and simulated models, that led to loss in accuracy. Such performance drop has been observed in other recognition problems as well. For example, loss in accuracy has been observed when a deep model trained on computer graphics data is tested on real world data [**154**].

In order to resolve the domain gap issue, we will investigate two future research directions. First, we will capture a large set of CA training data from our prototype system. Currently

class	top-1	top-2	top-3
BWS (3)	100 / 100	100 / 100	100 / 100
jumping jack (5)	0.0 / 0.0	100 / 0.0	100 / 40.0
standing up (1)	0.0 / 0.0	0.0 / 0.0	0.0 / 100
sitting down (2)	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0
hand waving (8)	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0

Table 5.7. Results on captured CA videos. Accuracies (in percentage) using *with* BE / *without* BE model are reported.

there is no publicly available CA dataset for action recognition. Collecting such a large scale coded aperture dataset is an interesting direction, and will be really valuable for wider research community working on privacy-preserving action recognition problem. Second, we will fine-tune the model that has been pre-trained on large scale simulated CA data, e.g., on NTU-UCF data. Such fine-tuning should help to achieve better robustness and generalization, as shown in RGB based action recognition tasks [**155**]. These are interesting future research directions.

5.7. Discussion

Restoration of coded aperture images. Restoration from CA images is a non-trivial task. Deconvolution can be done if the mask design is known (including PSF or mask code, pixel pitch, distance between the SLM and the sensor) [**136, 130**], although their masks are separable in x and y whereas ours are not. Even when the mask and camera parameters are known, restoring our CA images can be expected to be substantially more computational expensive.

If the mask pattern is unknown, reconstruction approaches can be designed by incorporating several properties of the encoding mask. Correlation-based approaches can be used for recovery as the pseudorandom masks have approximately a delta function as their autocorrelation. The autocorrelation of a CA image is equivalent to the autocorrelation of the scene image: $\boldsymbol{d} \star \boldsymbol{d} \simeq$

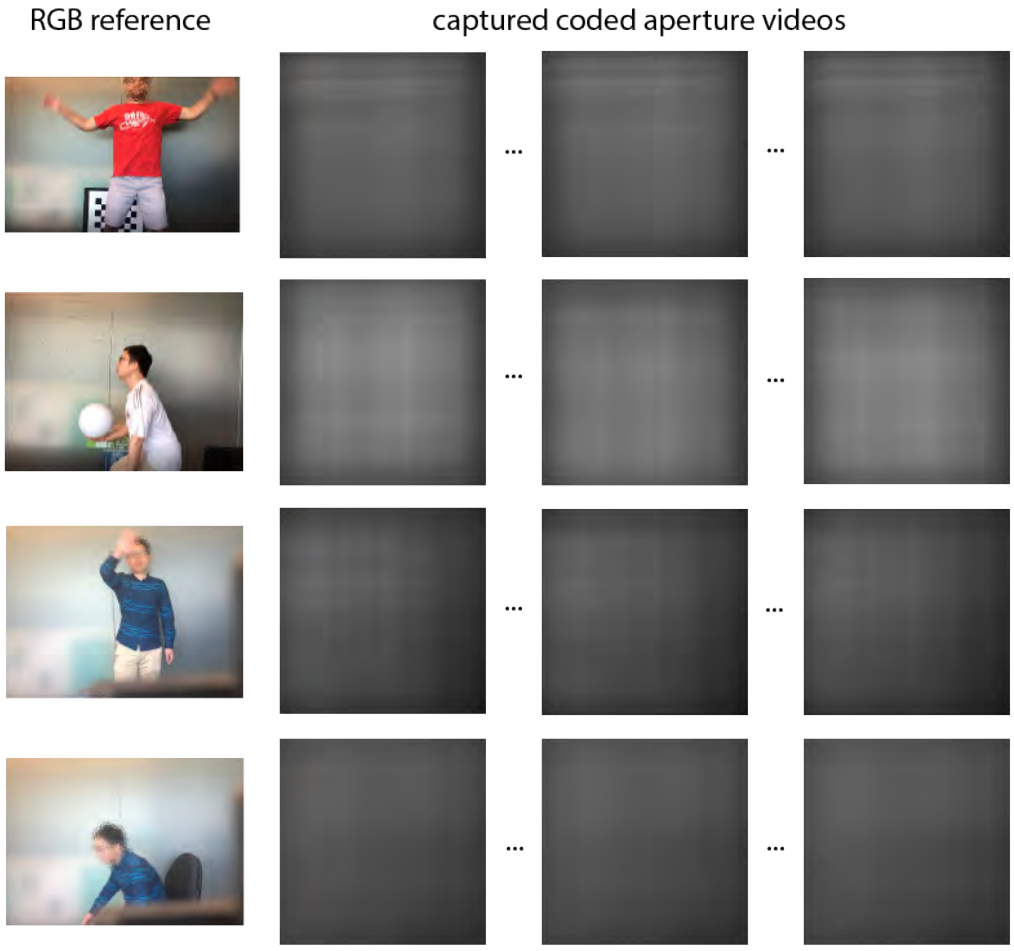


Figure 5.7. Examples of captured videos used for testing. The four rows from top to bottom show one example of the *"jumping jack"*, *"body weight squats"*, *"hand waving"* and *"sitting down"* classes respectively.

$(\boldsymbol{o} * \boldsymbol{a}) \star (\boldsymbol{o} * \boldsymbol{a}) = (\boldsymbol{o} \star \boldsymbol{o}) * (\boldsymbol{a} \star \boldsymbol{a}) \propto \boldsymbol{o} \star \boldsymbol{o}$. The object signal can thus be recovered using a phase retrieval algorithm [**156, 157**]. However, such methods can only restore a coarse image (specifically, near binary quality at high contrast areas). Other constraints such as coprime blur pairs (CBP) [**158**] can be applied for on/post capture video blurring and recovery. Although the polynomial CBP kernels can be estimated, it imposes higher numerical precision for the captured images.

Attacking our system through deep learning is plausible. A deep neural network may be designed to estimate the underlying optical parameters and mask pattern, or to reconstruct the original image; this assumes enough training data can be collected. Since a lensless coded aperture result in a global image transformation, a fully-connected layer may well be required.

Limitations. In our work, we assume that our camera is perfectly stationary, which is typically the case for indoor surveillance. Our FFT-based features are sensitive to extraneous global motion that is not related to body action; a source of such motion is camera shake. As noted earlier, our system is also unable to discern local multiple complex motions such as hand-waving and head scratching.

5.8. Conclusions

There are several interesting takeaways from our experiments. First, training directly on the CA videos results in poor performance. Second, varying the mask at random during training reduces overfitting and improves performance. Third, using multiple strides with TRS (MS-TRS) as input works the best. This is likely attributed to its ability to adapt to different speeds of motion. We also described our prototype, and results for real CA sequences are mixed. However, we believe this is a good first step towards proving the viability of using CA cameras for privacy-preserving action recognition.

CHAPTER 6

Guided Event Filtering: Synergy between Intensity Images and Neuromorphic Events for High Performance Imaging

This chapter documents our recent work on event-based vision, which has been an extension of chapter 2. Although this chapter does not propose a synergistic model, we believe this work has the potential to be further explored in such direction.

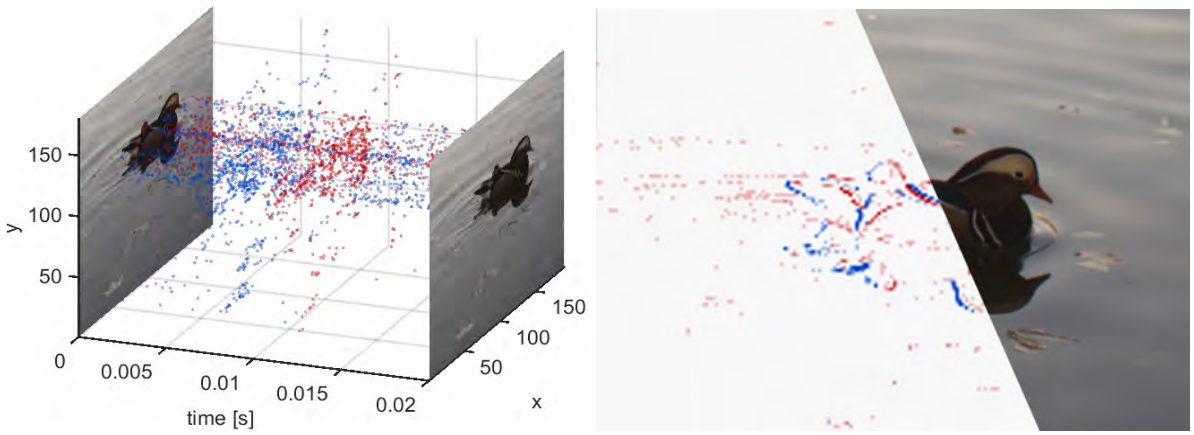
Many visual and robotics tasks in real-world scenarios rely on robust handling of high speed motion and high dynamic range (HDR) with effectively high spatial resolution and low noise. Such stringent requirements, however, cannot be directly satisfied by a single imager or imaging modality, rather by multi-modal sensors with complementary advantages. In this chapter, we address high performance imaging by exploring the synergy between traditional frame-based sensors with high spatial resolution and low sensor noise, and emerging event-based sensors with high speed and high dynamic range. We introduce a novel computational framework, termed Guided Event Filtering (GEF), to process these two streams of input data and output a stream of super-resolved yet noise-reduced events. To generate high quality events, GEF first registers the captured noisy events onto the guidance image plane according to our flow model. it then performs joint image filtering that inherits the mutual structure from both inputs. Lastly, GEF re-distributes the filtered event frame in the space-time volume while preserving the statistical characteristics of the original events. When the guidance images under-perform,

GEF incorporates an event self-guiding mechanism that resorts to neighbor events for guidance. We demonstrate the benefits of GEF by applying the output high quality events to existing event-based algorithms across diverse application categories, including high speed corner detection and tracking, depth estimation, high frame-rate video synthesis, and super resolution/HDR/color image restoration.

6.1. Introduction

The complexity of real-world vision and robotics underlines the importance of high performance imaging and sensing. Ideally, a high performance imaging system shall be able to acquire high speed motions without blur and capture high dynamic range (HDR) images, with high spatial resolution and low sensor noise. However, traditional imaging sensors, *e.g.*, CMOS, fall short in addressing all these aspects.

Event cameras are emerging sensor technology which brings a paradigm shift from traditional cameras. An example comparing the two imaging modalities is shown in Fig. 6.1. While a traditional RGB camera captures a sequence of images in a frame-by-frame manner, an event camera responds only to brightness variations and outputs a stream of asynchronous bipolar events. In Fig. 6.1 the blue/red events represent the brightness (log intensity) increase above or decrease below positive/negative thresholds. Event-based sensing has distinctive advantages including low temporal latency (10μs), HDR (120dB) and low power consumption (10mW) [**159, 160**]. Since its invention, event cameras have shown promising capability in solving classical as well as new computer vision and robotics tasks, including optical flow estimation [**161, 4**], HDR imaging [**23, 162**], high frame-rate video synthesis [**6, 8, 2**], 3D human motion



(a) 3D spatiotemporal view (b) 2D hybrid view

Figure 6.1. While a traditional RGB camera captures images at high spatial resolution, an event camera is capable of recording motion at high speed. (a) Data from our hybrid camera. Between two consecutive frames are high speed events; (b) Left: the event image (accumulated over time) has low resolution and unconventional noise. Right: the RGB image reveals rich spatial details.

capture [**163**], 3D scene reconstruction and odometry [**164, 165, 166, 167**], and autonomous wheeling prediction [**168**].

Despite the numerous advances of event-based vision [**169**], current event sensor prototypes, *e.g.*, DAVIS240, still bear low spatial resolution and unconventional sensor noise. Moreover, the unique event sensing mechanism renders event-based super resolution (SR) and denoising challenging. Although we have witnessed and experienced the success of CMOS sensors and image-based SR and denoising algorithms over the past decades, it is not straightforward to apply image-based algorithms to the novel event data. These sensory and algorithmic imbalances motivate us to design a unifying framework that leverages the complementary advantages of both ends.

Overview of this work:

In this paper, we establish a computational framework, termed guided event filtering (GEF), that bridges event-based sensing with frame-based sensing. As outlined in Fig. 6.2, GEF takes as input two streams of data and outputs a single stream of events with high spatio-temporal resolution and low noise. With improved quality, the output events can interface with existing event-based algorithms for a variety of downstream applications.

GEF consists of three main steps: the motion compensation step, the guided image filtering step and the event re-distribution step. (1) For the motion compensation step, we propose a novel algorithm, henceforth referred to as Joint Contrast Maximization (JCM), that associates events with adjacent image frame(s) via a motion model. We show experimentally results the benefit of employing intensity frames for noise-robust optical flow estimation. (2) For the second step, we filter the motion compensated event frame with the guidance of neighboring intensity image or the event frame itself, outputting a denoised and upsampled event frame. We design a novel switching mechanism that automatically determines the feasible guiding source. (3) For the event re-distribution step, we interpolate the filtered event frame along the flow direction (computed in the first step), preserving the statistical characteristics of the original events.

GEF has a broad range of applications, including high frame-rate video frame synthesis [**8, 6**], motion deblur [**2**], corner/feature detection and tracking [**5**], HDR imaging, depth estimation [**4**], and color event demosaicking [**170**]. We demonstrate and evaluate the effectiveness of GEF for each application. GEF is tested on datasets from two event-based imaging prototypes, *i.e.*, DAVIS240 and CeleX-V [**171**].

Our work has the following limitations: our motion model is based on the linear optical flow assumption. Non-linear motion, occlusions and fast illumination variations are not modeled in GEF and therefore may limit the performance of optical flow estimation. GEF produces optimal

performance when the high-resolution images are provided. In the event self-guided mode, however, the super resolution performance is limited by the lack of high resolution information (results shown for $2\times$). As our hardware prototype consists of two cameras, the system has a combined power consumption that no longer preserves the low power advantage of the event camera alone.

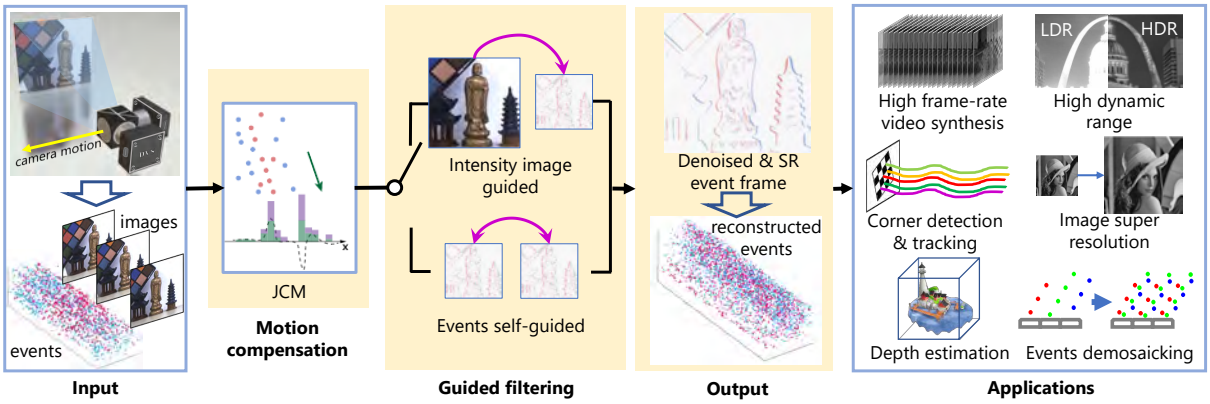


Figure 6.2. The framework of guided event filtering (GEF). Our imaging prototype consists of a high-resolution RGB camera and an event camera DAVIS240. To process the two streams of input signals, we first perform motion compensation to associate local events to image edges using our proposed joint contrast maximization (JCM) algorithm. Guided image filtering is then performed by setting the intensity or the motion compensated event image as guidance. The filtered output is a denoised and super-resolved event frame. The final output of GEF is a volume of densely distributed events that preserves the statistical characteristics of the original events. By generating high quality events, GEF has broad applications.

6.2. Related works

In this section, we briefly review recent advances in event denoising, event-based motion estimation and compensation, RGB and event-based vision, hybrid high-speed cameras, and guided/joint filters.

6.2.1. Event denoising

Unlike conventional RGB cameras, which have comprehensive in-device processing steps [**172**], research in event camera pipelines are still in a preliminary stage. Event denoising is considered a useful processing step in the literature [**173, 174, 175, 176, 177**]. In datasets such as [**177**], the weighted median filter is used to remove noisy events. Recent literature has shown that event denoising serves as an effective pre-processing step for downstream visual tasks. For example, Ev-gait [**178**] exploits local spatial-temporal correlations, and labels isolated events as noise to be canceled. EPM [**179**] presents a method for labeling noise-free event data by calculating the likelihood of generating an event at each pixel. But the accuracy is limited by the sensitivity of the inertial measurement unit. Nonetheless, existing event denoisers focused exclusively on canceling excessive events yet have limited capability retrieving missing events, *e.g.*, under-fired events due to low spatial contrast [**159**]. Our work solves event denoising from a different perspective. GEF performs event denoising by extracting mutual structures from events and images, and reproduce densely-distributed event streams while preserving the statistical properties.

6.2.2. Event-based motion estimation and compensation

Events are inherently associated with motion. One assumption is that local events are triggered by the same edge signal and should comply with the same motion flow [**180**]. The flow parameter can therefore be estimated by maximizing the contrast of the histogram/image of the warped events [**4**]. In [**181**], Almatrafi *et al.* proposed to calculate the optical flow based on the distance surface, each pixel of which is assigned by the distance measure to its nearest event. The

distance surface serves as an effective data association scheme suitable for optical flow computation. Recent work has incorporated the optical flow constraint in training convolutional neural networks (CNN) in an unsupervised manner [**182**]. Motion compensation is a reverse step for event motion and has shown benefits not only for estimating the optical flow but also camera pose estimation [**183**], depth estimation [**4**], motion segmentation [**184**] and feature tracking [**185**]. In [**183**] and [**186**], the 3-DoF model was used to solve the camera angular rotation while a 4-DoF model was used in [**187**] and [**188**]. In [**188**], the motion parameters are modeled as a function of time to achieve improved accuracy. Recently, Mitrokhin *et al.* have leveraged event surface representation for motion segmentation [**189**]. Our work extends to the motion compensation between image and events.We show that by maximizing the contrast of an image jointly formed by the warped events as well as the image edge, the motion parameters associated with the optical flow constraint can be robustly estimated.

6.2.3. RGB and event-based vision

Computer vision scientists have been paying increasing attention to solve visual tasks from the joint source of RGB images and events, taking advantages from both sides. Early works have tackled the problem of video frame synthesis [**1, 6**]. Scheerlinck *et al.* proposed a complementary filters to recover high speed video frames at continuous time [**1**]. To unify different fusion settings including interpolation, extrapolation and motion deblur, Wang *et al.* proposed a differentiable model that enables direct video reconstruction via automatic differentiation, followed by a residual neural network for refinement [**6**]. Recently, Han *et al.* has leveraged the HDR property of events and trained a generative adversarial network to restore a LDR image to its

HDR counterpart [**162**]. In [**190**], Zhu *et al.* leveraged the RGB images to enforce the photo-consistency loss in a self-supervised way while learning event-based optical flow. As events can encode high speed information, Pan *et al.* made used of this advantage to deblur motion blurred images via a double integral model [**2**], and recently proposed a variational approach to jointly estimate the optical flow as well as recovering the sharp image [**191**]. The motion deblur problem has also been approached using learning-based method [**192**]. For high speed feature tracking, Gehrig *et al.* proposed to detect the features on images and propagate the features along event tracks [**185, 193**]. Xu *et al.* applied this idea to recover high speed 3D human motions beyond the framerate of image-based approach [**163**]. Researchers have also revisited image compression. In [**194**], Banerjee *et al.* have proposed a quadtree based compression scheme for both image and events.

6.2.4. Guided/joint image filters

The goal of guided/joint image filters (GIF) is to transfer structural information from a reference image to a target image. The reference and the target can be identical, in which case the filtering process becomes an edge-preserving one [**195, 196, 197, 198**]. In [**195**], He *et al.* proposed a closed-form solution via least squares. Li *et al.* proposed edge-aware weighting scheme for encourage both global and local smoothing [**199**]. Shen *et al.* proposed to penalize for inconsistent edge structures so as to extract mutual structure [**200**]. Recent advances have proposed several CNN models to perform joint filtering [**201, 202, 197**]. Although similar ideas of guided/joint image filtering (GIF) have been explored between RGB and near infrared (NIR) images [**203**], 3D-ToF [**204**], and hyperspectral data [**205**], the major challenge for applying GIF to event cameras is that events do not directly form an image and are spatio-temporally

misaligned by scene motions or illumination variations. In this work, we demonstrate that the events, after a motion compensation step, have structural similarities with respect to the image gradient. The sought-after similarity enables structural transfer from the image to the events.

6.3. Guided event filtering

In this section, we first briefly review the event sensing preliminaries in Sec. 6.3.1, and derive its relation to intensity/frame sensing in Sec. 6.3.2. Our framework guided event filtering (GEF) is then introduced in Sec. 6.3.3 (for the motion compensation step), Sec. 6.3.4 (for the joint filtering step), Sec. 6.3.5 (for event self-guiding mechanism), Sec. 6.3.6 (for the space-time volume redistribution step) and Sec. 6.3.7 (for the implementation details). Figure 2.1 shows the conceptual pipeline of our GEF framework.

6.3.1. Event sensing preliminaries

Consider a latent space-time volume ($\Omega \times T \in \mathbb{R}^2 \times \mathbb{R}$) in which an intensity field is sampled simultaneously by a frame-based camera which outputs intensity images $I(x, y; t)$ and an event camera which outputs a set of events, *i.e.*, $\mathcal{E} = \{e_{t_k}\}_{k=1}^{N_e}$, where N_e denotes the number of events. Each event is a four-attribute tuple $e_{t_k} = (x_k, y_k, t_k, p_k)$, where x_k, y_k denote the spatial coordinates, t_k the timestamp (monotonically increasing), p_k the polarity. $p_k \in \{-1, 1\}$ indicates the sign of the intensity variation in log space. *I.e.*, $p_k = 1$ if $\theta_t > \epsilon_p$ and $p_k = -1$ if $\theta_t < \epsilon_n$, where $\theta_t = \log(I_t + b) - \log(I_{t-\delta t} + b)$. b is an infinitesimal positive number to prevent $\log(0)$. I_t and $I_{t-\delta t}$ denote the intensity values at time t and $t - \delta t$, respectively, and ϵ_p and ϵ_n are contrast thresholds. We will use $\mathcal{L}_t$ to denote the log intensity at time t, *i.e.*, $\mathcal{L}_t \doteq \log(I_t + b)$. For now, we assume that I and $\mathcal{E}$ have the same spatial resolution.

6.3.2. Event-intensity relation

We show that the event and intensity/frame sensing are bridged via temporal gradients. On the intensity side, we employ the optical flow assumption for deriving the temporal gradient of the latent field $\mathcal{L}$. Assume that in a small vicinity, there exists a small flow vector $\delta\mathbf{u} = [\delta x, \delta y, \delta t]^\top$ under which the intensity is assumed to be constant. Mathematically, this assumption can be expressed as:

$$\mathcal{L}(x + \delta x, y + \delta y, t_{\text{ref}} + \delta t) = \mathcal{L}(x, y, t_{\text{ref}}). \tag{6.1}$$

The Taylor series expansion of the left side of Eq. 6.1 gives:

$$\mathcal{L}_{t_{\text{ref}}+\delta t} = \mathcal{L}_{t_{\text{ref}}} + \nabla_{\text{xyt}}\mathcal{L}_{t_{\text{ref}}}\delta\mathbf{u} + o(|\delta x|+|\delta y|+|\delta t|), \tag{6.2}$$

where $\nabla_{\text{xyt}}\mathcal{L}_{t_{\text{ref}}} = [\frac{\partial\mathcal{L}}{\partial x}, \frac{\partial\mathcal{L}}{\partial y}, \frac{\partial\mathcal{L}}{\partial t}]|_{t_{\text{ref}}}$ denotes the gradient operator evaluated at time t_{ref}. If we substitute only the zero- and first-order terms to approximate $\mathcal{L}_{t_{\text{ref}}+\delta t}$ and re-arrange Eq. 6.1, we can obtain the following relation:

$$\frac{\partial\mathcal{L}}{\partial t}\Big|_{t_{\text{ref}}} \simeq -\nabla_{\text{xy}}\mathcal{L}_{t_{\text{ref}}}\mathbf{v} \doteq Q^l, \tag{6.3}$$

where $\nabla_{\text{xy}}\mathcal{L}_{t_{\text{ref}}} = [\frac{\partial\mathcal{L}_{t_{\text{ref}}}}{\partial x}, \frac{\partial\mathcal{L}_{t_{\text{ref}}}}{\partial y}]$ denotes the spatial gradient of $\mathcal{L}_{t_{\text{ref}}}$, and $\mathbf{v} = [\frac{\delta x}{\delta t}, \frac{\delta y}{\delta t}]^\top$ is the velocity vector. For future reference, we define the temporal gradient derived from intensity image as Q^l.

On the event side, the flow velocity $\mathbf{v}$ shall result in position shifts for local events. This is based on the assumption that local events are triggered by the same edge[1], as shown in Fig. 6.3(a). Therefore, the temporal gradient can be approximated by the tangent of a set of

[1]Events generated by illumination variation are not considered here.

warped events in a local window:

$$\frac{\partial \mathcal{L}}{\partial t}\Big|_{t_{\text{ref}}} \approx \frac{\sum_{(t_k - t_{\text{ref}}) \in (0, \delta t)} \epsilon_k \delta(\mathbf{x} - \mathbf{x}'_k)}{\delta t} \doteq Q^e, \tag{6.4}$$

where $\epsilon_k = \epsilon_p$, if $p_k = 1$; and $\epsilon_k = \epsilon_n$, if $p_k = -1$. $\delta(\cdot)$ is the Dirac delta function. $\mathbf{x}'_k$ is the event location by warping (back propagating) measured events to time t_{ref} according to the flow velocity $\mathbf{v}$, *i.e.*, $\mathbf{x}'_k = \mathbf{x}_k - (t_k - t_{\text{ref}})\mathbf{v}$, where $\mathbf{x} = [x, y]^\top$, $\mathbf{x}_k = [x_k, y_k]^\top$ and $\mathbf{x}'_k = [x'_k, y'_k]^\top$. For future reference, we define the temporal gradient derived from events as Q^e.

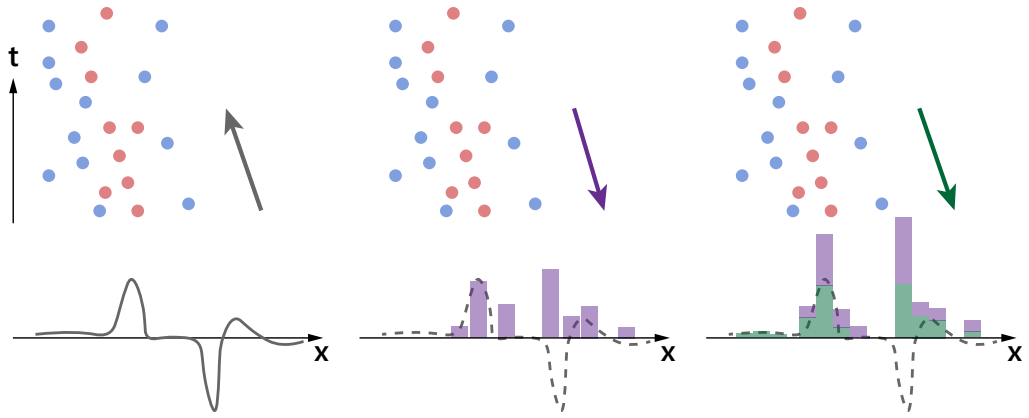
From Eq. 6.4 and Eq. 6.3 we obtain,

$$Q^e \simeq Q^l. \tag{6.5}$$

The above equation establishes the relation between events and image spatial gradients. There are two unknowns, ϵ_k and $\mathbf{v}$ in the relation, where $\epsilon_k \in \{\epsilon_p, \epsilon_n\}$ can be obtained from the event camera configuration. Numerically, ϵ_k can be viewed as a constant scaling value to match Q^e with Q^l. The key unknown is the flow velocity $\mathbf{v}$.

6.3.3. Joint contrast maximization

Previous work [**4**] proposed contrast maximization (CM) to optimize the flow parameter based on the contrast of the image (or histogram) formed only by the warped events, as shown in Fig. 6.3b. However, CM is designed for event data alone. In the presence of an intensity image, we extend the framework of CM and propose joint contrast maximization (JCM) to estimate the flow vector based on intensity image and events. Particularly, we propose to maximize the contrast of an image/histogram jointly formed by the absolute edge of the intensity image and



(a) (b) (c)

Figure 6.3. (a) A latent edge signal (gray curve) triggers a set of (noisy) events due to motion. (b) In contrast maximization (CM) [**4**], the events are warped back at t_{ref} to form a histogram (purple). (c) In our joint contrast maximization (JCM), an image is formed jointly by the events (purple) and the edge of the intensity image (green).

the warped events, as shown in Fig. 6.3c. Mathematically, the image of warped events and intensity edge is expressed as:

$$J(\mathbf{x};\mathbf{v}) = \sum_{k=1}^{N_e} \delta(\mathbf{x} - \mathbf{x}'_k(\mathbf{v})) + \alpha S(\mathbf{x}), \tag{6.6}$$

where $S(\mathbf{x})$ is the edge image which is defined as $S(\mathbf{x}) = \sqrt{|g_x I(\mathbf{x})|^2 + |g_y I(\mathbf{x})|^2}$. We use the Sobel edge (without thresholding) as a discrete approximation. The x-axis kernel can be defined as $g_x = [-1, 0, 1; -2, 0, 2; -1, 0, 1]$, $g_y = g_x^\top$, and $\alpha = \frac{N_e}{\sum_{i,j} S(i,j)}$ is a normalization coefficient to balance the energy of the two data.

The objective for estimating the flow velocity is:

$$\hat{\mathbf{v}} = \underset{\mathbf{v}}{\operatorname{argmax}} \frac{1}{N_p} \sum_{ij} (J_{ij} - \bar{J})^2, \tag{6.7}$$

where N_p indicates the number of pixels in an image patch J, while $\bar{J}$ denotes the mean value of J. Note that when no intensity image is available or it has low quality (*e.g.*, blurry), the Sobel term can be set to zero and the formulation degenerates to event-only contrast maximization [**4**]. With non-zero S, the maximal contrast corresponds to the flow velocity that transports events to the image edge. Non-optimal velocity will lead to a deterioration of the contrast.

Here, we perform a numerical comparison between CM and JCM, shown in Fig. 6.4. We follow the analysis in [**159**] and [**24**] for event simulation from images. More specifically, a thresholding operation ($\epsilon_p = 0.2$, $\epsilon_n = -0.2$) is applied to the difference image between the flow-shifted image and the original/last image. The event noise follows a Gaussian distribution around the per-pixel threshold values [**159**]. We consider a standard deviation range of $\sigma_e \in (0, 0.1)$, and compare the accuracy for flow estimation with respect to different flow directions with fixed flow radius of 5 pixels. We use the Euclidean distance to quantify the flow estimation error. The error is averaged over 18 images of size 30×30. Details of this experiment as well as visual examples can be found in the supplementary material. As shown in Fig. 6.4, both JCM and CM errors increase as noise level increases. However, JCM maintains low error across the range of noise levels, revealing a more noise-robust property than CM.

6.3.4. Joint filtering

Q^e and Q^l are two sensory observations of the same quantity, *i.e.*, temporal gradient. We employ joint filtering to construct an output temporal gradient image that inherits the joint characteristics of Q^e and Q^l. Our previous investigation [**206**] compared three filters and concluded that the mutual structure filter [**200**] performs best for both denoising and upsampling. Here, we

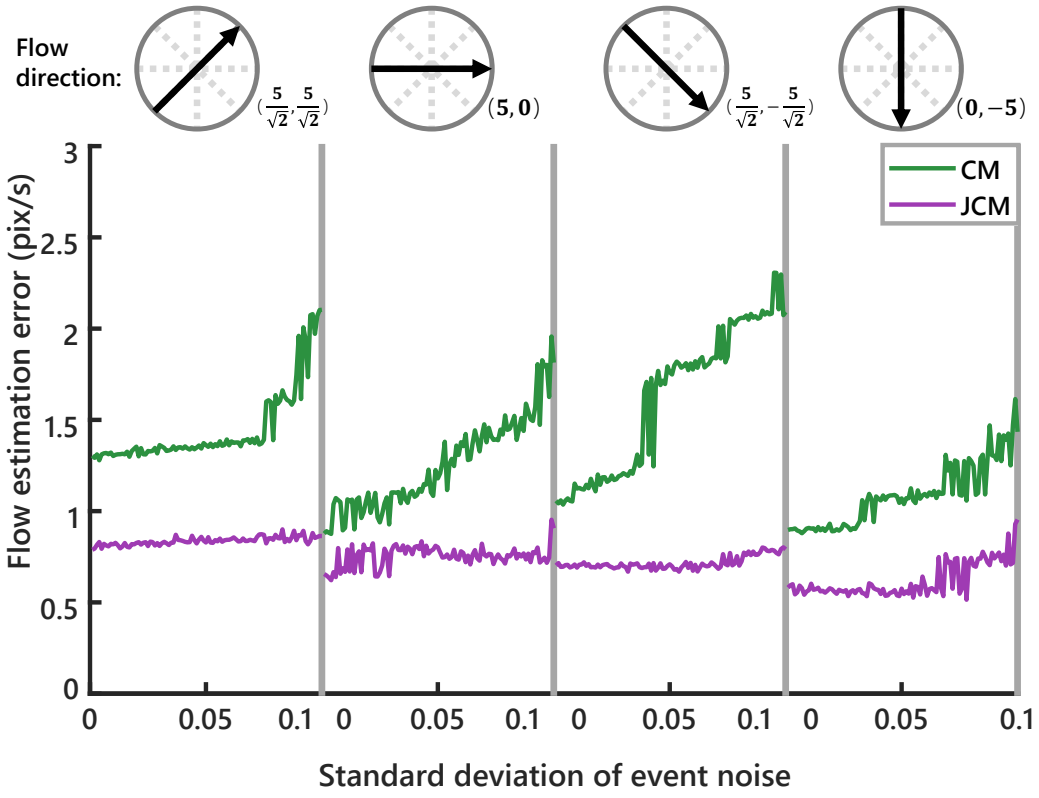


Figure 6.4. Comparison between CM and JCM [**4**] for flow estimation with respect to event noise.

use the same filter as our backbone optimizer and extend our investigation into the determination of cross and self filters. To reiterate, given two image patches centered at pixel location $\mathbf{x}$ with size of $w \times w$ (represented as $Q^e_{\mathbf{x}}$ and $Q^l_{\mathbf{x}}$), our objective is to minimize the cross linear representations:

$$\begin{aligned} &\underset{g_a,g_b,g'_a,g'_b}{\operatorname{argmin}} ||g_a Q^l_{\mathbf{x}} + g_b - Q^e_{\mathbf{x}}||_2^2 + ||g'_a Q^e_{\mathbf{x}} + g'_b - Q^l_{\mathbf{x}}||_2^2 \\ &\lambda_1(||g_a Q^l_{\mathbf{x}} + g_b - Q^l_{\mathbf{x}}||_2^2 + ||g'_a Q^e_{\mathbf{x}} + g'_b - Q^e_{\mathbf{x}}||_2^2) \\ &\lambda_2(g_a^2 + g_a'^2), \end{aligned} \tag{6.8}$$

where g_a, g_b, g'_a, g'_b are the parameters for the linear cross regressions in the first line. The second line is to enforce minimal deviation from the original patches, while the third line is to prevent the coefficients from being too large. The output Q^o is constructed as $Q^o_{\mathbf{x}} = g_a Q^l_{\mathbf{x}} + g_b$, when the image patch is used as the guidance.

If the image patch is not available or if it experiences strong degradation, the events should be switched to self-guiding mode. This is done by partitioning the event packet into two sub-packets according to the timestamps. We use $\mathcal{E}_a$ to denote events in $(0, T/2]$ and $\mathcal{E}_b$ to denote events in $(T/2, T]$. To perform motion compensation, we set timestamp $T/2$ as the reference timestamp t_{ref} in Eq. 6.4. $\mathcal{E}_a$ and $\mathcal{E}_a$ are warped to t_{ref} to form Q^e_a and Q^e_b via a common motion vector. Equation 6.8 is performed by setting $Q^l = Q^e_a$ and $Q^e = Q^e_b$. The output Q^o is constructed as $Q^o_{\mathbf{x}} = (g_a Q^e_{a\mathbf{x}} + g_b + g'_a Q^e_{b\mathbf{x}} + g'_b)/2$.

For the guided upsampling, Eq. 6.8 is recursively performed for every $2\times$ upsampling. In the image-guiding case, our system has high resolution Q^l at $8\times$. In the self-guiding case, both Q^e_a and Q^e_b are bicubically upsampled and then filtered.

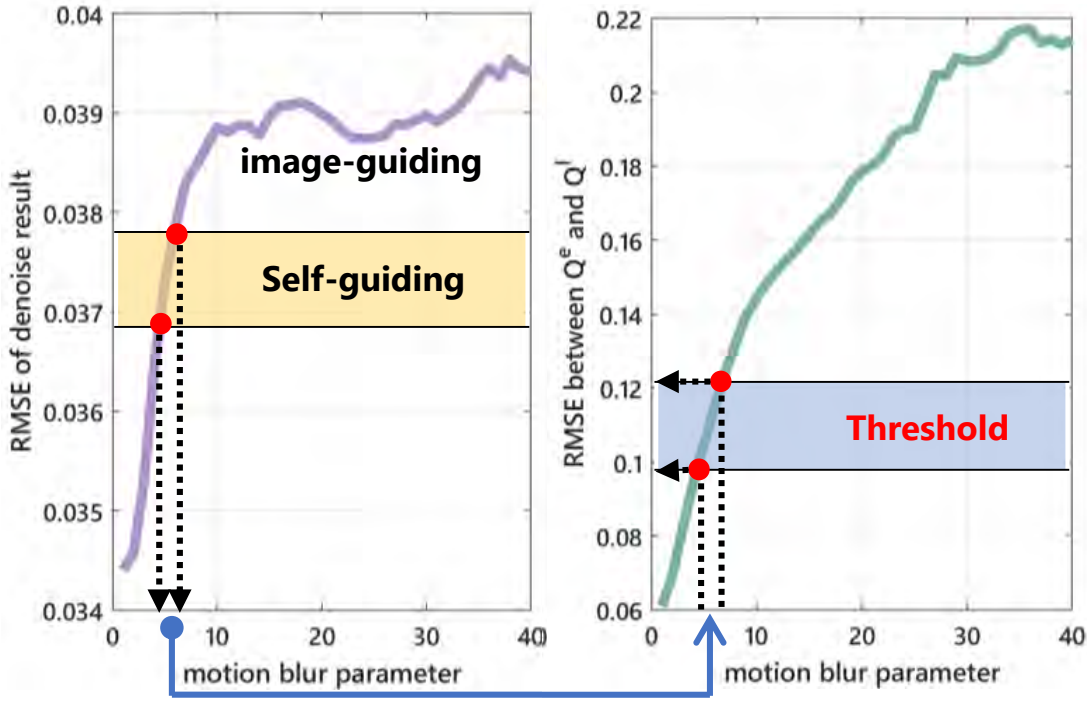
6.3.5. When does image guidance fail?

One critical step is to determine when to switch from the image-guiding mode to the event self-guiding mode. We study this problem by simulating the image degradation and comparing the performance for the two guided filtering modes. We consider the motion blur as our image degradation model, as shown in Fig. 6.5a. For motion blur, the blur kernel is a linear motion trajectory quantified by the length of the trajectory. Large kernel parameters result in significant blur artifacts. We report the average results for 20 simulation cases where events are simulated from the original sharp images. The RMSE (between the ground truth temporal gradient frame

and the filtered image) is used as the performance metric. The filtering results are plotted against the kernel parameters, shown as the purple curve in Fig. 6.5b. It has clearly shown that the error (RMSE score) increases as the blur grows. The range for the scores when the image-guiding filter under-perform the self-guiding filter are highlighted as the yellow shaded region. Through coordinate correspondence, we can find that when the blur radius of motion blur exceeds 6, the image-guiding mode results in larger error than the self-guiding mode. Therefore, it is sensible to switch to the event self-guiding mode.

However, the threshold values obtained from the simulated data cannot be directly applied to the real data because the blur parameters of the guidance images are unknown and non-trivial to estimate. To tackle this issue, we propose a simple yet effective approach via comparing the similarity between the guidance image Q^l and the warped event frame Q^e. We compute the same RMSE scores to rate the similarity, plotted as the green curve in Fig. 6.5c. Interestingly, the similarity score is also positively correlated with the blur parameter. In this case, it is straightforward to interpret the blur parameters by the RMSE scores, which is 0.11 for motion blur. This experiment has provided us an empirical RMSE value to determine when we should switch to the event self-guiding mode. That is, when the RMSE between normalized Q^e and Q^l is less than the threshold, GEF uses the images as the guidance, otherwise the event self-guiding mode is used. Considering the difference between simulated data and real data, as well as the particularity of individual data, we recommend that the threshold interval is [0.10, 0.12].

(a) Simulating motion blur images using different trajectory lengths.



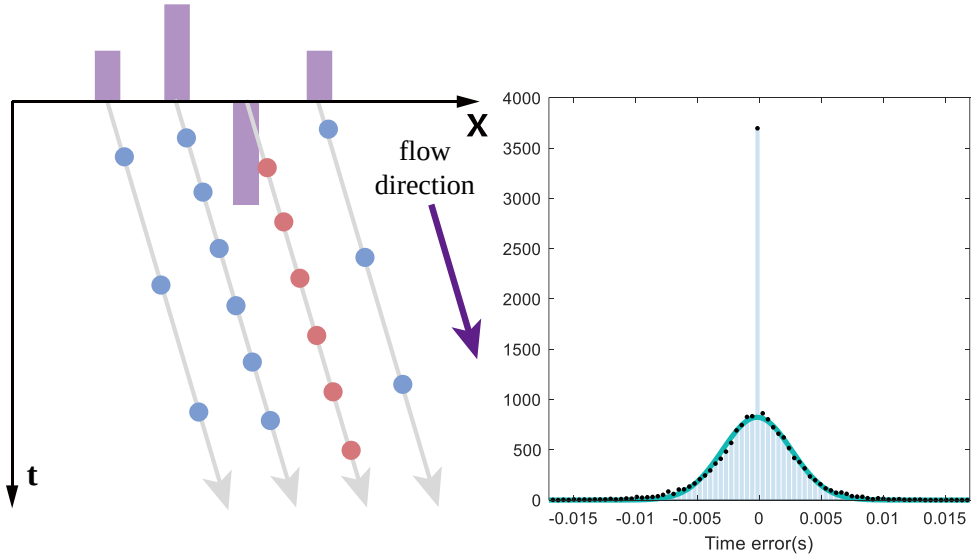
(b) (c)

Figure 6.5. Comparison between image-guided filtering and event self-guided filtering with respect to image blur degradation. (a) We consider and simulate the motion blur (numbers indicate the lengths of motion). (b) We use 20 clear (no blur) images to generate event simulation data, and then blur the guidance images with different blur kernels to perform GEF event denoising, compare the changes in denoising performance, and then determine the self-guiding switching threshold on blur kernel parameter. (c) We convert the threshold from the blur kernel parameter to the similarity between Q^e and Q^l. The shaded area indicates the recommended threshold range.

6.3.6. Space-time volume re-distribution

The output of the guided image filtering step is an optimized event frame. Although many existing event-based algorithms process events by first binning them into images, other event-based algorithms work directly with individual events or event packets. The difference is that the binned event images reduce the temporal resolution compared to the original microsecond-level timestamps. To address this issue, we propose a solution to re-distribute the events into space-time volume from the optimized 2D image, which is the inverse of Eq. 6.4. We first quantize the values in Q^o to signed integers. The integers therefore represent the number of events at the corresponding spatial location. We leverage the previously computed optical flow to propagate the event values, as illustrated in Fig. 6.6a. In the ideal case of linear motion and no sensor noise, the events are evenly distributed along the flow direction. However, this condition cannot be achieved in real scenarios. To preserve the statistical characteristics of the original events, we analyze timestamps of real event data and follow the time error distribution of real data during the re-distribution process.

We first estimate the optical flow using our JCM for 10 real-scenario image-events packets collected using our prototype (detailed in Sec. 2.4). We then group the event timestamps corresponding to the same warped location of Q^e into a sequence $\{rt_i\}, i \in (1, ..., n)$, and generate an ideal sequence $\{pt_i\}$ with timestamps evenly distributed. Then the difference between each rt_i and pt_i is recorded as time error. We calculated the time error of 17,000 events in total, and recorded their distribution histogram in Fig. 6.6b. As can be seen, most of the time error (t_{error}) is close to zero while the rest of the time error approximately follows a Gaussian distribution (Gaussian fitted as the green curve in Fig. 6.6b) on the data with non-zero time errors. From the plot, a probability model can be expressed as:



(a) (b)

Figure 6.6. (a) The purple histograms denote the denoised or upsampled Q^e obtained with GEF, we warped them back into the space-time volume along the computed flow direction to restore the ternary representation. (b) Histogram of the distribution of time errors in real data (light blue bars), and a Gaussian function (green curve) fitted to data with time errors.

$$p(\tau) = \begin{cases} 0.22 & \tau = 0 \\ \frac{0.78}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\tau^2}{2\sigma^2}\right) & \tau \neq 0, \end{cases} \tag{6.9}$$

where τ denotes the time difference compared to the equally interpolated events, and $\sigma = 0.0028$ (result of data fitting). The events can be re-distributed by first interpolating events along the flow vectors with equal intervals, and then applying a Gaussian perturbation following Eq. 6.9. We experimented with both the equal-interval distribution and the Gaussian distribution but found no substantial difference in performance (*e.g.*, image reconstruction). We therefore report the results for the equal distribution approach. The space-time volume re-distribution examples are shown in Sec. 6.4.

Algorithm 2 Guided Event Filtering (GEF)

Input: Intensity image I, events $\mathcal{E}$.
Output: Filtered temporal gradient Q^o, restored events $\mathcal{E}'$.
1: Estimate the flow field $\mathbf{v}$ using JCM in Eq. 6.7;
2: Compute Q^l in Eq. 6.3 and Q^e in Eq. 6.4;
3: Calculate RMSE between Q^l and Q^e, switch between image-guiding and self-guiding mechanism as in Sec. 6.3.5.
4: Perform guided filtering according to Eq. 6.8.
5: Re-distribute Q^o into space-time volume according to Sec. 6.3.6.

6.3.7. Implementation details

The steps of GEF are summarized in Alg. 2.

In the JCM step, we use a local window with radius r_w to estimate pixel-wise flow. Areas with events fewer than 1 are skipped. r_w may vary due to the structure of the scene. A large r_w can be used when the scene has sparse and isolated objects, in exchange for more time to compute the flow field. The intensity image support is slightly larger (about several pixels on four sides) than the event window to prevent fallout of events due to large velocity.

Both the computation of the flow velocity and Q^l use the spatial gradient. Therefore, the spatial gradient image can be computed once. Q^l is normalized to match the range of Q^e before the filtering step. This normalization step also functions as an estimation for the event threshold (ϵ_k). The output image Q^o is rounded to have integer values as the original events are integers. The integers can be interpreted as the event counts. In the switching step, we calculate the similarity error between normalized Q^e and Q^l. The threshold is set to 0.09.

In the joint filtering step, we set the window width to be equal to 1. The parameters are set as λ_1 (~ 3) and λ_2 ($\sim 1 \times 10^{-2}$). The filtering is run for 20 iterations to achieve convergence. In the filtering process, with an i7-8700K CPU, the average runtime for $2\times$ upsampling a 180×190 frame is about 0.2s.

6.4. Experiments

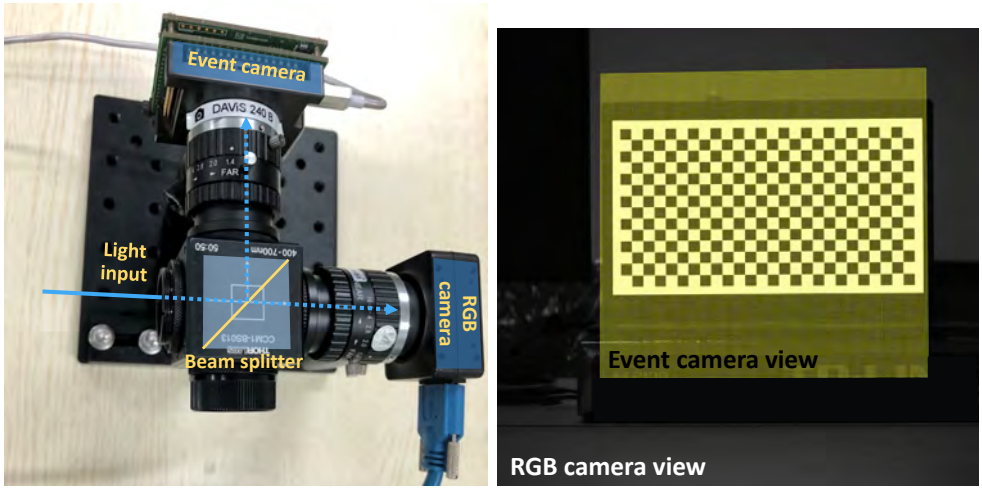
Our conference version [**206**] has focused on a comprehensive evaluation of the guided denoising and upsampling aspects of GEF. The numerical experimental results are included in the supplementary material. Here, we focus on applying our GEF to two hardware setup, our hybrid RGB-DAVIS camera and the CeleX event camera [**171**].

6.4.1. RGB-DAVIS camera system

To test GEF for real-world scenarios, we build a hybrid camera consisting of a high-resolution machine vision camera and a low-resolution event camera, *i.e.*, DAVIS. We refer to our camera prototype as RGB-DAVIS camera.

6.4.1.1. Setup and calibration. As shown in Fig. 6.7a, we collocate an event camera (DAVIS240b, resolution of 180×190 pixels, with F/1.4 lens) and a machine vision camera (Point Grey Chameleon3, resolution of 2448×2048 pixels, 50 FPS, with F/1.4 lens). A beam splitter (Thorlabs CCM1-BS013) is mounted in front of the two cameras with 50% splitting. We use a 13.9" 60Hz monitor for offline geometric calibration for two signals. For geometric calibration, we mainly consider homographic mapping between two camera views. In order to extract keypoints from event data, we display a blinking checkerboard pattern on the monitor and integrate the captured events over a time window to form a checkerboard image, as shown in Fig. 6.7b. For temporal synchronization, we write a synchronization script to trigger the two cameras simultaneously.

6.4.1.2. Dataset collection. We use RGB-DAVIS to collect various sequences of event-RGB video clips. Examples are shown in Fig. 6.8. Both indoor and outdoor scenarios are captured. The scenes widely range from simple shapes to complex structures. All the clips involve camera



(a) Experimental setup (b) Calibrated views

Figure 6.7. Our RGB-DAVIS imaging system.

motion and/or scene motion. In total, there are 20 video clips, with an average length of 8s for each clip. A full description of each clip is shown in Table 6.1.

6.4.1.3. Results. After calibration, we perform joint filtering with three upsampling scales, *i.e.*, $2\times$, $4\times$, $8\times$. The flow is estimated at $1\times$. The captured data as well as the filtered results are shown in Fig. 6.9, with the filtered output shown in Fig. 6.9 (c-f). As can be seen, the events are gradually and effectively upsampled and denoised. Please see additional results for scene motion as well as filtering results using other filters in the supplementary material. In Fig. 6.10, we show the re-distributed events. Figure 6.10 (b-d) show $2\times$, $4\times$, $8\times$ re-distributed results. Compared with the corresponding original LR data shown in Fig. 6.10a, events restored by GEF have both been improved the spatial resolution and significantly reduced noise. Besides, Fig. 6.11 shows three examples of self-guiding filtering results. Compared to Q_a^e and Q_b^e, noise in $1\times$ output has been significantly eliminated. For $2\times$ results, although the texture reconstruction is

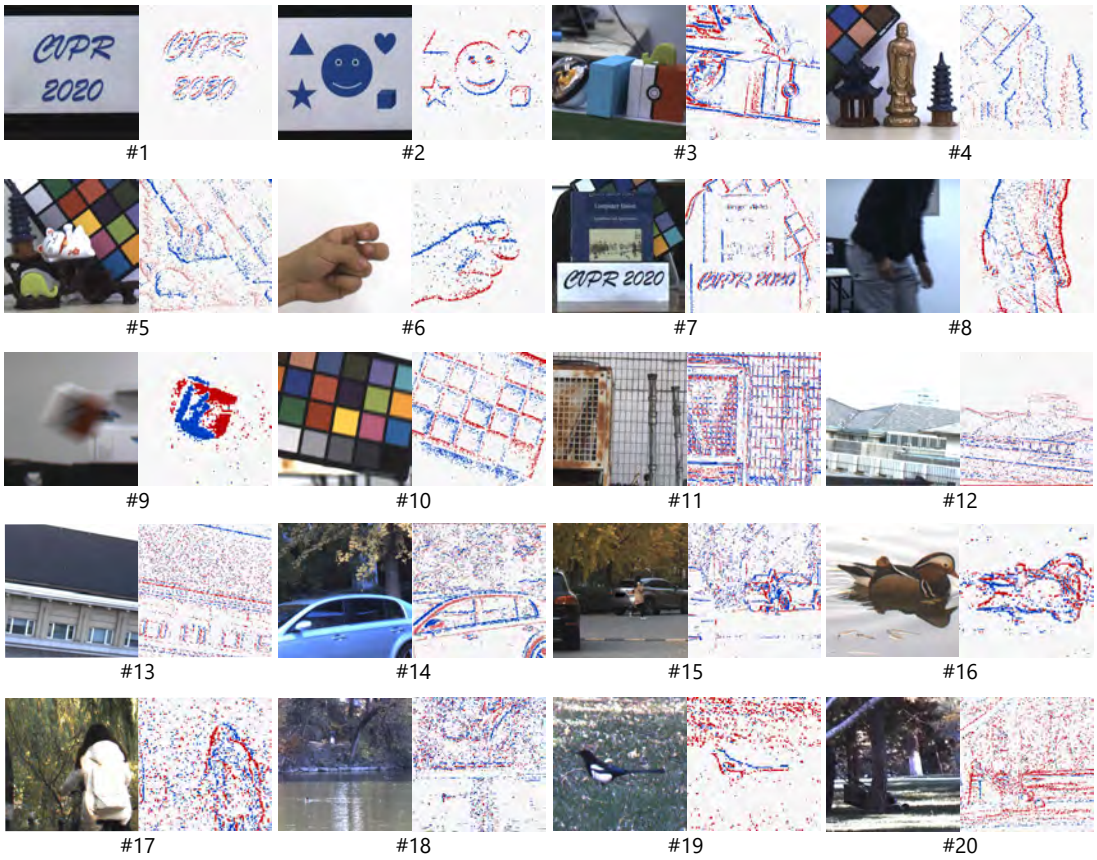
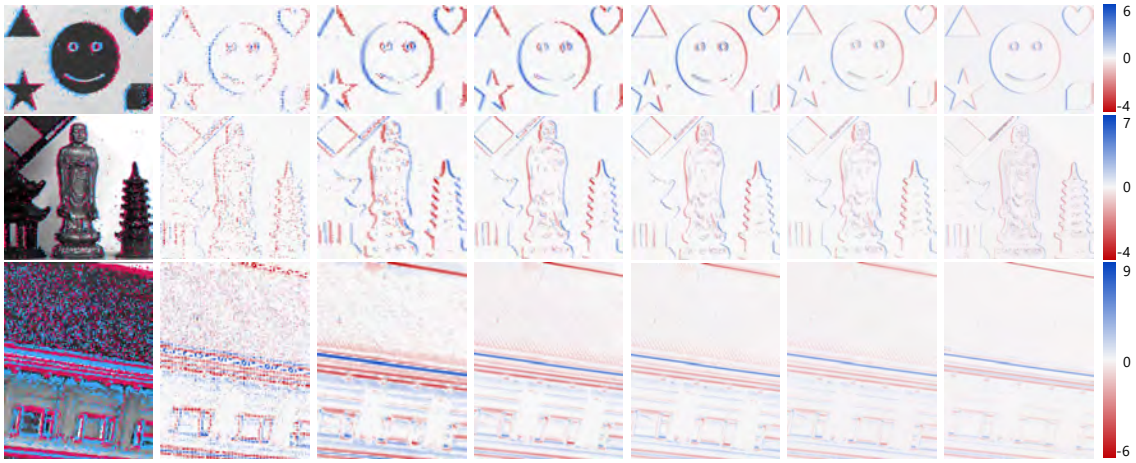


Figure 6.8. Our proposed RGB-DAVIS dataset. Shown images are screenshots of RGB videos (left) and event videos (right).

not as good as that under the RGB guidance, it still achieves the initial event image sampling task.

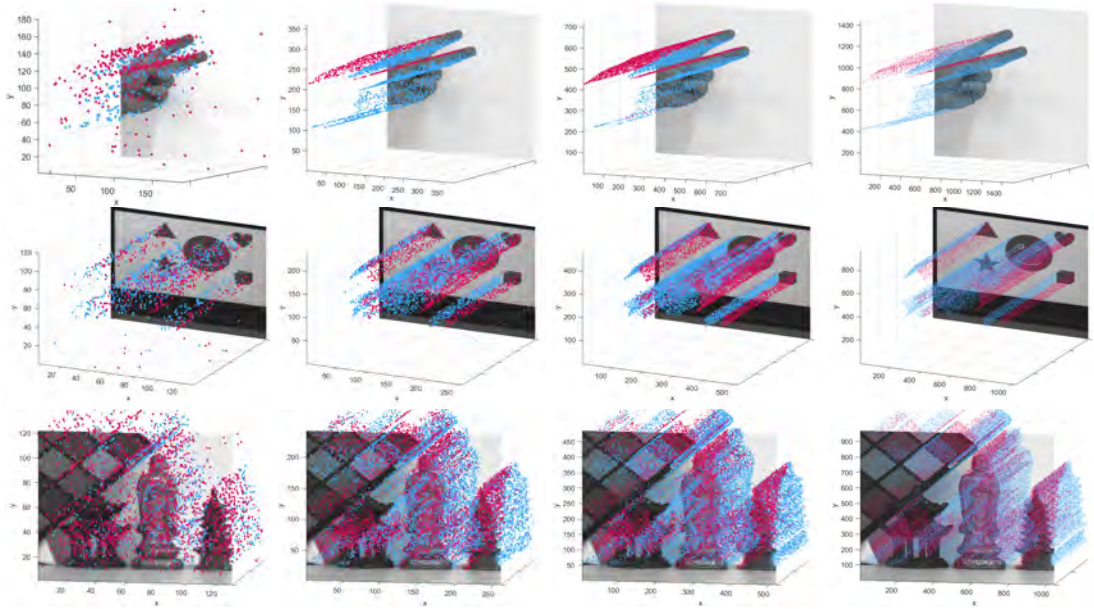
6.4.2. GEF for CeleX-V

The CeleX series event sensors have higher spatial resolution than the DAVIS. The newest model, CeleX-V [**171**], has spatial resolution of 1280×800 and is able to record pixel intensity with 120dB dynamic range. Benefiting from intensity recording and a special periodic pixel



(a) Image+events (b) Q^e (c) $1\times$ (d) $2\times$ (e) $4\times$ (f) $8\times$ (g) Q^l

Figure 6.9. Guided upsampling results on our RGB-DAVIS data.



(a) Original LR event data (b) $2\times$ (c) $4\times$ (d) $8\times$

Figure 6.10. Space-time volume redistribution results on our RGB-DAVIS data. We choose a 3D view for each example that helps to make a significant visual comparison.

Table 6.1. Details of our RGB-DAVIS dataset

clip	# of images	# of events	indoor/ outdoor	camera motion	scene motion	description
#1	250	2.6M	indoor		✓	text displayed and animated on the monit
#2	200	8.9M	indoor		✓	simple shapes displayed and animated on the r
#3	200	7.3M	indoor	✓		static objects with camera motion
#4	200	3.5M	indoor	✓		static objects with camera motion
#5	200	1.7M	indoor	✓		static objects with camera motion
#6	200	10.5M	indoor		✓	hand gestures
#7	150	2.4M	indoor	✓		textbook with background
#8	400	23.8M	indoor		✓	human body motion
#9	400	20.8M	indoor		✓	abruptly throwing an object
#10	400	21.8M	indoor		✓	color chart with hand-held motion
#11	200	3.8M	outdoor	✓		wall with grid structure
#12	190	3.6M	outdoor	✓		building with window
#13	200	4.3M	outdoor	✓		building
#14	400	9.3M	outdoor	✓	✓	car moving
#15	400	8.4M	outdoor	✓	✓	street with cars
#16	400	27.7M	outdoor	✓	✓	bird in a lake
#17	400	23.1M	outdoor	✓	✓	pedestrians walking on street
#18	400	22.6M	outdoor	✓		static objects with structured backgroun
#19	400	20.7M	outdoor	✓	✓	bird on grass
#20	150	23.7M	outdoor	✓	✓	a weeding worker in a park

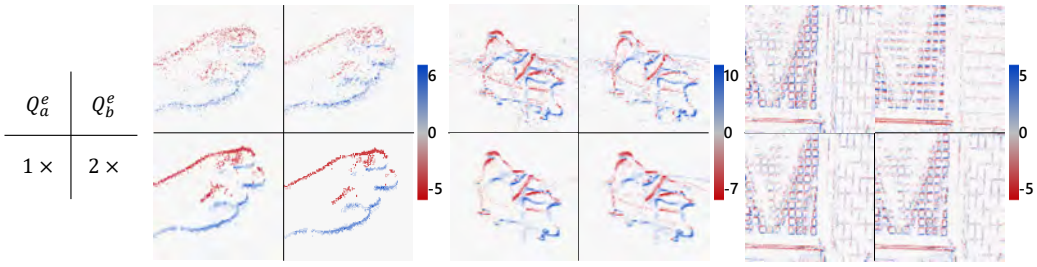
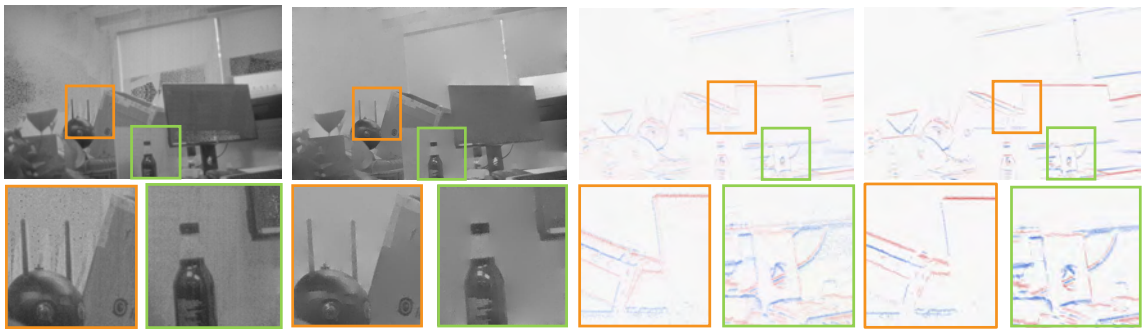


Figure 6.11. Event self-guiding filtering results on our RGB-DAVIS data.



(a) Accumulated frame w/o GEF (b) Accumulated frame w/ GEF (c) Filter input, Q^{ce} (d) Filter output

Figure 6.12. Our modified GEF for CeleX-V event camera

activation mechanism, CeleX-V can output grayscale images in real-time through events. However, the data output from CeleX-V still has high level of noise. The event noise also affects the real-time grayscale images reconstructed from pixel intensity, which is termed as the afterimage phenomenon. We address the imaging issues of CeleX-V by first anaylize the image formation model of the system.

6.4.2.1. CeleX-event sensing model. Slightly different from the DAVIS event camera (Sec. 6.3.1), the output of the CeleX camera is a set of events named c-events, *i.e.*, $\mathcal{E} = \{ce_{t_k}\}_{k=1}^{N_{ce}}$. Each c-event can be expressed as $ce_{t_k} = (x_k, y_k, t_k, p_k, a_k)$, where p_k denotes the polarity, $p_k = \{1, -1\}$ indicates the sign of the intensity variation in log space, and $p_k = 0$ means the pixel has no intensity variation but is activated by the periodic activation mechanism, and a_k records the pixel's

intensity I at (x_k, y_k, t_k). This sensing activation process can be approximated as:

$$p_k = \begin{cases} -1, & \theta_t < \epsilon_n \\ 1, & \theta_t > \epsilon_p \\ 0, & \theta_t \in [\epsilon_n, \epsilon_p] \quad \& \quad h(x_k, y_k, t) > \epsilon_t \\ \text{N.E.}, & \text{else}, \end{cases} \tag{6.10}$$

where ϵ_n and ϵ_p are contrast thresholds, ϵ_t denotes a time threshold. $\theta_t = \log(I_t + b) - \log(I_{t-\delta t} + b)$, and N.E. stands for no event being fired. h is a function that records the no-event duration of each pixel:

$$h(x_k, y_k, t + \delta t) = \begin{cases} \delta t, & p_k \in \{-1, 0, 1\} \\ h(x_k, y_k, t) + \delta t, & \text{else}. \end{cases} \tag{6.11}$$

The initial duration $h(x, y, 0)$ is set to zero. Then the real-time grayscale image G can be express by:

$$\begin{aligned} G_t(x, y) =& G_{t-\delta t}(x, y) \cdot (1 - W_t(x, y)) + \\ & \sum_{k=1}^{N_{ce_t}} a_k \cdot \delta(x - x_k, y - y_k), \end{aligned} \tag{6.12}$$

where N_{ce_t} donates the number of the events that are triggered at time t and $\delta(\cdot)$ is the Dirac delta function. $W_t(x, y)$ is a position filter, described by $W_t(x, y) = \sum_{k=1}^{N_{ce_t}} \delta(x - x_k, y - y_k)$. The initial grayscale image G_0 is a zero matrix, and the full frame grayscale image can be obtained after each pixel has been activated, which is named as the accumulated frame. We show an example of G_t in Fig. 6.12a. It has been captured in a scene where the event camera is rapidly shaking.

The afterimage phenomenon: The accumulated frame is a nice feature associated with the CeleX sensor. Compared to the activate pixel sensor (APS) image, the accumulated frame has much higher frame rate which is useful for the temporal synchronization with the adjacent events. However, since G_t is reconstructed by updating the pixel intensity in time according to the newly triggered c-events, the previously accumulated image will leave a shadow due to the lack of intensity update. We call this the afterimage phenomenon, as can be seen for the residual shadow in Fig. 6.12a.

6.4.2.2. Adapting GEF for CeleX-V. We adapt our GEF framework to apply to CeleX-V. In terms of the optical flow estimation, the contrast image is redefined as:

$$J(\mathbf{x};\mathbf{v}) = \sum_{k=1}^{N_e} a_k \cdot \delta(\mathbf{x} - \mathbf{x}'_k(\mathbf{v})) + \alpha S(\mathbf{x}). \tag{6.13}$$

Although the accumulated frame remains traces of past motion, the warped event frame preserves the structure of the real-time motion region and is still able to preserve the robustness for flow estimation. In the joint filtering step, we reversely set the event frame Q^{ce} as the guidance and the Q^l computed from the accumulated image. The output Q^o is then used to restore the gradient field $\nabla_{\text{xy}} G'$ with the estimated flow, and further reconstruct a real-time grayscale image G' without the afterimage effect. The results for removing the afterimage effect is shown in Fig. 6.12b. The temporal gradient image before and after our modified GEF is shown in Fig. 6.12d. Our modified GEF reveals structural enhancement, demonstrating its effectiveness.

6.5. Applications

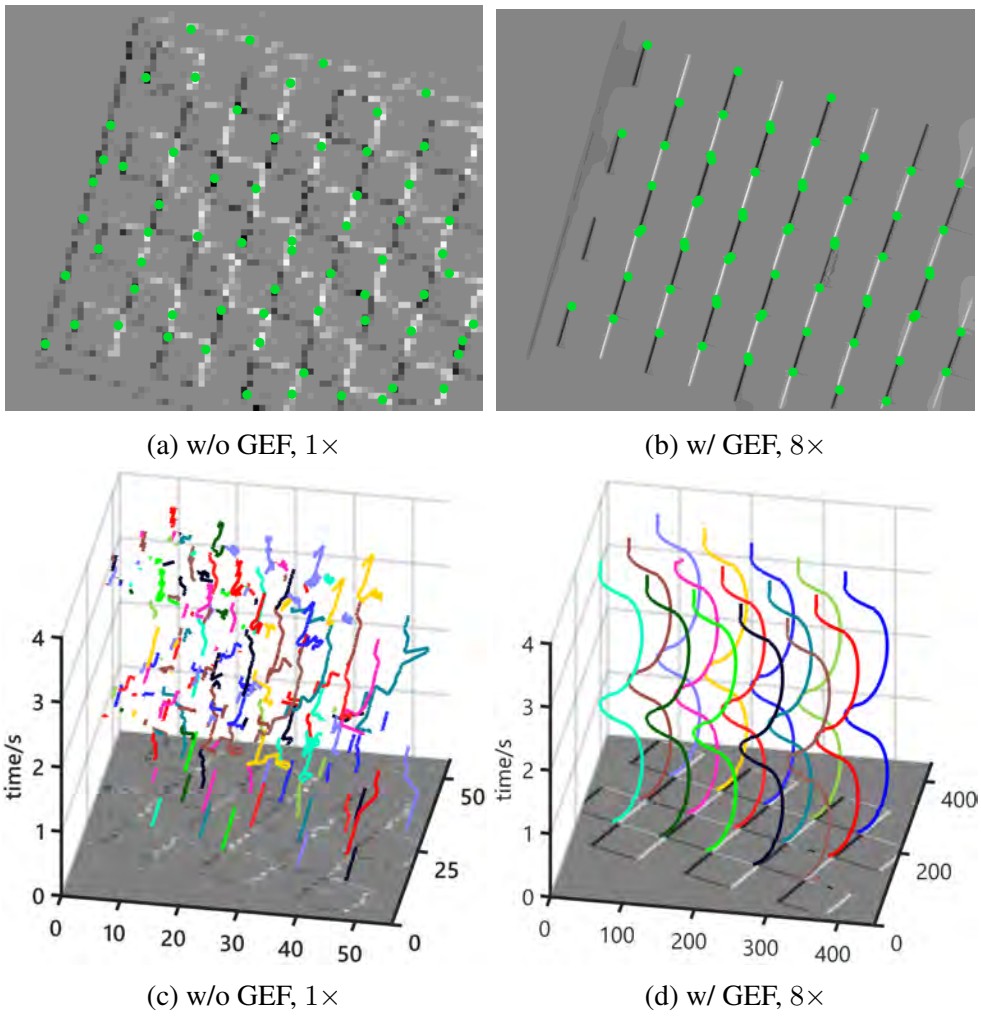
GEF has a wide variety of applications for event-based tasks. Here, we enumerate several example applications.

6.5.1. Corner detection and tracking

GEF can be applied on event-based feature/corner detection and tracking. To demonstrate the benefit of guided upsampling, we use RGB-DAVIS camera to capture a periodic circularly moving checkerboard pattern. We employ the event-based Harris corner detector (evHarris) [**5**] as the backbone corner detector. A slight difference between our implementation and the original evHarris is that we use the warped event image (motion compensated), instead of directly accumulating events in local windows. As shown in Figs. 6.13a and 6.13b, with GEF ($8\times$ guided upsampling), the checkerboard corners are detected more accurately than w/o GEF. We also compare the corner tracks computed both *w/o* and *w/* GEF process. The results are shown in Figs. 6.13c and 6.13d. As can be seen, the corner points that are upsampled by the GEF can be tracked more accurately than the original frames.

6.5.2. Depth estimation

We compare the JCM module with CM [**4**] for the task of depth estimation. Gallego *et al.* [**4**] proposed an event-based depth estimation method: when objects with different depths of field move in the XoY plane, the length of pixel movement on the image plane is linearly related to the depth of field. Given the camera matrix and trajectory of the DAVIS, it converts the problem of determining the optimal depth value into the problem of finding events contrast maximization. As an improvement, our proposed JCM leverages the image edge signals to

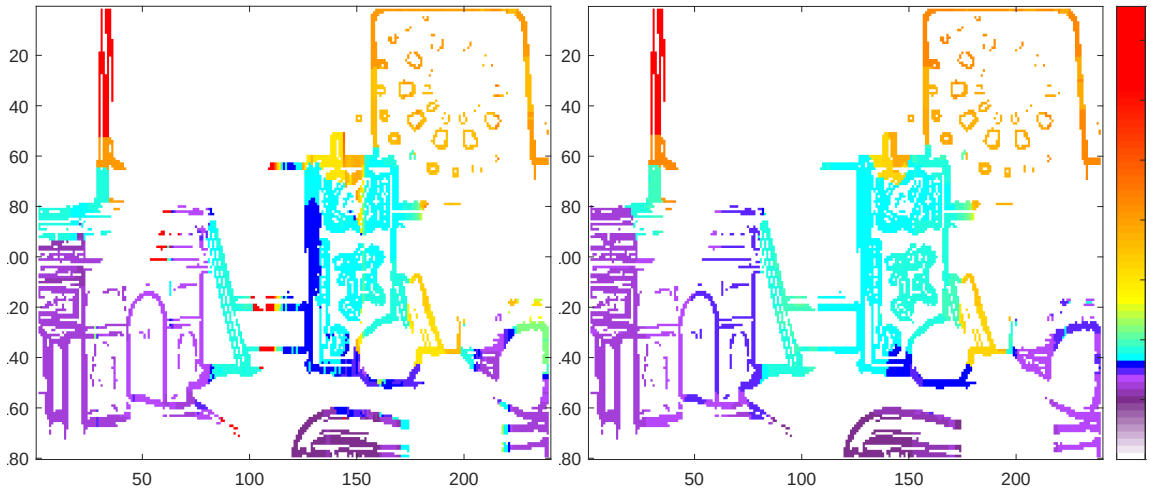


(a) w/o GEF, 1× (b) w/ GEF, 8×

(c) w/o GEF, 1× (d) w/ GEF, 8×

Figure 6.13. Corner detection using evHarris [**5**].

reduce the event noise and can improve the accuracy of depth estimation. In Fig. 6.14, we use an example data of [**167**] to campare CM and JCM, Figs. 6.14a and 6.14b are the depth maps estimated by CM and JCM respectively. Around the location where complex motion and occlusion are involved, JCM is able to provide stable and consistent depth.



(a) CM (b) JCM

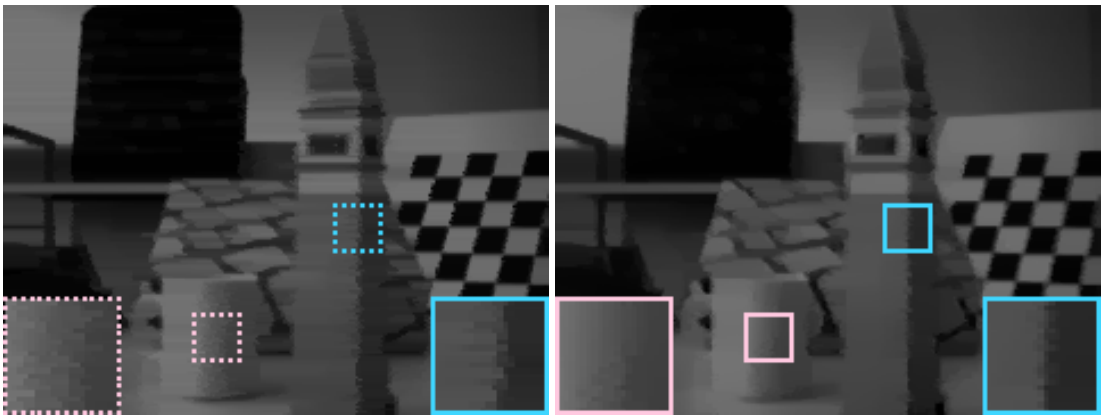
Figure 6.14. Event-based depth estimation.

6.5.3. High frame-rate video synthesis

The task is to reconstruct high frame-rate video frames using a hybrid input of image(s) and events [**2, 6**].

6.5.3.1. Future frame prediction. In this case, we perform future frame prediction, *i.e.*, given a start intensity frame and the subsequent events to predict the future frame. We implement the differentiable model-based reconstruction (DMR) method in [**6**]. Without GEF, the reconstruction performance for the case of "slider_depth" is 25.10 (PSNR) and 0.8237 (SSIM). With GEF, the reconstruction performance improves to 26.63 (PSNR) and 0.8614 (SSIM). For a qualitative comparison, the #5 frame out of 12 reconstructed frames is shown in Fig. 6.15.

6.5.3.2. Motion deblur. GEF can be applied to improve event-based motion deblur [**2**]. Given a blurry image (Fig. 6.16a) and the events captured during the exposure time (Fig. 6.16b), Pan *et al.* [**2**] proposed an event-based double integral (EDI) approach to recover the underlying sharp image(s), as shown in Fig. 6.16c. We employ the same formulation, but use our GEF



(a) w/o GEF (b) w/ GEF

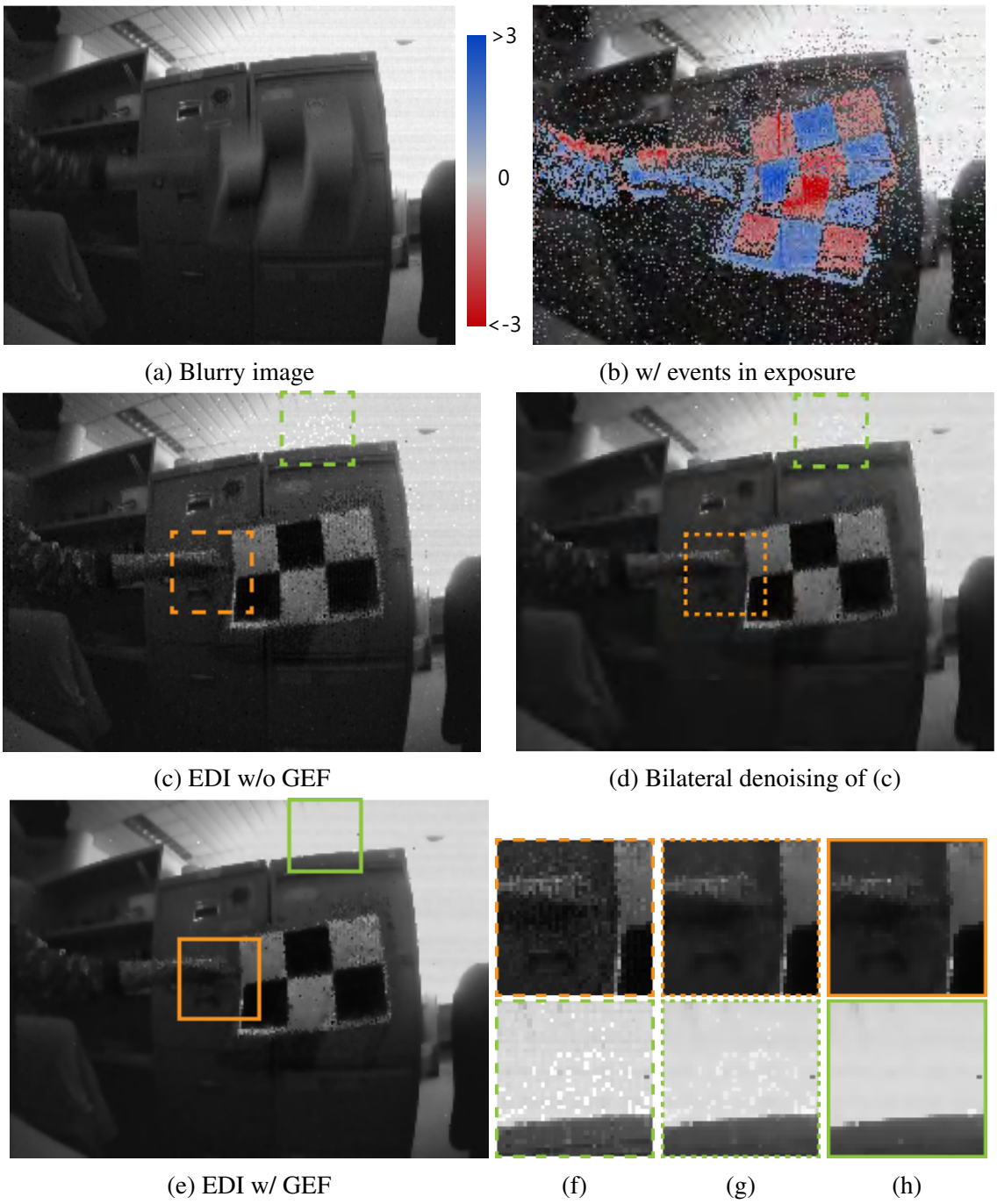
Figure 6.15. Frame prediction using the DMR method in [**6**].

to first filter the events. Note that in this case, the blurry image does not provide useful edge information, we therefore warp neighbor events to form the guidance images. The result is shown in Fig. 6.16e. Even without the guidance of an intensity image, GEF can still reduce the event noise using neighbor events. We further compare the EDI result with denoised EDI output using bilateral filtering, as shown in Fig. 6.16g. Compared to the post-denoising scheme, GEF (Fig. 6.16f) is more effective in eliminating the event noise.

6.5.4. Image reconstruction

In this subsection, we apply GEF for several existing event-based algorithms to address image reconstruction with improved HDR, spatial resolution, and color appearance.

6.5.4.1. HDR. GEF is able to improve HDR image reconstruction because of its effectiveness for motion compensation and denoising. As shown in Figs. 6.17a and 6.17c, the intensity image contains over-exposed regions while the warped event image preserves structures in those regions. We follow a previous approach which employs Poisson reconstruction for HDR reconstruction [**7**]. The difference in our case is that the intensity image is used for reconstruction.

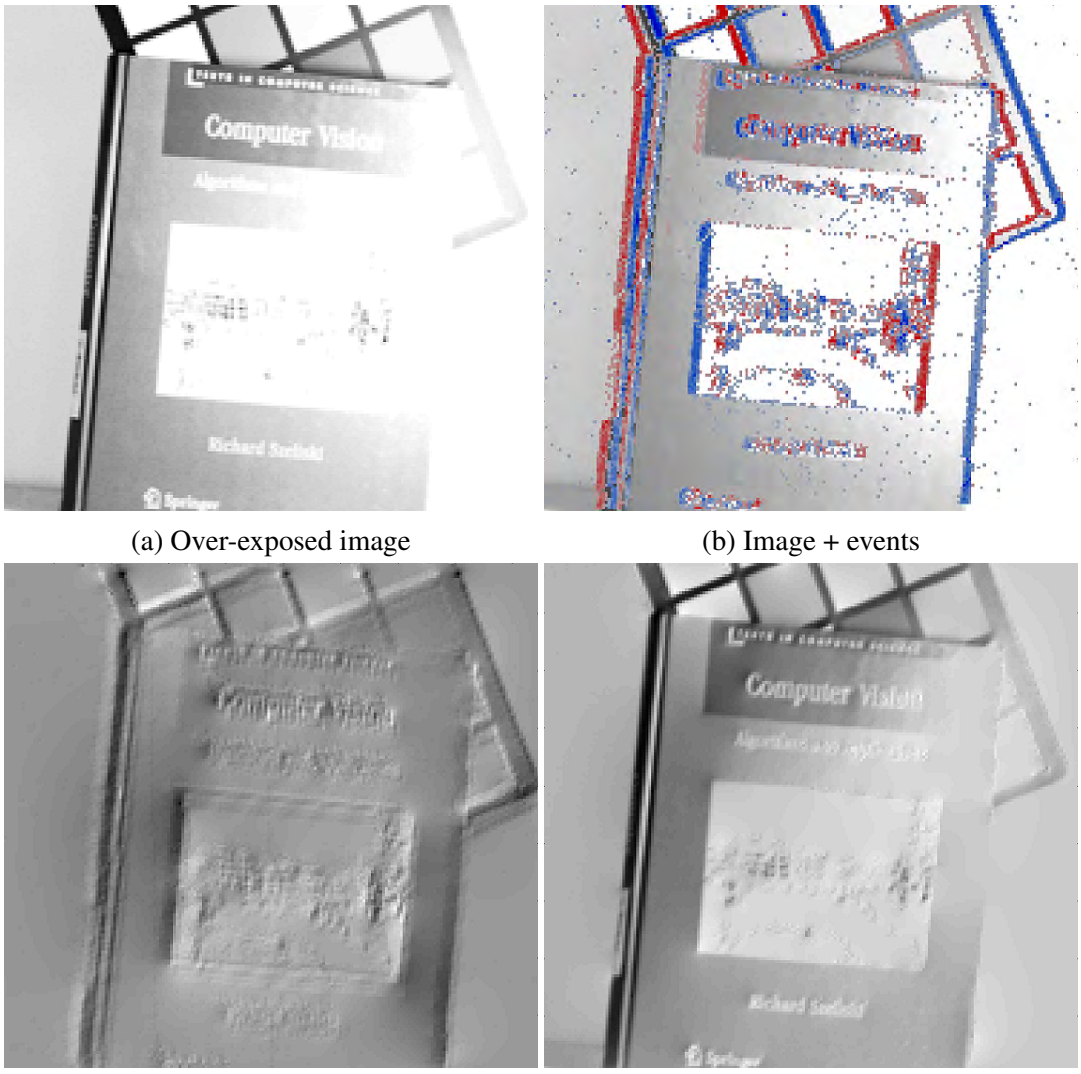


(a) Blurry image (b) w/ events in exposure

(c) EDI w/o GEF (d) Bilateral denoising of (c)

(e) EDI w/ GEF (f) (g) (h)

Figure 6.16. Motion deblur using EDI [**2**]. (f-h) Zoomed-in patches. (e) EDI w/o GEF. (f) EDI result (w/o GEF) + bilateral denoising. (g) EDI w/ GEF.



(a) Over-exposed image (b) Image + events

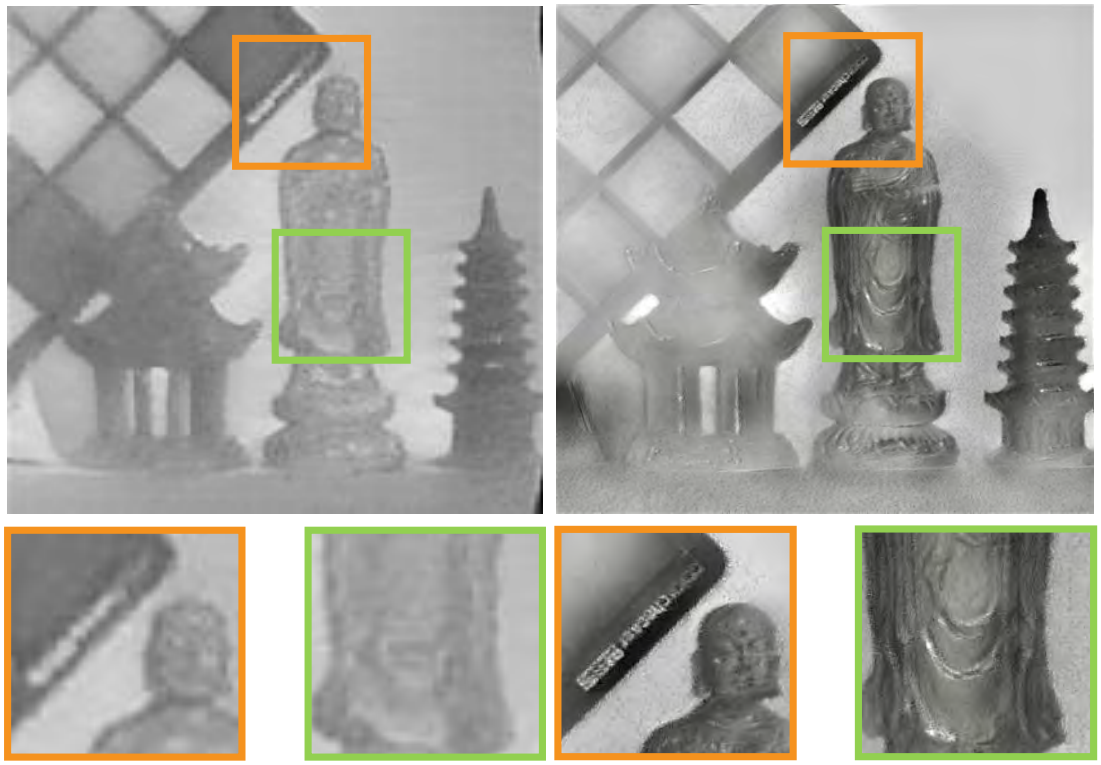
(c) w/o GEF (d) w/ GEF

Figure 6.17. HDR image reconstruction based on Poission method in [**7**]. (a) Low dynamic range image. (b) Overlaid with events. (c) Reconstructed HDR image w/o GEF. (f) Reconstructed HDR image w/ GEF.

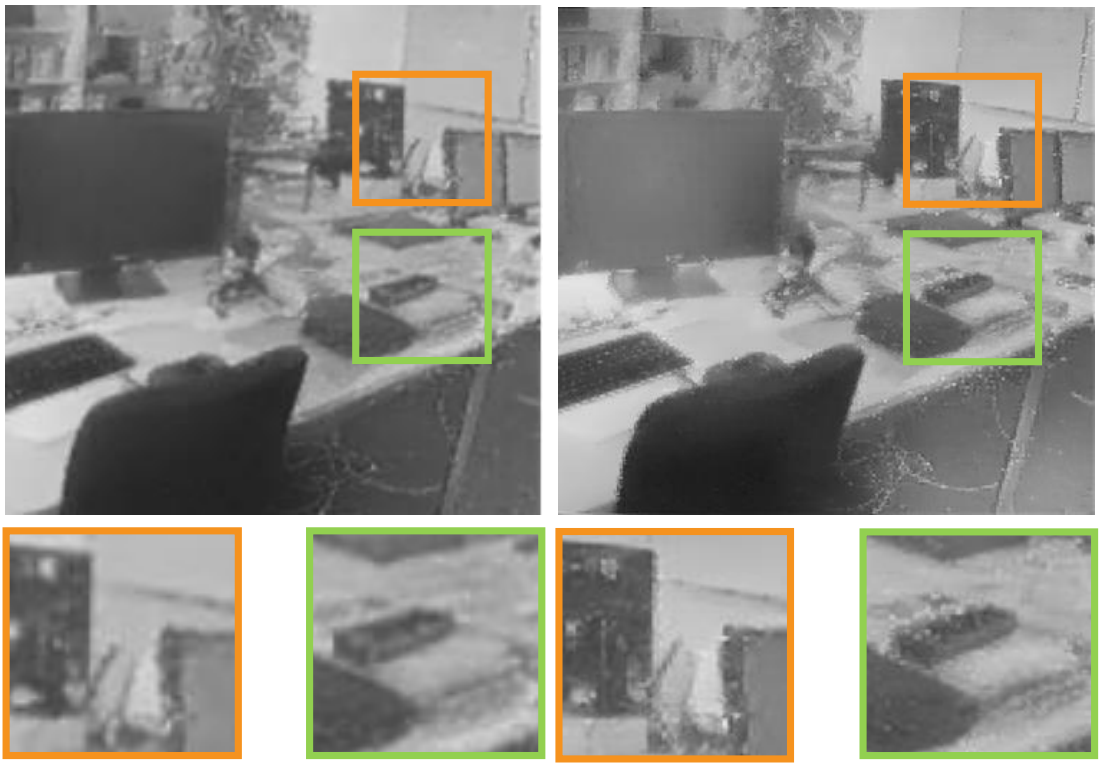
In such case, GEF is applied by setting the warped event image Q^e as guidance and Q^l as filter input. The restored gradient field $\nabla_{xy} I'$ along with the estimated flow $\mathbf{v}$ and the intensity image are then used to reconstruct an HDR image. As can be seen in Figs. 6.17c and 6.17d, the reconstructed HDR image w/ GEF has higher contrast and less artifacts than w/o GEF.

6.5.4.2. Super resolution. The super-resolved events are tasked to restore the intensity image at high resolution. For this task, we use E2VID [**8**] as our backbone image reconstruction algorithm. Figure 6.18 shows two examples of our results. Figure 6.18b is an SR video frame that is synthesized from $4\times$ our image-guided GEF output, it reveals salient textures compared to the result without using GEF (Fig. 6.18a). Figure 6.18d shows an $2\times$ result that synthesized from self-guided GEF. The comparison with Fig. 6.18c shows that GEF can still improve the image quality without image guidance. This application shows the capability of our re-distributed events working with downstream event-based algorithms and the improvement enabled by GEF.

6.5.4.3. Color demosaicking. Color demosaicking is a standard technique for color cameras but remains unestablished for event cameras. As novel event camera prototypes are equipped with the Bayer color filter array [**170**], event color demosacking has become a new and interesting problem to solve. We apply GEF to this task. Previous approach takes the raw mosaicked events directly as input to E2VID [**8**]. To make use of GEF, we first apply an off-the-shelf demosaicking algorithm [**207**] for the RGB image. The optical flow field is estimated using the grayscale image (converted from demosaicked RGB image) and all the events. The filtering step is performed per event color channel. The final 3-channel output events are fed into E2VID for image reconstruction. We compare our filtered events with the original events in terms of image reconstruction, as shown in Figs. 6.19a and 6.19b. The comparison for the reconstructed color images is shown in Figs. 6.19c and 6.19d. Interestingly, with the assistance of GEF, the image produces sharper edges for the letters. This indicates that GEF is able to transfer structural information from demosaicked image to raw events, alleviating the problem of event color demosaicking.



(a) w/o GEF (b) 4× SR w/ GEF



(c) w/o GEF (d) 2× SR w/ GEF

Figure 6.18. Super resolution image reconstruction using E2VID [**8**]. (b) SR by image-guiding GEF. (d) SR by self-guiding GEF.

(a) E2VID [**?**] for R/G/B channel w/o GEF

(b) E2VID [**?**] for R/G/B channel w/ GEF

(c) Color recon. w/o GEF

(d) Color recon. w/ GEF

Figure 6.19. Event-based color image reconstruction

6.6. Conclusion

This paper proposed a novel framework to bridge event-based sensing with frame-based sensing to output a stream of super-resolved yet noise-reduced events. Our experimental results showed that with the assistance of intensity images, performance improvement has been

achieved for flow estimation, event denoising, event super resolution (SR), and demosaicking. Compared to the conference version, this paper added an automatic switch between the image-guiding and the event self-guiding mode, and extended GEF with the event re-distribution module in order to interface seamlessly with downstream event-based algorithms. We also extended the set of applications with depth estimation, super-resolution and color image reconstruction. Besides, this paper derived a modified sensing model for the novel CeleX-V event cameras and adapted GEF for their special output data.

Several future directions may be explored based on our system and dataset, *e.g.*, higher-order motion models, learning-based strategies, task-driven filter design, and more visual applications.

In future, synergistic models can be designed based on this proposed framework. We expect further application of synergistic models in other computational imaging systems [**208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219**].

References

[1] Cedric Scheerlinck, Nick Barnes, and Robert Mahony, "Continuous-time intensity estimation using event cameras," in *Proc. of Asian Conference on Computer Vision (ACCV)*, December 2018.

[2] Liyuan Pan, Cedric Scheerlinck, Xin Yu, Richard Hartley, Miaomiao Liu, and Yuchao Dai, "Bringing a blurry frame alive at high frame-rate with an event camera," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[3] Simon Niklaus, Long Mai, and Feng Liu, "Video frame interpolation via adaptive separable convolution," in *Proc. of International Conference on Computer Vision (ICCV)*, 2017, pp. 261–270.

[4] Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza, "A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3867–3876.

[5] Valentina Vasco, Arren Glover, and Chiara Bartolozzi, "Fast event-based harris corner detection exploiting the advantages of event-driven cameras," in *Proc. of International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4144–4149.

[6] Zihao Winston Wang, Weixin Jiang, Kuan He, Boxin Shi, Aggelos Katsaggelos, and Oliver Cossairt, "Event-driven video frame synthesis," in *Proc. of International Conference on Computer Vision (ICCV) Workshops*, 2019.

[7] Souptik Barua, Yoshitaka Miyatani, and Ashok Veeraraghavan, "Direct face detection and video reconstruction from event cameras," in *Proc. of Winter Conference on Applications of Computer Vision (WACV)*, 2016, pp. 1–9.

[8] Henri Rebecq, Rene Ranftl, Vladlen Koltun, and Davide Scaramuzza, "Events-to-video: Bringing modern computer vision to event cameras," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3857–3866.

[9] Ce Liu, Richard Szeliski, Sing Bing Kang, C Lawrence Zitnick, and William T Freeman, "Automatic estimation and removal of noise from a single image," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 299–314, 2007.

[10] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[11] Ayan Chakrabarti, "Learning sensor multiplexing design through back-propagation," in *Advances in Neural Information Processing Systems*, 2016, pp. 3081–3089.

[12] Vincent Sitzmann, Steven Diamond, Yifan Peng, Xiong Dun, Stephen Boyd, Wolfgang Heidrich, Felix Heide, and Gordon Wetzstein, "End-to-end optimization of optics and image processing for achromatic extended depth of field and super-resolution imaging," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–13, 2018.

[13] Christopher A Metzler, Hayato Ikoma, Yifan Peng, and Gordon Wetzstein, "Deep optics for single-shot high-dynamic-range imaging," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1375–1385.

[14] Qilin Sun, Ethan Tseng, Qiang Fu, Wolfgang Heidrich, and Felix Heide, "Learning rank-1 diffractive optics for single-shot high dynamic range imaging," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1386–1396.

[15] Tadashi Okawara, Michitaka Yoshida, Hajime Nagahara, and Yasushi Yagi, "Action recognition from a single coded image," in *2020 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2020, pp. 1–11.

[16] Julien NP Martel, Lorenz Mueller, Stephen J Carey, Piotr Dudek, and Gordon Wetzstein, "Neural sensors: Learning pixel exposures for hdr imaging and video compressive sensing with programmable sensors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[17] Harel Haim, Shay Elmalem, Raja Giryes, Alex M Bronstein, and Emanuel Marom, "Depth estimation from a single image using deep learned phase coded mask," *IEEE Transactions on Computational Imaging*, vol. 4, no. 3, pp. 298–310, 2018.

[18] Ulugbek S Kamilov, Ioannis N Papadopoulos, Morteza H Shoreh, Alexandre Goy, Cedric Vonesch, Michael Unser, and Demetri Psaltis, "Learning approach to optical tomography," *Optica*, vol. 2, no. 6, pp. 517–522, 2015.

[19] Praneeth Chakravarthula, Yifan Peng, Joel Kollin, Henry Fuchs, and Felix Heide, "Wirtinger holography for near-eye displays," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–13, 2019.

[20] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck, "A 240$\times$ 180 130 db 3 μs latency global shutter spatiotemporal vision sensor," *Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.

[21] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck, "A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change," in *IEEE International Solid-State Circuits Conference*, 2006, pp. 2060–2069.

[22] Patrick Bardow, Andrew J Davison, and Stefan Leutenegger, "Simultaneous optical flow and intensity estimation from an event camera," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 884–892.

[23] Lin Wang, Yo-Sung Ho, Kuk-Jin Yoon, et al., "Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[24] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam," *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 142–149, 2017.

[25] Johannes Kopf, Michael F Cohen, Dani Lischinski, and Matt Uyttendaele, "Joint bilateral upsampling," in *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH)*. ACM, 2007, vol. 26, p. 96.

[26] Derek Chan, Hylke Buisman, Christian Theobalt, and Sebastian Thrun, "A noise-aware filter for real-time depth upsampling," in *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications-M2SFA2 2008*, 2008.

[27] Yijun Li, Jia-Bin Huang, Ahuja Narendra, and Ming-Hsuan Yang, "Deep joint image filtering," in *Proc. of European Conference on Computer Vision (ECCV)*. Springer, 2016.

[28] Pravin Bhat, C Lawrence Zitnick, Noah Snavely, Aseem Agarwala, Maneesh Agrawala, Michael Cohen, Brian Curless, and Sing Bing Kang, "Using photographs to enhance videos of a static scene," in *Proc. of the 18th Eurographics conference on Rendering Techniques*. Eurographics Association, 2007, pp. 327–338.

[29] Ankit Gupta, Pravin Bhat, Mira Dontcheva, Oliver Deussen, Brian Curless, and Michael Cohen, "Enhancing and experiencing spacetime resolution with videos and stills," in *Proc. of International Conference on Computational Photography (ICCP)*, 2009, pp. 1–9.

[30] M. Gupta, A. Agrawal, A. Veeraraghavan, and S. G. Narasimhan, "Flexible voxels for motion-aware videography," in *Proc. of European Conference on Computer Vision (ECCV)*, 2010, pp. 100–114.

[31] Falk Schubert and Krystian Mikolajczyk, "Combining high-resolution images with low-quality videos.," in *Proc. of British Machine Vision Conference (BMVC)*, 2008, pp. 1–10.

[32] Hagit Zabrodsky and Shmuel Peleg, "Attentive transmission," *Journal of Visual Communication and Image Representation*, vol. 1, no. 2, pp. 189–198, 1990.

[33] Richard G Baraniuk, Thomas Goldstein, Aswin C Sankaranarayanan, Christoph Studer, Ashok Veeraraghavan, and Michael B Wakin, "Compressive video sensing: algorithms, architectures, and applications," *IEEE Signal Processing Magazine*, vol. 34, no. 1, pp. 52–66, 2017.

[34] Kuan He, Zihao Wang, Xiang Huang, Xiaolei Wang, Seunghwan Yoo, Pablo Ruiz, Itay Gdor, Alan Selewa, Nicola J Ferrier, Norbert Scherer, et al., "Computational multifocal microscopy," *Biomedical Optics Express*, vol. 9, no. 12, pp. 6477–6496, 2018.

[35] Michael Iliadis, Leonidas Spinoulas, and Aggelos K Katsaggelos, "Deep fully-connected networks for video compressive sensing," *Digital Signal Processing*, vol. 72, pp. 9–18, 2018.

[36] P. Llull, X. Liao, X. Yuan, J. Yang, D. Kittle, L. Carin, G. Sapiro, and D. J. Brady, "Coded aperture compressive temporal imaging," *Opt. Express*, vol. 21, no. 9, pp. 10526–10545, May 2013.

[37] D. Reddy, A. Veeraraghavan, and R. Chellappa, "P2C2: Programmable pixel compressive camera for high speed imaging," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2011, pp. 329–336.

[38] Vladimir Stanković, Lina Stanković, and Samuel Cheng, "Compressive video sampling," in *European Signal Processing Conference*. IEEE, 2008, pp. 1–5.

[39] Zihao Wang, Leonidas Spinoulas, Kuan He, Lei Tian, Oliver Cossairt, Aggelos K Katsaggelos, and Huaijin Chen, "Compressive holographic video," *Optics Express*, vol. 25, no. 1, pp. 250–262, 2017.

[40] Gottfried Munda, Christian Reinbacher, and Thomas Pock, "Real-time intensity-image reconstruction for event cameras using manifold regularisation," vol. 126, no. 12, pp. 1381–1393, 2018.

[41] Lin Wang, Yo-Sung Ho, Kuk-Jin Yoon, et al., "Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[42] Byeong-Doo Choi, Jong-Woo Han, Chang-Su Kim, and Sung-Jea Ko, "Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 4, pp. 407–416, 2007.

[43] Ravi Krishnamurthy, John W Woods, and Pierre Moulin, "Frame interpolation and bidirectional prediction of video using compactly encoded optical-flow fields and label fields," *IEEE transactions on circuits and systems for video technology*, vol. 9, no. 5, pp. 713–726, 1999.

[44] Martin Luessi and Aggelos K Katsaggelos, "Efficient motion compensated frame rate up-conversion using multiple interpolations and median filtering," in *Proc. of International Conference on Image Processing (ICIP)*, 2009, pp. 373–376.

[45] James R Bergen, Patrick Anandan, Keith J Hanna, and Rajesh Hingorani, "Hierarchical model-based motion estimation," in *Proc. of European Conference on Computer Vision (ECCV)*. Springer, 1992, pp. 237–252.

[46] Andreas Wedel, Thomas Pock, Jürgen Braun, Uwe Franke, and Daniel Cremers, "Duality tv-l1 flow with fundamental matrix prior," in *Proc. of the IEEE 23rd International Conference on Image and Vision Computing*, 2008, pp. 1–6.

[47] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, vol. 2.

[48] Ziwei Liu, Raymond Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala, "Video frame synthesis using deep voxel flow," in *Proc. of International Conference on Computer Vision (ICCV)*, 2017, vol. 2.

[49] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz, "Super slomo: High quality estimation of multiple intermediate frames for

video interpolation," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 9000–9008.

[50] Michael Mathieu, Camille Couprie, and Yann LeCun, "Deep multi-scale video prediction beyond mean square error," *arXiv preprint arXiv:1511.05440*, 2015.

[51] Prateep Bhattacharjee and Sukhendu Das, "Temporal coherency based criteria for predicting video frames using deep multi-stage generative adversarial networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 4271–4280.

[52] Tianfan Xue, Jiajun Wu, Katherine Bouman, and Bill Freeman, "Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks," in *Advances in Neural Information Processing Systems 29*, pp. 91–99. Curran Associates, Inc., 2016.

[53] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas, "Deblurgan: Blind motion deblurring using conditional adversarial networks," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8183–8192.

[54] Xinyi Zhang, Hang Dong, Zhe Hu, Wei-Sheng Lai, Fei Wang, and Ming-Hsuan Yang, "Gated fusion network for joint image deblurring and super-resolution," in *Proc. of British Machine Vision Conference (BMVC)*, 2018.

[55] Meiguang Jin, Givi Meishvili, and Paolo Favaro, "Learning to extract a video sequence from a single motion-blurred image," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6334–6342.

[56] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[57] Kai Zhang, Wangmeng Zuo, and Lei Zhang, "Ffdnet: Toward a fast and flexible solution for cnn based image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, 2018.

[58] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey, "Need for speed: A benchmark for higher frame rate object tracking," in *Proc. of International Conference on Computer Vision (ICCV)*, 2017, pp. 1134–1143.

[59] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg, "Plug-and-play priors for model based reconstruction," in *Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2013, pp. 945–948.

[60] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[61] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza, "ESIM: an open event camera simulator," *Conf. on Robotics Learning (CoRL)*, Oct. 2018.

[62] Jian Su, Xingpeng Yan, Yingqing Huang, Xiaoyu Jiang, Yibei Chen, and Teng Zhang, "Progress in the synthetic holographic stereogram printing technique," *Applied Sciences*, vol. 8, no. 6, pp. 851, 2018.

[63] Dominic J DeBitetto, "Holographic panoramic stereograms synthesized from white light recordings," *Applied optics*, vol. 8, no. 8, pp. 1740–1741, 1969.

[64] Parameswaran Hariharan and P Hariharan, *Optical Holography: Principles, techniques and applications*, Cambridge University Press, 1996.

[65] Derek Tseng, Onur Mudanyali, Cetin Oztoprak, Serhan O Isikman, Ikbal Sencan, Oguzhan Yaglidere, and Aydogan Ozcan, "Lensfree microscopy on a cellphone," *Lab on a Chip*, vol. 10, no. 14, pp. 1787–1792, 2010.

[66] Waheb Bishara, Uzair Sikora, Onur Mudanyali, Ting-Wei Su, Oguzhan Yaglidere, Shirley Luckhart, and Aydogan Ozcan, "Holographic pixel super-resolution in portable lensless on-chip microscopy using a fiber-optic array," *Lab on a Chip*, vol. 11, no. 7, pp. 1276–1279, 2011.

[67] Alon Greenbaum, Yibo Zhang, Alborz Feizi, Ping-Luen Chung, Wei Luo, Shivani R Kandukuri, and Aydogan Ozcan, "Wide-field computational imaging of pathology slides using lens-free on-chip microscopy," *Science translational medicine*, vol. 6, no. 267, pp. 267ra175–267ra175, 2014.

[68] Euan McLeod, T Umut Dincer, Muhammed Veli, Yavuz N Ertas, Chau Nguyen, Wei Luo, Alon Greenbaum, Alborz Feizi, and Aydogan Ozcan, "High-throughput and label-free single nanoparticle sizing based on time-resolved on-chip microscopy," *ACS nano*, vol. 9, no. 3, pp. 3265–3273, 2015.

[69] Mingjun Wang and Jigang Wu, "Iterative digital in-line holographic reconstruction with improved resolution by data interpolation," in *Holography, Diffractive Optics, and Applications VI*. International Society for Optics and Photonics, 2014, vol. 9271, p. 927110.

[70] Shaodong Feng, Mingjun Wang, and Jigang Wu, "Digital in-line holographic microscope based on the grating illumination with improved resolution by interpolation," in

Holography, Diffractive Optics, and Applications VII. International Society for Optics and Photonics, 2016, vol. 10022, p. 1002205.

[71] Yair Rivenson, Yichen Wu, Hongda Wang, Yibo Zhang, Alborz Feizi, and Aydogan Ozcan, "Sparsity-based multi-height phase recovery in holographic microscopy," *Scientific reports*, vol. 6, pp. 37862, 2016.

[72] Ting-Wei Su, Liang Xue, and Aydogan Ozcan, "High-throughput lensfree 3d tracking of human sperms reveals rare statistics of helical trajectories," *Proceedings of the National Academy of Sciences*, vol. 109, no. 40, pp. 16018–16022, 2012.

[73] John Weidling, Serhan O Isikman, Alon Greenbaum, Aydogan Ozcan, and Elliot L Botvinick, "Lens-free computational imaging of capillary morphogenesis within three-dimensional substrates," *Journal of biomedical optics*, vol. 17, no. 12, pp. 126018, 2012.

[74] Ivan Pushkarsky, Yunbo Liu, Westbrook Weaver, Ting-Wei Su, Onur Mudanyali, Aydogan Ozcan, and Dino Di Carlo, "Automated single-cell motility analysis on a chip using lensfree microscopy," *Scientific reports*, vol. 4, pp. 4717, 2014.

[75] G. Bub, M. Tecza, M. Helmes, P. Lee, and P. Kohl, "Temporal pixel multiplexing for simultaneous high-speed, high-resolution imaging," *Nature Methods*, vol. 7, pp. 209–U66, 2010.

[76] A. Agrawal, M. Gupta, A. Veeraraghavan, and S. G. Narasimhan, "Optimal coded sampling for temporal super-resolution," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 599–606.

[77] D. Liu, J. Gu, Y. Hitomi, M. Gupta, T. Mitsunaga, and S. K. Nayar, "Efficient space-time sampling with pixel-wise coded exposure for high-speed imaging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 248–260, Feb. 2014.

[78] J. Holloway, A. C. Sankaranarayanan, A. Veeraraghavan, and S. Tambe, "Flutter shutter video camera for compressive sensing of videos," in *IEEE Int. Conf. Computational Photography*, April 2012, pp. 1–9.

[79] Liang Gao, Jinyang Liang, Chiye Li, and Lihong V Wang, "Single-shot compressed ultrafast photography at one hundred billion frames per second," *Nature*, vol. 516, no. 7529, pp. 74–77, 2014.

[80] Stanislav Hrivňak, Jozef Uličnỳ, Ladislav Mikeš, Angelica Cecilia, Elias Hamann, Tilo Baumbach, Libor Švéda, Zdenko Zápražnỳ, Dušan Korytár, Eva Gimenez-Navarro, et al.,

"Single-distance phase retrieval algorithm for bragg magnifier microscope," *Optics Express*, vol. 24, no. 24, pp. 27753–27762, 2016.

[81] James R Fienup, "Reconstruction of an object from the modulus of its fourier transform," *Optics letters*, vol. 3, no. 1, pp. 27–29, 1978.

[82] James R Fienup, "Phase retrieval algorithms: a comparison," *Applied optics*, vol. 21, no. 15, pp. 2758–2769, 1982.

[83] D. J. Brady, K. Choi, D. L. Marks, R. Horisaki, and S. Lim, "Compressive holography," *Opt. Express*, vol. 17, no. 15, pp. 13040–13049, July 2009.

[84] Loïc Denis, Dirk Lorenz, Eric Thiébaut, Corinne Fournier, and Dennis Trede, "Inline hologram reconstruction with sparsity constraints," *Optics letters*, vol. 34, no. 22, pp. 3475–3477, 2009.

[85] Joonku Hahn, Sehoon Lim, Kerkil Choi, Ryoichi Horisaki, and David J. Brady, "Video-rate compressive holographic microscopic tomography," *Opt. Express*, vol. 19, no. 8, pp. 7289–7298, Apr. 2011.

[86] Alexander Szameit, Yoav Shechtman, E Osherovich, Elad Bullkich, Pavel Sidorenko, Hod Dana, S Steiner, Ernst B Kley, Snir Gazit, Tzipi Cohen-Hyams, et al., "Sparsity-based single-shot subwavelength coherent diffractive imaging," *Nature materials*, vol. 11, no. 5, pp. 455–459, 2012.

[87] Y. Rivenson, A. Stern, and B. Javidi, "Compressive fresnel holography," *J. Display Technol.*, vol. 6, no. 10, pp. 506–509, Oct. 2010.

[88] Conor P McElhinney, John B McDonald, Albertina Castro, Yann Frauel, Bahram Javidi, and Thomas J Naughton, "Depth-independent segmentation of macroscopic three-dimensional objects encoded in single perspectives of digital holograms," *Optics letters*, vol. 32, no. 10, pp. 1229–1231, 2007.

[89] PWM Tsang, KWK Cheung, T Kim, You Seok Kim, and T-C Poon, "Fast reconstruction of sectional images in digital holography," *Optics letters*, vol. 36, no. 14, pp. 2650–2652, 2011.

[90] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489 – 509, feb. 2006.

[91] E. J. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Trans. Inform. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[92] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory,*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[93] Yair Rivenson, Adrian Stern, and Joseph Rosen, "Reconstruction guarantees for compressive tomographic holography," *Optics letters*, vol. 38, no. 14, pp. 2509–2511, 2013.

[94] Ralph Gerchberg and W Saxton, "A practical algorithm for the determination of phase from image and diffraction plane pictures," *Optik*, vol. 35, pp. 237, 1972.

[95] Andrew V Martin, Fenglin Wang, ND Loh, Tomas Ekeberg, Filipe RNC Maia, Max Hantke, Gijs van der Schot, Christina Y Hampton, Raymond G Sierra, Andrew Aquila, et al., "Noise-robust coherent diffractive imaging with a single diffraction pattern," *Optics Express*, vol. 20, no. 15, pp. 16650–16661, 2012.

[96] Emmanuel J Candes, Thomas Strohmer, and Vladislav Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Communications on Pure and Applied Mathematics*, vol. 66, no. 8, pp. 1241–1274, 2013.

[97] Emmanuel J Candes, Yonina C Eldar, Thomas Strohmer, and Vladislav Voroninski, "Phase retrieval via matrix completion," *SIAM review*, vol. 57, no. 2, pp. 225–251, 2015.

[98] Subhadip Mukherjee and Chandra Sekhar Seelamantula, "An iterative algorithm for phase retrieval with sparsity constraints: application to frequency domain optical coherence tomography," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 553–556.

[99] Yoav Shechtman, Yonina C Eldar, Alexander Szameit, and Mordechai Segev, "Sparsity based sub-wavelength imaging with partially incoherent light via quadratic compressed sensing," *Optics express*, vol. 19, no. 16, pp. 14807–14822, 2011.

[100] Ryoichi Horisaki, Yusuke Ogura, Masahiko Aino, and Jun Tanida, "Single-shot phase imaging with a coded aperture," *Opt. Lett.*, vol. 39, no. 22, pp. 6466–6469, Nov. 2014.

[101] M. Ben-Ezra and S. K. Nayar, "Motion-based Motion Deblurring," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 689–698, Jun. 2004.

[102] Yu-Wing Tai, Hao Du, Michael S Brown, and Stephen Lin, "Correction of spatially varying image and video motion blur using a hybrid camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 1012–1028, 2010.

[103] Ramesh Raskar, Amit Agrawal, and Jack Tumblin, "Coded exposure photography: Motion deblurring using fluttered shutter," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 795–804, July 2006.

[104] Jinwei Gu, Yasunobu Hitomi, Tomoo Mitsunaga, and Shree Nayar, "Coded rolling shutter photography: Flexible space-time sampling," in *Computational Photography (ICCP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1–8.

[105] Donghun Ryu, Zihao Wang, Kuan He, Guoan Zheng, Roarke Horstmeyer, and Oliver Cossairt, "Subsampled phase retrieval for temporal resolution enhancement in lensless on-chip holographic video," *Biomedical Optics Express*, vol. 8, no. 3, pp. 1981–1995, 2017.

[106] Rw W Gerchberg, "Phase determination for image and diffraction plane pictures in the electron microscope," *Optik (Stuttgart)*, vol. 34, pp. 275, 1971.

[107] M. K. Kim, *Digital Holographic Microscopy*, Springer, 2011.

[108] David L Donoho, Michael Elad, and Vladimir N Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Transactions on information theory*, vol. 52, no. 1, pp. 6–18, 2006.

[109] Michal Aharon, Michael Elad, and Alfred Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[110] Jianchao Yang, Zhaowen Wang, Zhe Lin, Xianbiao Shu, and Thomas Huang, "Bilevel sparse coding for coupled feature spaces," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 2360–2367.

[111] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma, "Image super-resolution via sparse representation," *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[112] Ryo Nakagaki and Aggelos K Katsaggelos, "A vq-based blind image restoration algorithm," *IEEE transactions on image processing*, vol. 12, no. 9, pp. 1044–1053, 2003.

[113] Lei Tian, Nick Loomis, José A Domínguez-Caballero, and George Barbastathis, "Quantitative measurement of size and three-dimensional position of fast-moving bubbles in air-water mixture flows using digital holography," *Applied Optics*, vol. 49, no. 9, pp. 1549–1554, 2010.

[114] David G Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Ieee, 1999, vol. 2, pp. 1150–1157.

[115] Vijayaraghavan Thirumalai and Pascal Frossard, "Correlation estimation from compressed images," *Journal of Visual Communication and Image Representation*, vol. 24, no. 6, pp. 649–660, 2013.

[116] Aswin C Sankaranarayanan, Pavan K Turaga, Richard G Baraniuk, and Rama Chellappa, "Compressive acquisition of dynamic scenes," in *European Conference on Computer Vision*. Springer, 2010, pp. 129–142.

[117] Pradeep Nagesh and Baoxin Li, "A compressive sensing approach for expression-invariant face recognition," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1518–1525.

[118] Suhas Lohit, Kuldeep Kulkarni, Pavan Turaga, Jian Wang, and Aswin C Sankaranarayanan, "Reconstruction-free inference on compressive measurements," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 16–24.

[119] Kuldeep Kulkarni and Pavan Turaga, "Reconstruction-free action inference from compressive imagers," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 4, pp. 772–784, 2016.

[120] Francesco Pittaluga and Sanjeev Jagannatha Koppal, "Pre-capture privacy for small vision sensors," *IEEE TPAMI*, vol. 39, no. 11, pp. 2215–2226, 2017.

[121] Shota Nakashima, Yuhki Kitazono, Lifeng Zhang, and Seiichi Serikawa, "Development of privacy-preserving sensor for person detection," *Procedia-Social and Behavioral Sciences*, vol. 2, no. 1, pp. 213–217, 2010.

[122] Leonidas Spinoulas, Oliver S. Cossairt, Aggelos K. Katsaggelos, Patrick Gill, and David G. Stork, "Performance comparison of ultra-miniature diffraction gratings with lenses and zone plates," in *Imaging and Applied Optics 2015*. 2015, p. CM3E.1, Optical Society of America.

[123] Nick Antipa, Grace Kuo, Reinhard Heckel, Ben Mildenhall, Emrah Bostan, Ren Ng, and Laura Waller, "Diffusercam: lensless single-exposure 3d imaging," *Optica*, vol. 5, no. 1, pp. 1–9, 2018.

[124] Ankur Chattopadhyay and Terrance E Boult, "Privacycam: A privacy preserving camera using uCLinux on the Blackfin DSP," in *CVPR*, 2007, pp. 1–8.

[125] Lieven De Strycker, Pascale Termont, Jan Vandewege, J Haitsma, AACM Kalker, Marc Maes, and Geert Depovere, "Implementation of a real-time digital watermarking process for broadcast monitoring on a TriMedia VLIW processor," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 147, no. 4, pp. 371–376, 2000.

[126] Elias Kougianos, Saraju P Mohanty, and Rabi N Mahapatra, "Hardware assisted watermarking for multimedia," *Computers & Electrical Engineering*, vol. 35, no. 2, pp. 339–358, 2009.

[127] TM Cannon and EE Fenimore, "Coded aperture imaging: Many holes make light work," *Optical Engineering*, vol. 19, no. 3, pp. 193283, 1980.

[128] RH Dicke, "Scatter-hole cameras for x-rays and gamma rays," *The Astrophysical Journal*, vol. 153, pp. L101, 1968.

[129] E. E. Fenimore and T. M. Cannon, "Coded aperture imaging with uniformly redundant arrays.," *Appl. Opt.*, vol. 17, no. 3, pp. 337–347, 1978.

[130] Michael J DeWeert and Brian P Farm, "Lensless coded-aperture imaging with separable Doubly-Toeplitz masks," *Optical Engineering*, vol. 54, no. 2, pp. 023102, 2015.

[131] Edward R Dowski and W Thomas Cathey, "Extended depth of field through wave-front coding," *Applied Optics*, vol. 34, no. 11, pp. 1859–1866, 1995.

[132] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM Trans. Graph.*, vol. 26, no. 3, July 2007.

[133] Chia-Kai Liang, Tai-Hsu Lin, Bing-Yi Wong, Chi Liu, and Homer H Chen, "Programmable aperture photography: Multiplexed light field acquisition," in *ACM Transactions on Graphics (TOG)*, 2008, vol. 27, p. 55.

[134] Ashok Veeraraghavan, Ramesh Raskar, Amit Agrawal, Ankit Mohan, and Jack Tumblin, "Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing," *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 69, 2007.

[135] Jesse K Adams, Vivek Boominathan, Benjamin W Avants, Daniel G Vercosa, Fan Ye, Richard G Baraniuk, Jacob T Robinson, and Ashok Veeraraghavan, "Single-frame 3D fluorescence microscopy with ultraminiature lensless FlatScope," *Science Advances*, vol. 3, no. 12, pp. e1701548, 2017.

[136] M Salman Asif, Ali Ayremlou, Aswin Sankaranarayanan, Ashok Veeraraghavan, and Richard G Baraniuk, "Flatcam: Thin, lensless cameras using coded aperture and computation," *IEEE Transactions on Computational Imaging*, vol. 3, no. 3, pp. 384–397, 2017.

[137] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld, "Learning realistic human actions from movies," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[138] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu, "Action recognition by dense trajectories," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 3169–3176.

[139] Karen Simonyan and Andrew Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.

[140] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proc. of International Conference on Computer Vision (ICCV)*, 2015, pp. 4489–4497.

[141] Christoph Feichtenhofer, Axel Pinz, Richard P Wildes, and Andrew Zisserman, "What have we learned from deep representations for action recognition?," *Connections*, vol. 19, pp. 29, 2018.

[142] Ji Dai, Jonathan Wu, Behrouz Saghafi, Janusz Konrad, and Prakash Ishwar, "Towards privacy-preserving activity recognition using extremely low temporal and spatial resolution cameras," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2015, pp. 68–76.

[143] Michael S Ryoo, Kiyoon Kim, and Hyun Jong Yang, "Extreme low resolution activity recognition with multi-siamese embedding learning," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[144] Zhongzheng Ren, Yong Jae Lee, and Michael S Ryoo, "Learning to anonymize faces for privacy preserving action detection," in *Proc. of European Conference on Computer Vision (ECCV)*, 2018, pp. 620–636.

[145] Nisarg Raval, Ashwin Machanavajjhala, and Landon P Cox, "Protecting visual secrets using adversarial nets," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1329–1332.

[146] Francesco Pittaluga, Sanjeev Koppal, and Ayan Chakrabarti, "Learning privacy preserving encodings through adversarial training," in *Proc. of Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 791–799.

[147] Zhenyu Wu, Zhangyang Wang, Zhaowen Wang, and Hailin Jin, "Towards privacy-preserving visual recognition via adversarial training: A pilot study," in *Proc. of European Conference on Computer Vision (ECCV)*, 2018, pp. 606–624.

[148] B Srinivasa Reddy and Biswanath N Chatterji, "An FFT-based technique for translation, rotation, and scale-invariant image registration," *IEEE Transactions on Image Processing*, vol. 5, no. 8, pp. 1266–1271, 1996.

[149] Vasileios Argyriou and Theodore Vlachos, "A study of sub-pixel motion estimation using phase correlation.," in *Proc. of British Machine Vision Conference (BMVC)*, 2006, pp. 387–396.

[150] Huy Tho Ho and Roland Goecke, "Optical flow estimation using Fourier Mellin transform," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[151] A Murat Tekalp, *Digital video processing*, Prentice Hall Press, 2015.

[152] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[153] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[154] Stephan Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun, "Playing for data: Ground truth from computer games," in *Proc. of European Conference on Computer Vision (ECCV)*, 2018.

[155] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[156] James R Fienup, "Phase retrieval algorithms: A comparison," *Applied Optics*, vol. 21, no. 15, pp. 2758–2769, 1982.

[157] Ori Katz, Pierre Heidmann, Mathias Fink, and Sylvain Gigan, "Non-invasive single-shot imaging through scattering layers and around corners via speckle correlations," *Nature Photonics*, vol. 8, no. 10, pp. 784, 2014.

[158] Feng Li, Zijia Li, David Saunders, and Jingyi Yu, "A theory of coprime blurred pairs," in *Proc. of International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 217–224.

[159] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck, "A 128×128 120 db 15 μs latency asynchronous temporal contrast vision sensor," *IEEE journal of solid-state circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[160] Bongki Son, Yunjae Suh, Sungho Kim, Heejae Jung, Jun-Seok Kim, Changwoo Shin, Keunju Park, Kyoobin Lee, Jinman Park, Jooyeon Woo, et al., "A 640×480 dynamic vision sensor with a 9μm pixel and 300Meps address-event representation," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2017, pp. 66–67.

[161] Bodo Rueckauer and Tobi Delbruck, "Evaluation of event-based algorithms for optical flow with ground-truth from inertial measurement sensor," *Frontiers in Neuroscience*, vol. 10, pp. 176, 2016.

[162] Jin Han, Chu Zhou, Peiqi Duan, Yehui Tang, Chang Xu, Chao Xu, Tiejun Huang, and Boxin Shi, "Neuromorphic camera guided high dynamic range imaging," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[163] Lan Xu, Weipeng Xu, Vladislav Golyanik, Marc Habermann, Lu Fang, and Christian Theobalt, "Eventcap: Monocular 3d capture of high-speed human motions using an event camera," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[164] Henri Rebecq, Guillermo Gallego, Elias Mueggler, and Davide Scaramuzza, "EMVS: Event-based multi-view stereo—3D reconstruction with an event camera in real-time," *International Journal of Computer Vision*, vol. 126, no. 12, pp. 1394–1414, 2018.

[165] Elias Mueggler, Basil Huber, and Davide Scaramuzza, "Event-based, 6-DoF pose tracking for high-speed maneuvers," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 2761–2768.

[166] Hanme Kim, Stefan Leutenegger, and Andrew J. Davison, "Real-time 3D reconstruction and 6-DoF tracking with an event camera," in *Proc. of European Conference on Computer Vision (ECCV)*. Springer, 2016.

[167] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza, "Ultimate SLAM? combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 994–1001, 2018.

[168] Ana I. Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza, "Event-based vision meets deep learning on steering prediction for self-driving cars," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[169] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Joerg Conradt, Kostas Daniilidis, et al., "Event-based vision: A survey," *arXiv preprint arXiv:1904.08405*, 2019.

[170] Cedric Scheerlinck, Henri Rebecq, Timo Stoffregen, Nick Barnes, Robert Mahony, and Davide Scaramuzza, "CED: Color event camera dataset," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019, pp. 0–0.

[171] Shoushun Chen and Menghan Guo, "Live demonstration: Celex-v: a 1m pixel multi-mode event-based sensor," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019.

[172] David L Donoho, "De-noising by soft-thresholding," *IEEE transactions on information theory*, vol. 41, no. 3, pp. 613–627, 1995.

[173] Alireza Khodamoradi and Ryan Kastner, "O(N)-space spatiotemporal filter for reducing noise in neuromorphic vision sensors," *IEEE Transactions on Emerging Topics in Computing*, 2018.

[174] Vandana Padala, Arindam Basu, and Garrick Orchard, "A noise filtering algorithm for event-based asynchronous change detection image sensors on truenorth and its implementation on truenorth," *Frontiers in Neuroscience*, vol. 12, pp. 118, 2018.

[175] Daniel Czech and Garrick Orchard, "Evaluating noise filtering for event-based asynchronous change detection image sensors," in *6th International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2016, pp. 19–24.

[176] Hongjie Liu, Christian Brandli, Chenghan Li, Shih-Chii Liu, and Tobi Delbruck, "Design of a spatiotemporal correlation filter for event-based sensors," in *Proc. of International Symposium on Circuits and Systems (ISCAS)*, 2015, pp. 722–725.

[177] Wensheng Cheng, Hao Luo, Wen Yang, Lei Yu, Shoushun Chen, and Wei Li, "DET: A high-resolution DVS dataset for lane extraction," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR) workshops*, 2019.

[178] Yanxiang Wang, Bowen Du, Yiran Shen, Kai Wu, Guangrong Zhao, Jianguo Sun, and Hongkai Wen, "EV-Gait: Event-based robust gait recognition using dynamic vision sensors," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[179] R. Wes Baldwin, Mohammed Almatrafi, Vijayan Asari, and Keigo Hirakawa, "Event probability mask (epm) and event denoising convolutional neural network (edncnn) for neuromorphic cameras," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[180] Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozzi, "Event-based visual flow," *transactions on neural networks and learning systems*, vol. 25, no. 2, pp. 407–417, 2013.

[181] Mohammed Mutlaq Almatrafi, Raymond Baldwin, Kiyoharu Aizawa, and Keigo Hirakawa, "Distance surface for event-based optical flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[182] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 989–997.

[183] Guillermo Gallego and Davide Scaramuzza, "Accurate angular velocity estimation with an event camera," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 632–639, 2017.

[184] Timo Stoffregen, Guillermo Gallego, Tom Drummond, Lindsay Kleeman, and Davide Scaramuzza, "Event-based motion segmentation by motion compensation," in *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[185] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza, "Asynchronous, photometric feature tracking using events and frames," in *Proc. of European Conference on Computer Vision (ECCV)*, 2018.

[186] Daqi Liu, Álvaro Parra, and Tat-Jun Chin, "Globally optimal contrast maximisation for event-based motion estimation," *arXiv preprint arXiv:2002.10686*, 2020.

[187] Anton Mitrokhin, Cornelia Fermüller, Chethan Parameshwara, and Yiannis Aloimonos, "Event-based moving object detection and tracking," in *Proc. of International Conference on Intelligent Robots and Systems (IROS)*. IEEE/RSJ, 2018, pp. 1–9.

[188] Jie Xu, Meng Jiang, Lei Yu, Wen Yang, and Wenwei Wang, "Robust motion compensation for event cameras with smooth constraint," *IEEE Transactions on Computational Imaging*, vol. 6, pp. 604–614, 2020.

[189] Anton Mitrokhin, Zhiyuan Hua, Cornelia Fermuller, and Yiannis Aloimonos, "Learning visual motion segmentation using event surfaces," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 14414–14423.

[190] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis, "Ev-flownet: Self-supervised optical flow estimation for event-based cameras," in *Proc. of Robotics: Science and Systems*, 2018.

[191] Liyuan Pan, Miaomiao Liu, and Richard Hartley, "Single image optical flow estimation with an event camera," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1672–1681.

[192] Zhe Jiang, Yu Zhang, Dongqing Zou, Jimmy Ren, Jiancheng Lv, and Yebin Liu, "Learning event-based motion deblurring," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3320–3329.

[193] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza, "Eklt: Asynchronous photometric feature tracking using events and frames," *International Journal of Computer Vision*, pp. 1–18, 2019.

[194] Srutarshi Banerjee, Zihao W Wang, Henry H Chopp, Oliver Cossairt, and Aggelos Katsaggelos, "Quadtree driven lossy event compression," *arXiv preprint arXiv:2005.00974*, 2020.

[195] Kaiming He, Jian Sun, and Xiaoou Tang, "Guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, 2012.

[196] Xiaojie Guo, Yu Li, Jiayi Ma, and Haibin Ling, "Mutually guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[197] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang, "Deep joint image filtering," in *Proc. of European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 154–169.

[198] Pingfan Song, Xin Deng, João FC Mota, Nikos Deligiannis, Pier-Luigi Dragotti, and Miguel Rodrigues, "Multimodal image super-resolution via joint sparse representations induced by coupled dictionaries," *IEEE Transactions on Computational Imaging*, 2019.

[199] Zhengguo Li, Jinghong Zheng, Zijian Zhu, Wei Yao, and Shiqian Wu, "Weighted guided image filtering," *TIP*, vol. 24, no. 1, pp. 120–129, 2014.

[200] Xiaoyong Shen, Chao Zhou, Li Xu, and Jiaya Jia, "Mutual-structure for joint filtering," in *Proc. of International Conference on Computer Vision (ICCV)*, 2015.

[201] Xin Deng and Pier Luigi Dragotti, "Deep convolutional neural network for multi-modal image restoration and fusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[202] Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik Learned-Miller, and Jan Kautz, "Pixel-adaptive convolutional neural networks," in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11166–11175.

[203] Qiong Yan, Xiaoyong Shen, Li Xu, Shaojie Zhuo, Xiaopeng Zhang, Liang Shen, and Jiaya Jia, "Cross-field joint image restoration via scale map," in *Proc. of International Conference on Computer Vision (ICCV)*, 2013, pp. 1537–1544.

[204] Jaesik Park, Hyeongwoo Kim, Yu-Wing Tai, Michael S Brown, and Inso Kweon, "High quality depth map upsampling for 3D-ToF cameras," in *Proc. of International Conference on Computer Vision (ICCV)*, 2011, pp. 1623–1630.

[205] Jiahui Qu, Yunsong Li, and Wenqian Dong, "Hyperspectral pansharpening with guided filter," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 11, pp. 2152–2156, 2017.

[206] Zihao Wang, Peiqi Duan, Oliver Cossairt, Aggelos Katsaggelos, Tiejun Huang, and Boxin Shi, "Joint filtering of intensity images and neuromorphic events for high-resolution noise-robust imaging," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[207] H. S. Malvar, Li-wei He, and R. Cutler, "High-quality linear interpolation for demosaicing of bayer-patterned color images," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004, vol. 3.

[208] Fengqiang Li, Huaijin Chen, Adithya Pediredla, Chiakai Yeh, Kuan He, Ashok Veeraraghavan, and Oliver Cossairt, "Cs-tof: High-resolution compressive time-of-flight imaging," *Optics express*, vol. 25, no. 25, pp. 31096–31110, 2017.

[209] Fengqiang Li, Joshua Yablon, Andreas Velten, Mohit Gupta, and Oliver Cossairt, "High-depth-resolution range imaging with multiple-wavelength superheterodyne interferometry using 1550-nm lasers," *Applied optics*, vol. 56, no. 31, pp. H51–H56, 2017.

[210] Fengqiang Li, Florian Willomitzer, Prasanna Rangarajan, Mohit Gupta, Andreas Velten, and Oliver Cossairt, "Sh-tof: Micro resolution time-of-flight imaging with superheterodyne interferometry," in *2018 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2018, pp. 1–10.

[211] Chia-Kai Yeh, Nathan Matsuda, Xiang Huang, Fengqiang Li, Marc Walton, and Oliver Cossairt, "A streamlined photometric stereo framework for cultural heritage," in *European Conference on Computer Vision*. Springer, 2016, pp. 738–752.

[212] Florian Willomitzer, Fengqiang Li, Muralidhar Madabhushi Balaji, Prasanna Rangarajan, and Oliver Cossairt, "High resolution non-line-of-sight imaging with superheterodyne remote digital holography," in *Computational Optical Sensing and Imaging*. Optical Society of America, 2019, pp. CM2A–2.

[213] Chia-Kai Yeh, Fengqiang Li, Gianluca Pastorelli, Marc Walton, Aggelos K Katsaggelos, and Oliver Cossairt, "Shape-from-shifting: Uncalibrated photometric stereo with a mobile device," in *2017 IEEE 13th International Conference on e-Science (e-Science)*. IEEE, 2017, pp. 551–558.

[214] Houde Wu, Ming Zhao, Fengqiang Li, Zhiming Tian, and Meijing Zhao, "Underwater polarization-based single pixel imaging," *Journal of the Society for Information Display*, vol. 28, no. 2, pp. 157–163, 2020.

[215] Fengqiang Li, Florian Willomitzer, Prasanna Rangarajan, and Oliver Cossairt, "Megapixel time-of-flight imager with ghz modulation frequencies," in *Computational Optical Sensing and Imaging*. Optical Society of America, 2019, pp. CTh2A–2.

[216] Florian Willomitzer, Fengqiang Li, Prasanna Rangarajan, and Oliver Cossairt, "Non-line-of-sight imaging using superheterodyne interferometry," in *Computational Optical Sensing and Imaging*. Optical Society of America, 2018, pp. CM2E–1.

[217] Fengqiang Li, Huaijin Chen, Chia-Kai Yeh, Adithya Pediredla, Kuan He, Ashok Veeraghvan, and Oliver Cossairt, "Compressive time-of-flight imaging," in *Applied Industrial Optics: Spectroscopy, Imaging and Metrology*. Optical Society of America, 2018, pp. AM2A–5.

[218] Fengqiang Li, Nathan Matsuda, Marc Walton, and Oliver Cossairt, "Fluorescence lifetime estimation using a dynamic vision sensor," in *Computational Imaging II*. International Society for Optics and Photonics, 2017, vol. 10222, p. 102220N.

[219] Yicheng Wu, Fengqiang Li, Florian Willomitzer, Ashok Veeraraghavan, and Oliver Cossairt, "Wished: Wavefront imaging sensor with high resolution and depth ranging," in *2020 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2020, pp. 1–10.

List of Publications

[1] Z. Wang, L. Spinoulas, K. He, L. Tian, O. Cossairt, A. K. Katsaggelos, and H. Chen, "Compressive holographic video," in *Optics Express* 25, no. 1 (2017): 250-262.

[2] D. Ryu, Z. Wang, K. He, G. Zheng, R. Horstmeyer, and O. Cossairt, "Subsampled phase retrieval for temporal resolution enhancement in lensless on-chip holographic video," in *Biomedical Optics Express* 8, no. 3 (2017): 1981-1995.

[3] Z. W. Wang, V. Vineet, F. Pittaluga, S. N. Sinha, O. Cossairt, and S. B. Kang, "Privacy-preserving action recognition using coded aperture videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0-0. 2019.

[4] K. He, Z. Wang, X. Huang, X. Wang, S. Yoo, P. Ruiz, I. Gdor et al, "Computational multifocal microscopy," in *Biomedical Optics Express* 9, no. 12 (2018): 6477-6496.

[5] Z. Wang, Q. Dai, D. Ryu, K. He, R. Horstmeyer, A. K. Katsaggelos, and O. Cossairt, "Dictionary-based phase retrieval for space-time super resolution using lens-free on-chip holographic video." in *Computational Optical Sensing and Imaging*, pp. CTu2B-3. Optical Society of America, 2017.

[6] Z. W. Wang, and M. R. Luo, "Looking into special surface effects: diffuse coarseness and glint impression," in *Coloration Technology* 132, no. 2 (2016): 153-161.

[7] Z. W. Wang, W. Jiang, K. He, B. Shi, A. Katsaggelos, and O. Cossairt. "Event-driven video frame synthesis," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 0-0. 2019.

[8] Z. Wang, L. Xu, Y. Hu, F. Mirjalili, and M. R. Luo, "Gloss evaluation from soft and hard metrologies," in *Journal of the Optical Society of America A* 34, no. 9 (2017): 1679-1686.

[9] Z. W. Wang, P. Duan, O. Cossairt, A. Katsaggelos, T. Huang, and B. Shi, "Joint filtering of intensity images and neuromorphic events for high-resolution noise-robust imaging." in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1609-1619. 2020.

[10] Z. Wang, D. Ryu, K. He, A. K. Katsaggelos, and O. Cossairt, "4d tracking of biological samples using lens-free on-chip in-line holography," in *Digital Holography and Three-Dimensional Imaging*, pp. Tu2A-4. Optical Society of America, 2017.

[11] S. Banerjee, Z. W. Wang, H. H. Chopp, O. Cossairt, and A. Katsaggelos. "Quadtree Driven Lossy Event Compression," in *arXiv preprint* arXiv:2005.00974 (2020).

[12] K. He, X. Wang, Z. W. Wang, H. Yi, N. F. Scherer, A. K. Katsaggelos, and O. Cossairt, "Snapshot multifocal light field microscopy," in *Optics Express* 28, no. 8 (2020): 12108-12120.