DEPARTMENT OF ELECTRICAL
ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY
MADRAS
CHENNAI - 600036

Reconstructing High Temporal and Angular Resolution Videos from Low Data Bandwidth Measurements

A Thesis

Submitted by

PRASAN A SHEDLIGERI

For the award of the degree

Of

DOCTOR OF PHILOSOPHY

May 2022

QUOTATIONS

Somewhere. Somehow.
There's a story that
Wants to be found.

A poem that wants
To fall suitably,
Into your words.

A painting,
Awaiting to fit
In your shades.

For a dream
That wants to be
Realized -

No matter,
How stupid; how boring.
In all your subtleties-

You; in yourself are
A piece of work.
Yet, unveiled, unfolded.

DEDICATION

To my beloved friends and family

THESIS CERTIFICATE

This is to undertake that the Thesis titled **RECONSTRUCTING HIGH TEMPORAL AND ANGULAR RESOLUTION VIDEOS FROM LOW DATA BANDWIDTH MEASUREMENTS**, submitted by me to the Indian Institute of Technology Madras, for the award of Ph.D., is a bona fide record of the research work done by me under the supervision of Dr. Kaushik Mitra. The contents of this Thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Place: Chennai 600 036

Date: 19th May 2022

Prasan A Shedligeri
Research Scholar

Dr. Kaushik Mitra
Research Guide

LIST OF PUBLICATIONS

I. REFEREED JOURNALS BASED ON THE THESIS

1. Prasan Shedligeri and Kaushik Mitra, "High Frame Rate Optical Flow Estimation from Event Sensors via Intensity Estimation" *Elsevier Journal of Computer Vision and Image Understanding (CVIU)*, 208-209, 103208, 2021.
2. Prasan Shedligeri, Anupama S and Kaushik Mitra, "CodedRecon: Video reconstruction for coded exposure imaging techniques" *Elsevier Journal of Software Impacts (SIMPAC)*, 8, 100064, 2021.
3. Prasan Shedligeri and Kaushik Mitra, "Photorealistic image reconstruction from hybrid intensity and event-based sensor", *SPIE Journal of Electronic Imaging (JEI)*, 28(6), 063012, 2019.

II. PUBLICATIONS IN CONFERENCE PROCEEDINGS

1. Prasan Shedligeri, Florian Schiffers, Sushobhan Ghosh, Oliver Cossairt and Kaushik Mitra, "SeLFVi: Self-supervised Light Field Video Reconstruction from Stereo Video", *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2491–2501, 2021.
2. Prasan Shedligeri, Anupama S and Kaushik Mitra "A Unified Framework for Compressive Video Recovery from Coded Exposure Techniques", *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 1600–1609, 2021.
3. Anupama S, Prasan Shedligeri, Abhishek Pal and Kaushik Mitra, "Video Reconstruction by Spatio-Temporal Fusion of Blurred-Coded Image Pair", *IEEE International Conference on Pattern Recognition (ICPR)*, 7953–7960, 2021.
4. Prasan Shedligeri, Sreyas Mohan and Kaushik Mitra, "Data driven coded aperture design for depth recovery", *IEEE International Conference on Image Processing (ICIP)*, 56–60, 2017.

ACKNOWLEDGEMENTS

Stand on the shoulder of the giants

This thesis was made possible because of the strong support and foundation laid down by Dr. Kaushik Mitra. And many others helped me cross the finish line too. I am very grateful and appreciative of everyone who believed in me, tirelessly encouraged me towards the goal, and helped directly or indirectly in finally achieving it. I am thankful to all my Doctoral Committee members: Prof. Arvind, Prof. A. N. Rajagopalan, Prof. Ravindran and Prof. Umesh for their continuous support. I am also thankful to all the anonymous reviewers who selflessly reviewed my papers and journals and helped improve my work and to make a valuable contribution to the field.

I feel immensely lucky to have had Dr. Kaushik Mitra as my PhD advisor. He has always been very encouraging and supportive. He would patiently listen to my wildest ideas and then slowly prune and direct me towards more promising directions. I am also grateful for his support through the tough times of a PhD life. With him, I learned to give my best to a project and then not attach myself too much to the successes and failures. This helped me maintain a consistent and stable motivation for my work. I also feel grateful that he encouraged and helped me apply for several scholarships and fellowships. This opened me up to several new experiences and learning opportunities.

As a result of one such scholarship, I got a chance to work with Prof. Oliver Cossairt and Prof. Aggelos Katsaggelos at Northwestern University. I gratefully cherish the entire time I was at Northwestern, working on several interesting projects. I was very warmly welcomed into the lab by Ollie, Florian Schiffers, Florian Willomitzer, and Fabian Wagner. Each one of them helped me get used to the new environment and supported me in all the projects that I wanted to do. I am also grateful for the support I received from Winston, Fengqiang, Lionel, Semih, Henry, Pablo, Manuel, Hamid, Weixin, Sushobhan, Nathan, and Srutarshi during the time I spent in Evanston. My visit to the Northwestern University was initially made possible by a travel scholarship from

Robert Bosch Center for Data Science and Artificial Intelligence. So, I am extremely thankful to Mr. Sridhar and Prof. Balaraman Ravindran who made this possible.

Another such great learning opportunity for me was my internship at Samsung Research India, Bangalore. I feel very thankful towards Pawan Prasad and Dr. Loki who pulled all the strings to give me a shot at the internship interview. After I got the internship and entered the office, I was warmly welcomed by Dr. Rituparna Sarkar, Dr. Venkat and Green Rosh. Dr. Loki and Dr. Ritu gave me challenging problems and a very supportive environment to work during my time at the Samsung Research office. Thanks to the efforts of all these people, I got an exciting first-hand experience at working in industrial research lab.

As a PhD candidate, a significant part of my day was spent inside the lab. And ESB 221 was a very lively busy lab, always bustling with graduate and under-graduate students. I remember that sometimes all the 15 odd chairs would be occupied and even I did not have space to sit and work in the lab. And I will very much miss that lively and informal atmosphere inside the lab. The discussions about work, food, and life that I had with my labmates were some of the very best moments of my life. I am lucky to have had colleagues like Sreyas, Anil, Akshat, Sharath, Renju, Honey, Gopi, Mohit, Salman, Ketul, Dhruv, Anupama, Meenakshi, Sneha, Subeesh, Kuldeep, Mahesh, Nimisha, Kranthi, Praveen, Atreyee, Bimal, Ramana, Anuja, and Praful. Some of these colleagues keep growing closer and still show the same support and encouragement as they did during my PhD. There were simply too many colleagues to name everyone here, but I nevertheless feel very appreciative towards every one of them.

I am also grateful to my school and undergraduate friends. They have stuck with me through thick and thin and I cannot appreciate enough the encouragement I received from them. Sometimes they believed in me more than I believed in myself. So, I say thank you to Nikhil, Wasim, Ajay, Mr. Marisamy, Sanjeev, Prithviraj, Manjunath, Sharath, Shridhar, Sourabh, Nemi, Kumar, Aju, Jyoti, Pradeep, Mahantesh, Chetan, and Ankit. I would also like to thank my colleagues at my first job after college who showed immense support for all my endeavors. I particularly thank Deepak, Praveen, Muthu, Bala, Vijay, Shashidhar, Rupesh and Ranjita for their support. I cannot understate the support I received from my parents and my brother. They never questioned my decisions and always encouraged me to do my best.

Finally, I would like to acknowledge and remember all the people who worked behind the scenes. Their work ensured that I did not have to worry about anything other than research. These are my hostel managers, housekeeping staff at the office and hostel, dining hall employees, administrative employees at IIT Madras, student representatives, food delivery drivers, cab drivers and so many others. While they were all probably just “doing their duty”, their duty was critically important for me to keep my focus on research and not anything else.

Thank you for making this happen!

ABSTRACT

KEYWORDS: High-speed imaging; Light-field video; Deep learning; Event sensors; Stereo camera; Coded-exposure sensors; Coded-2-bucket sensor; Optical flow

The complete visual signal is described by a 7-dimensional function, known as the *plenoptic function*. The plenoptic function can be formally defined as the *radiance received along any direction (characterized by two angles θ and ϕ) arriving at any point in space (three Cartesian coordinates (x, y, z)), at any time (t) and over any range of wavelength (λ)*. Current camera hardware densely samples only the 2 spatial dimensions (x, y), while either ignoring or sparsely sampling the other dimensions of the plenoptic function. Specialized hardware is necessary for dense sampling of either spectral, temporal or angular dimensions. E.g. light-field cameras acquire dense angular information using micro-lens arrays stacked near the image sensor. Also, dense sampling in multiple dimensions simultaneously, leads to more data being generated at the sensor level. Hence, these specialized cameras should be equipped with hardware and software capable of processing and storing this large bandwidth of data in real-time. As this increases the cost of the camera, it is imperative to sample a compressed measurement, that requires low data bandwidth at the sensor level, from the original high bandwidth signal. The ill-posed nature of the original signal reconstruction from corresponding low data bandwidth measurements requires us to use complex and advanced signal processing techniques. In this thesis, we discuss acquisition of two signals requiring high data bandwidth at the sensor level, namely the high-speed video and the light field (LF) video, through intelligent sampling and computational reconstruction.

High-speed video is a 3 dimensional signal where the plenoptic function is sampled densely in both spatial and temporal dimensions. Typically, a video acquired at a frame-rate of more than 250 frames per second (fps) can be considered as high-speed video. Most commercial cameras are limited to videos at 30 fps at a maximum spatial resolution of 8 mega-pixels (MP). Acquiring a 8 MP resolution video at, say, 500 fps is equivalent to a 30 fps video at ~130 MP spatial resolution. This is a very large data bandwidth for any modern commercial camera to handle. In this thesis, we explore two different techniques to sample high-speed videos as low data bandwidth signals. The first technique, known as coded-exposure imaging, temporally multiplexes several high-speed frames into a single frame of a low frame-rate video. This multiplexed low frame-rate video serves as our intelligently sampled low data bandwidth signal for high-speed video reconstruction. The second technique is based on a novel neuromorphic event-based sensor. The event-based sensor acquires data equivalent to a temporal difference between successive high-speed video frames at about million frames per second. These sparse temporal differences are encoded as binary events from which we aim to reconstruct the high frame-rate videos.

Coded-exposure imaging temporally compresses multiple high-speed video frames into a single frame. Recovering the video from these measurements is an ill-posed problem that requires strong spatio-temporal signal prior to be imposed. Prior approaches have explored both analytic and data-driven signal priors, with superior results being obtained with data-driven approaches. We propose a convolutional neural network based technique that reconstructs the full-resolution video in a single forward pass. This is unlike several previous approaches using dictionary learning and fully-connected networks that reconstruct individual patches independently before merging them as videos. We demonstrate state-of-the-art reconstructions for videos up to 480 fps with the coded-exposure sensor operating at only 30 fps.

Event-based sensors acquire only the brightness differences between successive frames as binary events. The events are acquired at microsecond temporal resolution leading to a theoretical frame-rate of a million frames per second. These events are also sensitive to a very large dynamic range of about 120 decibels (dB). However, the binary events are a quantized form of the actual brightness changes and are hence affected by noise. Relying only on the event-sensor data for reconstruction leads to artifacts such as trailing-edges and sensor-noise related degradation. Hence, we propose to utilize additional intensity images to compensate for the lost spatial texture information during high frame-rate video reconstruction. The intensity sensor operates at a very low frame-

rate of $20-30$ fps, while the events are still acquired at a million frames per second. By relying on the event sensor data for only dense camera-motion estimation, we demonstrate high-quality, artifact-free high-speed video reconstruction. By relying more on the intensity images, the dynamic range of the output video is much lower than the event sensor dynamic range of 120 dB. Hence we wish to further exploit event-sensor data to reconstruct high dynamic range videos. We propose a semi-supervised learning based technique to reconstruct high dynamic range and high-speed videos from event sensors. This approach does not rely on intensity image input and instead utilizes learning-based technique to infer the texture information from event sensor alone. We qualitatively demonstrate superior results and generalization ability to novel event sensor data than previous learning-based approaches. With event-based sensors we achieve frame-rate upsampling of up to $60\times$, leading to reconstructed videos at frame-rate greater than 1000 fps.

LF video is another high data bandwidth video that is challenging to acquire. For a reasonable angular resolution of 7×7, we require ~$50\times$ the bandwidth of the standard monocular video. If each of the LF angular views have a spatial resolution of 1 MP, then it's equivalent to acquiring a 50 MP video from the camera. As such high data bandwidth videos are not handled by current image sensors, it becomes necessary to reconstruct these videos from a low data bandwidth signal. We propose to use a stereo video as our low data bandwidth signal, as it can be thought of as a sparse sample of the LF angular views. Unlike LF videos, commercial image sensors can easily acquire stereo videos as they require only $2\times$ the bandwidth of a monocular video. Supervised learning-based methods cannot be used to solve this ill-posed problem due to a lack of high-quality training data of LF videos. We propose a self-supervised learning based scheme for LF *video* reconstruction that only requires easy-to-acquire stereo videos. We achieve an angular super-resolution of about ~$40\times$ reconstructing 9×9 LF videos from just stereo videos. We achieve an angular super-resolution of about ~$40\times$ reconstructing 9×9 LF videos from just stereo videos.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

fps	frames per second
MP	mega-pixels
LF	light field
6-DoF	6 degrees of freedom
dB	decibels
DAVIS	Dynamic and Active Pixel Vision Sensor
DVS	Dynamic Vision Sensor
C2B	Coded-2-Bucket
DSLR	Digital Single Lens Reflex
SVC	shift-variant convolutional
FS	flutter shutter
AR	augmented reality
VR	virtual reality
SLAM	simultaneous localization and mapping
PSNR	peak signal to noise ratio
LSTM	Long Short-Term Memory
ConvLSTM	Convolutional Long Short-Term Memory
ATIS	Asynchronous Time-based Image Sensor
LiDAR	Light Detection and Ranging
EPI	epipolar plane image
SAI	sub-aperture image
TV	total-variation
ReLU	Rectified Linear Unit
SBN	stacking by number
SBT	stacking by time

MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
LPIPS	Learned Perceptual Image Patch Similarity

NOTATION

English Symbols

3**D**	three dimensional
S	Video Signal
$\mathbf{S}$	Vectorized video signal S
$\hat{S}$	Estimated video signal
s_t	t^{th} frame of the video S
s_{t-1}	$t-1^{th}$ frame of the video S
s_{t+1}	$t+1^{th}$ frame of the video S
$\hat{s_t}$	Predicted t^{th} frame of the video S
d_t	Depth map for t^{th} frame of the video S
d_{t+1}	Depth map for t^{th} frame of the video S
x	Horizontal spatial co-ordinate
y	Vertical spatial co-ordinate
u	Horizontal angular co-ordinate
v	Vertical angular co-ordinate
H	Height of an image or video
W	Width of an image or video
U	Number of horizontal angular views
V	Number of vertical angular views
T	Number of temporal frames
$\mathbf{u}_l$	SAI co-ordinate for left-view of the light-field
$\mathbf{u}_r$	dfds
$\mathbf{L}_t$	t^{th} light-field frame
$\hat{\mathbf{L}}_t$	Estimated t^{th} light-field frame

$\mathbf{L}_{t-1}$	t^{th} light-field frame
$\hat{\mathbf{L}}_t$	t^{th} estimated light-field frame
Y	Compressed coded measurement
$\mathbf{Y}$	Lexicographically ordered compressed measurement
K	Camera intrinsic matrix
E_t	Estimated pseudo-intensity frames
E_{t+1}	Estimated pseudo-intensity frames
b	Binary mask

Greek Symbols

∇	Finite difference gradient operator
∇_x	Finite difference horizontal gradient operator
∇_y	Finite difference vertical gradient operator
Φ	Coded exposure sequence
$\mathbf{\Phi}$	Matrix form of the coded exposure sequence
ϕ_t	Coded exposure sequence for t^{th} frame
α	Parameter for alpha blending
ξ_t^j	Relative pose between j^{th} and t^{th} frames
ξ_{t+1}^j	Relative pose between j^{th} and $t+1^{th}$ frames

CHAPTER 1

Introduction

The complete visual signal in nature can be characterized by a 7-dimensional function known as the *plenoptic function* (Bergen and Adelson, 1991). The plenoptic function can be formally defined as the *radiance received along any direction V arriving at any point E in space, at any time 't' and over any range of wavelength λ*. While the spatial location E requires 3 spatial/Cartesian coordinates, describing the direction V requires 2 angular coordinates. The temporal coordinate t, and the spectral coordinate λ, form the last two components of the 7D plenoptic function. However, commercial camera hardware is limited for capturing only a 2D spatial projection of this complete 7D function. Color-imaging and video acquisition only sparsely sample the spectral and temporal dimensions of the plenoptic function respectively. We require specialized hardware for dense sampling in spectral, angular, or temporal dimensions. E.g. LF cameras acquire dense angular information using a micro-lens array placed near the image sensor. Also, sampling more dimensions means more data is being captured at the sensor level. It becomes challenging to process and store this large amount of data in real-time. Hence, these cameras also require hardware that are capable of handling a large bandwidth of data in real-time. As this increases the camera's cost, it is favorable to acquire only low data bandwidth signals at the sensor-level. However, this low data bandwidth signal needs to be intelligently sampled from the required high data bandwidth signal, so that it enables high fidelity recovery of the original signal (in this thesis, bandwidth is used in the context of computing, referring to the rate of data transfer). While the sampling is performed at the sensor-level, complex and advanced signal-processing techniques are used to recover the original signal.

This thesis explores high-speed (or high frame-rate) video and LF video reconstruction from their corresponding low data bandwidth samples. High frame-rate videos are useful in entertainment, scientific and industrial applications. In scientific applications, high frame-rate cameras are used to characterize events that happen too fast for a traditional camera to capture it fully. E.g. aerodynamic motion of hummingbirds, bees, arrows, bullets *etc.*, shock-wave after a nuclear explosion and many other significant phenomena. High frame-rate videos have seen widespread use in entertainment industry as well. They have become especially popular with their introduction in modern premium smartphones. Even viewing regular human activities like jumping in the pool, sliding on a wet surface, etc. in slow-motion make for some amusing and entertaining videos. However, smartphones are limited to acquiring these videos for only a fraction of a second because of their limited power and memory. LF imaging has also become popular as it enables intuitive and simple post-capture focus control. This lets an user to choose where to focus the image and how much depth-of-field to have after an image has been captured. LF video acquisition is especially important for fast moving events, where the focus has to quickly shift from one depth plane to another. If we have LF videos, then the video editor can easily control where to focus the image for each frame. AS high-speed videos and LF videos have several important applications, it is important to have practical approaches for acquiring these videos. This thesis proposes frameworks for acquisition of both high-speed videos and LF videos.

The overall design of each framework considered here operate on the following principles. Initially, a hardware setup capable of encoding the high data bandwidth video into low data bandwidth measurements is chosen or designed. The next step is to decode the high data bandwidth video from the low data bandwidth measurements. The decoding employs a signal processing framework tailored to the specific application. In this thesis, learning-based techniques form a major component of the proposed decoding frameworks. The following chapters of this thesis discuss specific hardware setups and tailored signal processing techniques to recover high data bandwidth videos.

1.1 Motivation, objectives and scope

This thesis explores the reconstruction of high temporal resolution videos and high angular resolution videos from low data bandwidth measurements. A high-speed video is a 3 dimensional signal where the plenoptic function is sampled densely in both the spatial (x,y) and temporal (t) dimensions. Typically, a video acquired at a frame-rate of more than 250 fps is considered as a high-speed video. Most commercial cameras are limited to acquiring videos at 30 fps with a maximum spatial resolution of 8 MP. Acquiring a 8 MP resolution video at, say, 500 fps is equivalent to capturing a 30 fps video at ~130 MP spatial resolution. This is a very large bandwidth for any modern commercial camera to handle.

In this thesis, we propose two different systems to sample the high-speed videos into corresponding low data bandwidth measurements. The first system consists of a coded-exposure sensor where the shutter of the sensor is modulated at a much higher frequency than the original sensor frame-rate. This process temporally multiplexes multiple frames of the high-speed video into a single compressed measurement as a video frame. Recovering the high-speed video from this compressed measurement is an ill-posed problem and several approaches have been proposed to tackle the same. In Sec. 1.1.1, we briefly elaborate on the shortcomings of previous approaches and how our proposed approach overcomes those challenges to recover high quality video reconstructions.

Another system for high-speed imaging explored in this thesis is based on novel event-based sensors. These sensors sense only the pixel-level, temporal brightness changes as binary events at a very high temporal resolution. Hence, they promise reconstruction of videos at thousands of frames per second while requiring much lower data bandwidth than a frame-based sensor. With this novel sensor, we propose two different methods to reconstruct high-speed videos from event-based sensors. In the first method, we utilize an additional intensity sensor information to obtain artifact-free reconstruction (Sec. 1.1.2). The second method overcomes the low dynamic range limitation of the first one by exploiting the state-of-the-art learning-based techniques (Sec 1.1.3).

LF video is another high data bandwidth video that is challenging to acquire. For a reasonable angular resolution of 7×7 we require ~$50\times$ the bandwidth of the standard monocular video. If each of the LF angular views have a spatial resolution of 1 MP, then it's equivalent to acquiring a 50 MP video from the camera. As such high data bandwidth videos are not handled by current image sensors, it becomes necessary to reconstruct these videos from a low data bandwidth signal. Here, we propose to use a stereo video as our low data bandwidth signal, as it can be thought of as a sparse measurement of the required LF video. Unlike LF, stereo videos only require ~2× the bandwidth of the normal monocular video. The bandwidth required for a stereo video can be easily handled by the commercial image sensors. Several standalone commercial cameras exist that can acquire stereo videos at a very high spatial resolutions. Even some modern premium smartphones have started supporting acquisition of dual-lens videos. This motivates us to tackle the challenging problem of reconstruction of LF video from the corresponding stereo video sequence (Sec. 1.1.4).

1.1.1 High-speed video reconstruction with coded-exposure sensors

Coded-exposure imaging has been a popular computational photography technique for a variety of applications such as motion deblurring and high frame-rate video reconstruction (Raskar *et al.*, 2006; Reddy *et al.*, 2011; Holloway *et al.*, 2012; Llull *et al.*, 2013; Liu *et al.*, 2013; Yoshida *et al.*, 2018; Iliadis *et al.*, 2020; Martel *et al.*, 2020; Li *et al.*, 2020; Anupama *et al.*, 2021). Deep-learning based techniques have greatly improved the video reconstruction quality with their ability to model data-driven priors (Yoshida *et al.*, 2018; Iliadis *et al.*, 2020; Martel *et al.*, 2020; Li *et al.*, 2020; Anupama *et al.*, 2021). Prior deep-learning based approaches use fully-connected networks and recover the video one patch at a time (Yoshida *et al.*, 2018; Iliadis *et al.*, 2020). However, fully-connected networks have fallen out of favor for image and video processing tasks in deep-learning. Fully connected networks tend to have large number of parameters, requiring a large amount of training data. Video recovery is also very slow as the network has to be run for each patch of the input measurement separately. In our work, we demonstrate that locally-connected, fully convolutional networks are better suited for this task. Better results can be obtained when the convolutional layer supports spatially varying filters or weights. This is unlike the standard convolutional layer where all spatial locations share identical convolutional filters. The convolutional layer with spatially varying filters is implemented as the SVC layer and forms the first layer of our proposed network. The features extracted from this layer is input to a deep neural network to extract the full-resolution video sequence. With minimal changes, our proposed network can be trained for different coded-exposure sensors such as flutter shutter (Holloway *et al.*, 2012) and P2C2 (Reddy *et al.*, 2011).

Techniques such as flutter shutter and P2C2 acquire a single compressed measurement in each exposure. However, a recently introduced C2B sensor (Sarhangnejad *et al.*, 2019) is capable of acquiring *two* compressed measurements per exposure. We adapt our proposed technique to exploit information from both the compressed measurements. The adaptation is done via modifying only the first layer of our neural network, namely the SVC layer. Our proposed technique is the first algorithm to exploit the two compressed measurements from C2B to obtain high fidelity video reconstructions. This also makes our network a unified framework to reconstruct video from the three different coded-exposure techniques. Extensive comparison shows that the proposed network achieves state-of-the-art reconstructions on all three coded-exposure techniques. With our unified technique, we make an extensive quantitative comparison for video reconstruction from the three coded-exposure techniques. And, as expected, C2B technique shows the highest video reconstruction performance, owing to the acquisition of two compressed measurements. Further comparison also shows the advantage of the two measurement being the highest for the case when scene is largely static. And only a marginal improvement in reconstruction quality over the single measurement case for a dynamic scene.

1.1.2 Photorealistic image reconstruction with event sensor

Event sensors operate with microsecond temporal resolution, and allow reconstruction of videos at thousands of frames per second. This was exploited in (Bardow *et al.*, 2016; Barua *et al.*, 2016; Munda *et al.*, 2018; Reinbacher *et al.*, 2016*b*) to reconstruct high-frame rate videos from the event sensor data. However, by relying solely on events, these techniques had the following drawbacks:

- Trailing edge artifacts due to integration of events to generate images.
- Some of the edges/objects in the scene can also go missing in the recovered frames because they are not producing any events (edges parallel to the sensor motion or objects that have zero relative motion with respect to the event sensor)
- Loss of absolute scene intensity information in encoding events leads to non-photorealistic reconstructions.
- Prone to failure due to excessive noisy events caused due by rapid object/camera motion and highly-textured scenes.

All these drawbacks could be overcome by exploiting the additional spatial texture information from a low frame-rate conventional image sensor. A hybrid sensor consisting of co-located intensity and event sensor was proposed by Brandli *et al.* (2014). Our work exploits the complementary nature of the two sensors to produce photorealistic high frame-rate reconstructions. While we extract the motion information from the event sensor, the image sensor is used to extract texture-rich spatial information. The algorithm uses the temporally dense event information to predict temporally dense camera motion in the form of 6-DoF relative camera pose. The motion information is then used to warp the image frames to the temporally dense locations of events. This results in a temporally dense sequence of photorealistic image frames from the hybrid sensor setup. Relying on the highly-noisy event sensor data only for 6-DoF pose estimation ensures that the algorithm is robust to cases of rapid motion where event sensor noise is the highest.

1.1.3 High dynamic range video reconstruction for event sensors

Our proposed photorealistic reconstruction pipeline does not fully exploit the high dynamic range nature of the event sensors. While event sensors possess a dynamic range of ~ 120 dB, the reconstructed photorealistic videos are limited by the dynamic range of the intensity sensor. By using the additional intensity sensor information along with event sensors, we were able to eliminate several drawbacks on previous techniques. However, the need for an additional intensity sensor to obtain high quality reconstructions was eliminated with the use of powerful deep-learning based techniques. Learning-based frameworks proposed in (Rebecq *et al.*, 2019*b*,*a*) showed promising results for high-dynamic range and high-frame rate video reconstruction from event sensor data alone. In (Rebecq *et al.*, 2019*b*,*a*), the neural networks were supervised using large corpus of synthetic data consisting of pairs of event and intensity frames. The event-sensor noise cannot be realistically simulated in a synthetic data for all scenarios. Hence, when the event sensor noise became dominant in a scene, these algorithms failed to produce good quality reconstructions.

To overcome the reliance on synthetic data, we propose a semi-supervised learning-based technique to reconstruct high-frame rate and high dynamic range videos from event sensors. The proposed technique is designed to use real event sensor data acquired from hybrid sensors such as DAVIS (Brandli *et al.*, 2014). Intensity frame prediction is supervised using the ground truth frames from the conventional image sensor. Unlike (Rebecq *et al.*, 2019*b*), our technique does not require ground truth optical flow to enforce temporal consistency between successive intensity frames. Our proposed technique directly predicts optical flow from the event sensor data. The predicted optical flow is utilized to enforce temporal consistency between successive predicted frames. Optical flow prediction is based on self-supervised learning with the help of predicted intensity frames (Meister *et al.*, 2018; Jason *et al.*, 2016; Ren *et al.*, 2017). We demonstrate generalization of our technique to event sensor input from various challenging scenarios and multiple sensors of varying spatial resolutions.

1.1.4 Light field video reconstruction from stereo video

LF imaging with high angular, spatial and temporal resolution is challenging due to complex hardware requirements and bandwidth constraints. Commercial LF cameras acquire LF videos at a mere 3 fps (Wang *et al.*, 2017), a frame rate much lower than a typical frame-rate of 30 fps. Wang *et al.* (2017) made one of the first attempts at reconstructing LF video sequences at 30 fps using a hybrid camera set up of a commercial LF camera and a Digital Single Lens Reflex (DSLR) camera. The proposed hardware setup is too bulky to be of any practical use. Then, Hajisharif *et al.* (2020) overcome the challenge of bulky setup using a compressive sensing technique based on a single image sensor and a coded attenuating mask. This mask efficiently encodes the angular information so that it enables better recovery of LF from the sensor measurements. While the hardware setup was now reduced to a single sensor, it required meticulous placement and calibration of this mask. Recently a learning-based technique was proposed for the recovery of LF video from a monocular video in Bae *et al.* (2021). This technique simplified the hardware requirement even further, requiring only a monocular camera for obtaining input. However, this technique relies on computer generated synthetic training data and hence not guaranteed to generalize well to unseen real-world test sequences.

In this thesis, we consider the case of reconstructing LF videos from stereo video sequences. While acquiring a full LF video requires a staggering $50\times$ the bandwidth of the normal monocular video, a stereo video just requires $2\times$ the bandwidth. The stereo video can be considered as a sparse sample of the full LF video that we wish to reconstruct. Stereo videos are low-bandwidth signals that can be acquired by modern commercially available devices such as smartphones and standalone cameras. We propose a self-supervised learning based framework for reconstruction of LF videos from stereo videos. To regularize the LF frame reconstruction, we utilize a multi-layer LF-display based representation (Wetzstein *et al.*, 2012) as a prior. LF video reconstruction is guided via the geometric and temporal information embedded in the input stereo videos. Various consistency costs enforce the epipolar geometric, photometric and temporal consistency on the reconstructed LF videos. The geometric consistency is enforced via disparity maps estimated from individual stereo frames. The temporal consistency on the LF video is enforced via optical flow obtained from the left and right video sequences of the stereo input. We demonstrate LF video reconstruction from stereo videos captured using commercially available stereoscopic cameras.

1.1.5 Contributions of the thesis

- ○ We propose a supervised learning-based framework for reconstruction of video sequences from coded-exposure techniques. Our proposed approach can take as input images from three different coded-exposure sensing schemes with minimal changes to the network architecture. While providing state-of-the-art results, the network also allows for a fair comparison between the various coded-exposure schemes. We show that C2B has significant advantage over per-pixel exposure coding in reconstructing videos of scenes consisting of mostly static regions.

- ○ We propose a pipeline for reconstructing high-frame rate photorealistic intensity images using a hybrid event and low-frame-rate intensity sensor. Using real data captured using a commercially available hybrid sensor called DAVIS, we show high-frame rate photorealistic intensity reconstructions. The proposed algorithm's robustness to abrupt camera motion and noisy event sensor data is also shown.

- ○ Then to preserve the high-dynamic range advantage of event sensors, we propose a semi-supervised learning based technique for simultaneous reconstruction of intensity and optical flow. Our proposed technique does not require ground truth optical flow but still reconstructs accurate sparse optical flow from just event sensor data. The proposed algorithm is also shown to generalize to a various event-sensor datasets captured using multiple types of sensors under different lighting conditions and motions.

- Finally, we propose a self-supervised learning based technique to reconstruct LF videos from stereo videos. The technique makes an effective use of a layered LF display based representation as a regularization for LF video prediction. We demonstrate post-training finetuning of the neural network on novel test sequences leading to improved results. We also showcase variable angular view prediction for both view interpolation and extrapolation.

1.2 Organization of the thesis

The rest of the thesis is organized as follows. Chapter 3 proposes a unified learning based framework for high frame-rate video reconstruction from coded-exposure sensors. In Chapters 4 and 5 we discuss frameworks for high-frame rate video reconstruction from event-based sensor. In Chapter 6, we propose a framework for reconstruction of high angular resolution LF videos from stereo videos. Finally, in Chapter 7, we conclude the thesis with some insights into future directions.

CHAPTER 2

Technical Background

In this chapter we present some of the technical background that is essential in understanding the research works presented in the subsequent chapters. We will review the workings of some of the novel image sensors used in this thesis. We begin by describing the coded-exposure image sensor and its mode of operation. Then the novel neuromorphic event-based sensors will be discussed. Finally, we briefly discuss the the concept of high angular resolution imaging as 4D LF images.

2.1 Coded-exposure sensors

In a conventional camera, an image is captured with the press of a button that opens the shutter and exposes the image sensor to the incoming light. The shutter remains open for a fixed duration, known as shutter speed, which is either specified by the user or computed automatically (known as auto-exposure). The sensor then collects all the incoming light falling onto it and outputs an intensity image. In case of a global shutter camera, all the pixels start and stop collecting the light simultaneously. In case of a coded-exposure sensor, the light can be blocked or allowed onto the sensor depending on an external binary code. This is in contrast to the conventional image sensor, where all the light is allowed onto the sensor during the exposure. In a coded-exposure sensor, an exposure duration of time T, is divided into multiple sub-exposures. Within each sub-exposure, the user can decide to either allow the light (binary code 1) or block the light (code 0) from falling onto the sensor. This process has the equivalent effect of temporally multiplexing multiple high-speed video frames into a single compressed measurement. The frame rate of the high-speed video is the inverse of the duration of each sub-exposure in the exposure sequence.

Based on whether the shutter modulation can be controlled at pixel-level or the sensor-level, coded-exposure sensors can be broadly categorized into two types:

- ○ Global coded-exposure sensor (see Fig. 2.1a)
- ○ Pixel-wise coded-exposure sensor (see Fig. 2.1b and 2.2)

Next, we elaborate on each of these categories of the coded-exposure sensor.

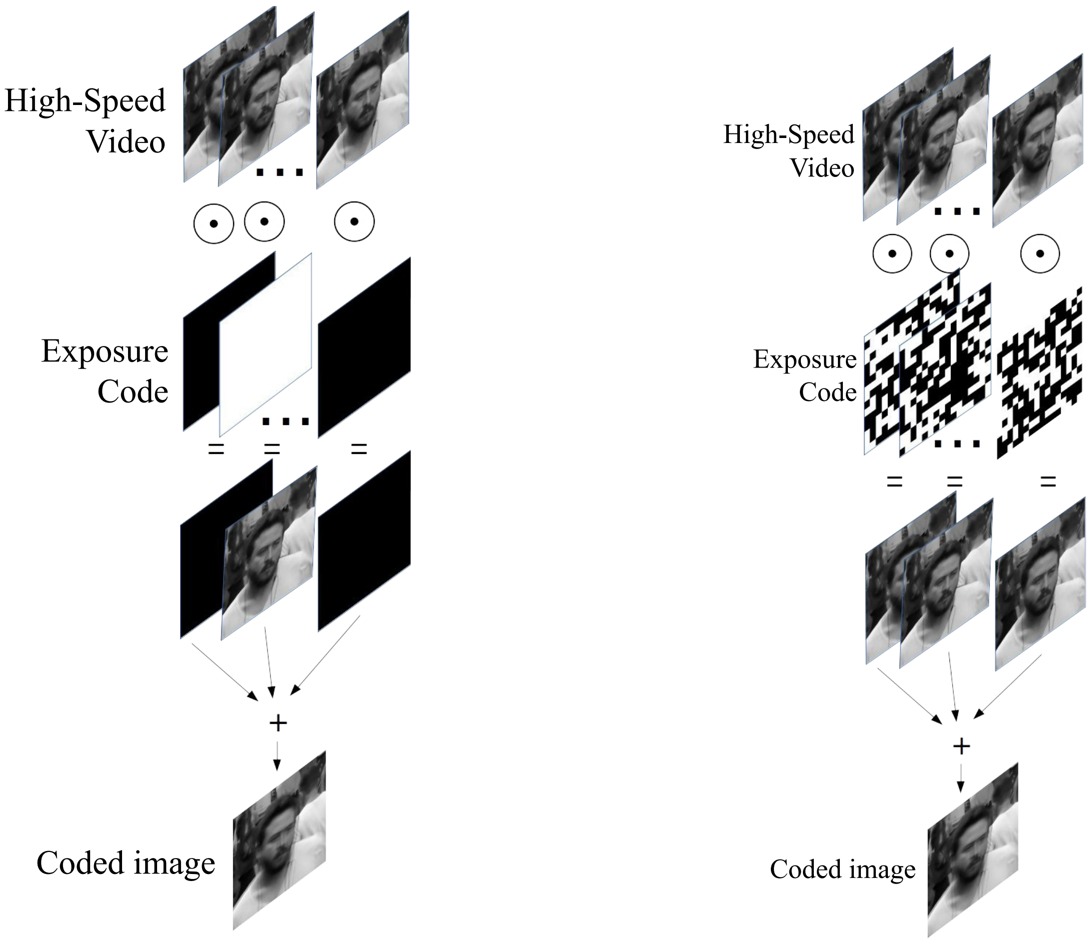
2.1.1 Global coded exposure sensor

As the name suggests, in a global coded-exposure sensor, each pixel's shutter is modulated with an identical binary exposure sequence. This technique was first introduced by Raskar *et al.* (2006) for image deblurring. Due to its simplicity, we could also use a mechanical shutter to implement such a global coded-exposure sensor. Through shutter modulation, a global coded-exposure sensor temporally multiplexes multiple high-speed video frames into a single measurement. We provide a visual and intuitive explanation of this process in Fig. 2.1a. We defer the mathematical formulation to Chapter 3 where it is more relevant.

2.1.2 Pixel-wise coded-exposure sensor

In this case, we are allowed to modulate the shutter of each pixel independent of the other. Hence, global coded-exposure sensor is a restricted case of the pixel-wise coded-exposure sensor. This technique was introduced in (Gu *et al.*, 2010; Reddy *et al.*, 2011) for high frame-rate video reconstruction. We show a visual and intuitive representation of its operation in Fig. 2.1b. Unlike global coded-exposure sensors, pixel-wise coded exposure sensors are harder to implement. Reddy *et al.* (2011) used a spatial light modulator along with relay lenses to externally modulate the images before they are incident on the camera sensor. Gu *et al.* (2010) instead modify the CMOS image sensor to acquire the coded measurements albeit with only a restricted set of modulation sequences.

Recently, a novel prototype sensor was introduced, named C2B (Sarhangnejad *et al.*, 2019), that significantly simplifies the process of coded-image acquisition. This sensor allows the user to simply program the coded-exposure sequence into the sensor and the coded-exposure measurement is acquired without any additional hardware. This sensor is built on the multi-bucket sensor technology (Sarhangnejad *et al.*, 2019) where each

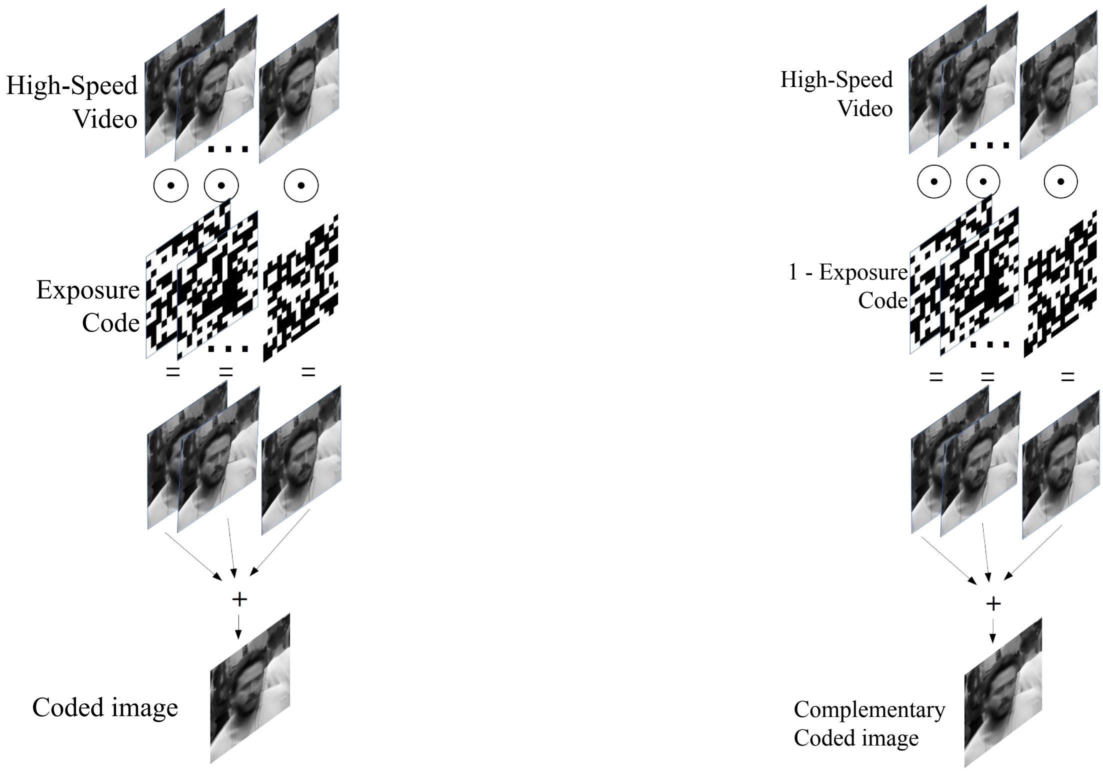


(a) Global coded-exposure

(b) Pixel-wise coded-exposure

Fig. 2.1: In a global coded-exposure sensor, all the pixels either block or allow the incoming light within a single sub-exposure. However, in a pixel-wise coded-exposure sensor, each pixel can independently block or allow the incoming light onto the sensor.

pixel is divided into two different light-collecting buckets. When the shutter is open, the electrons generated by the incident light are collected in bucket-0. And when the shutter is closed, instead of the light being blocked, it's collected in bucket-1 of the same pixel. Hence, C2B now provides two compressed measurements: one compressed with the input exposure sequence, and another with the complementary exposure sequence. This has a significant impact on the fidelity of the reconstructed video as we show in Chapter 3. A visual and intuitive explanation of the operation of C2B is shown in Fig. 2.2.



(a) Bucket 0

(b) Bucket 1

Fig. 2.2: In a C2B sensor, each pixel is divided into two light-collecting buckets. In a particular sub-exposure, if the exposure code is 1, then bucket-0 collects the incoming light. Else, if the exposure code is 0, then instead of light getting blocked from entering the pixel (as in pixel-wise coded exposure), it gets collected by bucket-1. Hence, C2B sensors produce two different compressed measurements within a single exposure sequence.

2.2 Neuromorphic Event sensors

Event sensors trigger events asynchronously whenever there's a brightness change in the scene. Output of an event sensor is a 4-tuple (x, y, p, t) where x and y represent the location of the pixel in the sensor, $p \in [-1, +1]$ is the polarity of the triggered event and t is the micro-second precise timestamp at which the event was triggered. An event sensor triggers a positive or a negative event whenever the log brightness change is more than or less than a threshold τ respectively. The polarity p of the triggered event is given by:

$$p = \begin{cases} +1, \ log(I_{t+\delta t}) - log(I_t) \geq \tau \\ -1, \ log(I_{t+\delta t}) - log(I_t) \leq -\tau \end{cases}$$

where $\delta t > 0$ and is of the order of microseconds. In cases where the log brightness change at a particular pixel is within the threshold, the event sensor does not trigger any event and hence saves power and bandwidth. As event sensors output only the brightness changes, it is impossible to recover the absolute scene intensity information. Further, the noise introduced due to the non-ideal hardware and the event quantization, makes it more challenging to recover intensity image information.

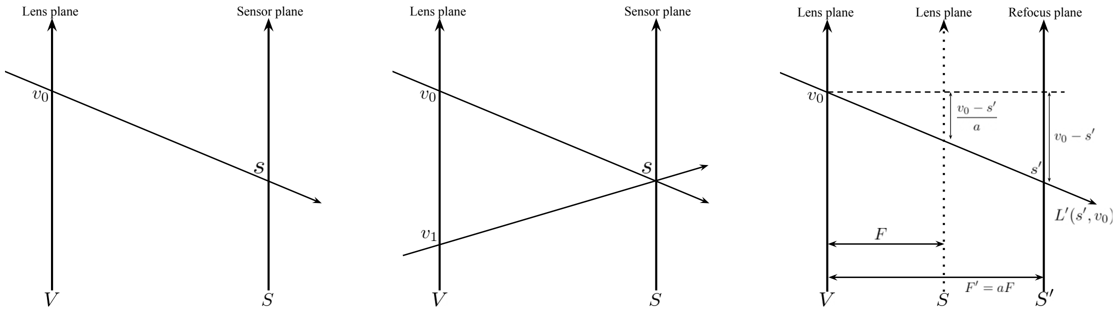
In a majority of our experiments a commercially available hybrid sensor named as "DAVIS240C" supplied by Inivation is used. This particular sensor consists of a co-located event sensor and an image sensor, each with a resolution of 180×240 pixels. We use a 6mm lens which approximately gives a horizontal and vertical field-of-view of 40.5 degrees and 49.4 degrees respectively. The image sensor gives a raw grayscale intensity image with 8 bits and the frame rate is 24 fps. The event sensor uses a AER (Address Event Representation) to record the data from the environment. The data is logged in a '.aedat' which is a binary file containing 8 bytes of data per line. The 8 bytes of data consists of the timestamp, the spatial location of the event and the event polarity. DAVIS240C is capable of sending out a maximum of 12M events per second.

2.3 Light field imaging

Since the beginning of photography, capturing only a 2D projection of the scene with accurate color reproduction and highest resolution has been of main interest. However, humans with their two eyes can perceive the world in 3D, whose information is lost when capturing only a 2D projection. While capturing this 2D projection, light rays arriving at each sensor pixel from different directions of the corresponding scene point get integrated. A LF camera on the other hand tries to preserve the intensity values of the different light-rays falling on the sensor from different directions. Thus, LF enables the representation of light-rays at any spatial location and in any direction.

2.3.1 Novel view synthesis and refocusing with LF

LF images have found applications in novel-view synthesis and post-capture refocusing. This is possible, as the LF image captures intensity of the light-ray arriving from different directions. To understand how LF enables post-capture refocusing and novel view synthesis, we create an analogy of the LF in 2 dimensions. This is also known as the *Flatland analogy* where the LF image is only 2D and an ordinary picture is 1D. In the flatland analogy, a light-ray is represented by a line on a plane which can be characterized by specifying two points through which it passes. This representation is shown in Fig. 2.3a, known as the *light slab parameterization*. From this representation, the intensity of any light-ray passing through coordinates (s, v) can be represented as $L(s, v)$.



(a) Light slab parameterization (b) Default focus (c) Refocused image

Fig. 2.3: (a)We show the 2D light-slab representation of a 2D LF signal. The light rays are represented as arrows whose direction and position is specified by its intersection on two axes S and V. (b) We show the light-rays representing the default LF signal directly captured by the camera. (c) The LF signal can be appropriately transformed to refocus the image onto different depth planes of the scene.

Novel view synthesis Synthesizing novel views of the scene from the complete LF function is very straightforward. If one needs to generate an image arriving from different directions, v_0 and v_1 then one simply needs to evaluate the function $L(s, v_0)$ and $L(s, v_1)$ for various values of s, respectively.

Post-capture refocusing A conventional camera with a finite aperture has a finite depth-of-field and the scene within that finite region appear to be focused (see Fig. 2.3b). By varying the distance between the lens and the sensor, different depth regions of the scene can be brought to focus. However, once the image has been captured, it becomes extremely complicated to control which region of the scene is in focus. Capturing a LF image on the other hand allows us to control the focus after the picture has been captured. The process of refocusing is illustrated in Fig. 2.3c. In Fig. 2.3c, we denote the lens place and the sensor plane by V and S axes. When the sensor position is moved from S to S', a different plane of scene comes into focus. This novel image where the plane of focus and depth-of-field are different than the original image can be computed when we have acquired the whole LF function $L(s, v)$. The LF at S' at a distance aF from V axis can be written as

$$L'\left(s', v_0\right) = L\left(v_0 - \frac{v_0 - s'}{a}, v_0\right) \tag{2.1}$$

where $v_0 \in v$ is one angular position on the lens-plane and a is the ratio between the distances of the current and original sensor plane from the lens plane. Now, the intensity at the sensor position s can be written as

$$I(s) = \int_v L'(s', v)\, dv = \int_v L\left(v - \frac{v - s'}{a}, v\right)\, dv\ . \tag{2.2}$$

$I(s)$ gives the refocused image obtained by virtually shifting the sensor plane in the camera.

CHAPTER 3

High Frame-Rate Video Reconstruction from Coded-Exposure Sensors

3.1 Introduction

In Chapter 1 we discussed that densely sampling the temporal dimension of the plenoptic function requires specialized hardware. A primary reason is that the hardware should be equipped to handle the large bandwidth of data being generated in real-time. Hence, a typical workaround is to first acquire a low frame rate video where only a small amount of data is generated during capture. Then a computational technique is used to upsample the videos temporally and obtain a high frame-rate video (Herbst *et al.*, 2009; Niklaus *et al.*, 2017*a*,*b*; Jiang *et al.*, 2018). However, these techniques are highly ill-posed due to the loss of motion information between successive input frames. In this chapter, we discuss a coded-exposure sensor based system for high frame-rate video reconstruction. These sensors compress the complete motion information in the scene into frame-like measurements. This is done via modulating the shutter of each pixel at a rate much higher than that of the sensor frame-rate. This is equivalent to temporally multiplexing several successive frames of the high-speed video into a single video frame. Hence, only a low frame-rate video is generated and the sensor does not have to handle large bandwidths. However, recovering several individual frames from the compressed measurement is still ill-posed and requires strong video signal priors (Baraniuk *et al.*, 2017). Several systems and methods have been proposed over the years to solve this ill-posed problem of recovering the high-speed video (Raskar *et al.*, 2006; Gu *et al.*, 2010; Reddy *et al.*, 2011; Holloway *et al.*, 2012; Llull *et al.*, 2013; Liu *et al.*, 2013; Iliadis *et al.*, 2018; Yoshida *et al.*, 2018; Iliadis *et al.*, 2020; Martel *et al.*, 2020; Li *et al.*, 2020).

Coded-exposure techniques have found applications in motion deblurring and high frame-rate video recovery (Raskar *et al.*, 2006; Reddy *et al.*, 2011; Holloway *et al.*,

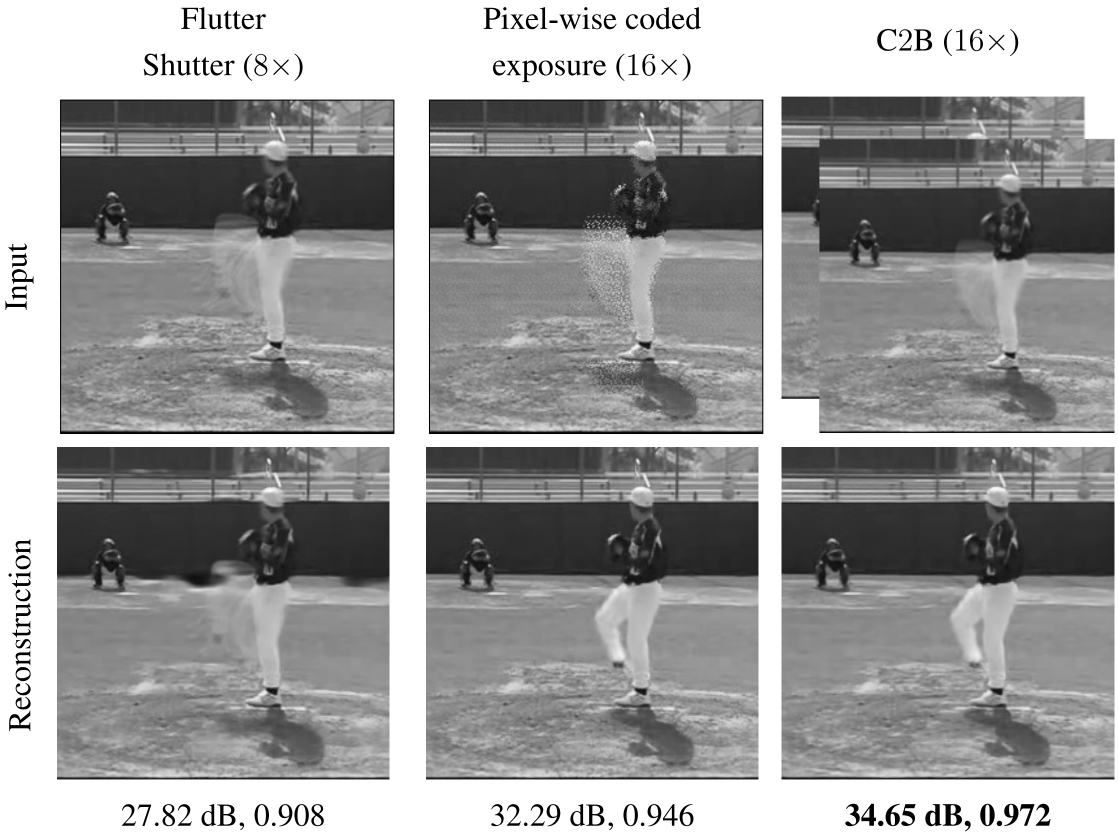


Fig. 3.1: We propose a unified deep learning-based framework that allows us to compare the performance of various coded exposure techniques. The figure shows the input and the middle frame of the reconstructed video for each of the exposure techniques.

2012; Llull *et al.*, 2013; Liu *et al.*, 2013; Yoshida *et al.*, 2018; Iliadis *et al.*, 2020; Martel *et al.*, 2020; Li *et al.*, 2020; Anupama *et al.*, 2021). Several techniques have been proposed to recover the high frame-rate video from the temporally compressed, low data-bandwidth measurements. While deep-learning techniques have shown promising results for video recovery, they generally employ fully-connected networks and recover the video one patch at a time (Yoshida *et al.*, 2018; Iliadis *et al.*, 2020). However, fully connected networks have fallen out of favor as they are hard to scale up for large spatial/temporal resolutions. Hence, we propose a fully-convolutional learning framework, that enables full resolution video reconstruction in a single forward pass. Martel *et al.* (2020) demonstrated that a fully convolutional network provides better reconstruction results than fully connected networks. In Sec. 3.2.1, we provide an intuitive explanation for why a convolutional network with local spatial connectivity is actually more suitable for this problem than fully connected networks with global connectivity over a small spatial patch. Our framework also uses the recently proposed SVC layer (Okawara *et al.*, 2020) that has shown to be effective for feature extraction from a coded image input. Our proposed algorithm is divided into two stages, where the first stage uses the SVC layer for an exposure code aware feature extraction. In the second stage, a deep, fully convolutional neural network is used to learn the non-linear mapping to the full resolution video sequence. Due to the use of the SVC layer, our proposed framework can be adapted with minimal changes to both global and pixel-level coded exposure techniques. We evaluate the proposed technique on recovering video from both pixel-wise and global coded exposure technique. As expected, pixel-wise coded exposure techniques produce much better video reconstructions than global coded exposure technique such as FS.

Recently, a novel prototype sensor based on multi-bucket pixels named C2B sensor was introduced by Sarhangnejad *et al.* (2019). While allowing for per-pixel control of the "shutter", this sensor can acquire two compressed measurements in a single exposure. This is achieved by using 2 light-collecting buckets per pixel and having control which bucket collects the incoming photons. With C2B giving us two compressed measurements, we can now broadly classify the coded exposure techniques into two categories: a) single compressed measurement (such as FS and pixel-wise coding) (Holloway *et al.*, 2012; Raskar *et al.*, 2006; Llull *et al.*, 2013; Reddy *et al.*, 2011; Liu *et al.*, 2013; Iliadis *et al.*, 2018; Yoshida *et al.*, 2018; Iliadis *et al.*, 2020; Martel *et al.*, 2020; Li *et al.*, 2020) and b) two compressed measurements per exposure (Sarhangnejad *et al.*, 2019). It is expected that two measurements should lead to better video reconstruction quality compared to a single measurement. However, the performance improvement provided by two compressed measurements over a single compressed measurement is yet to be investigated. As the C2B sensor is recently introduced, no previous video reconstruction algorithm exists that utilizes information from both the compressed measurements. Hence, there is no extensive quantitative or qualitative comparison between the single and two compressed measurement techniques. Such a comparison can help determine how much advantage is gained by acquiring two measurements over just one. This comparison of the different sensing architectures will also provide users with a tool to determine which sensing technique is better for a given scenario.

Due to the use of SVC layer, our technique can be adapted to high frame-rate video recovery from the C2B sensor with minimal changes to the architecture. This makes our proposed algorithm the first technique to recover high frame-rate video from the C2B sensor exploiting the two compressed measurements. We make an extensive quantitative comparison of video reconstruction quality from global, pixel-wise and C2B sensing techniques. We show that our proposed learning-based framework provides state of the art results on all three sensing techniques. We also confirm that acquiring two compressed measurements as in C2B is better than capturing just a single compressed measurement. And the advantage of having two compressed measurements becomes significant for a largely stationary scene (Fig. 3.8). However, C2B is only marginally beneficial over a single pixel-wise coded compressed measurement when most scene points undergo motion.

In summary we make the following contributions:

- We propose a unified deep-learning-based framework for video reconstruction from three different coded exposure imaging techniques.
- Our proposed approach matches or exceeds the reconstruction quality of state-of-the-art video reconstruction algorithms for each of the three sensing techniques.
- We show that C2B has significant advantage over per-pixel exposure coding in reconstructing videos of scenes that are mostly static.

3.1.1 Related Work

High speed imaging techniques with conventional sensor Conventional image sensors acquire videos at about 30 fps, with each exposure duration being shorter than 1/30s. Hence, multiple frames can be interpolated in time between the successive videos frames of the acquired low frame-rate video. Frame interpolation techniques (Herbst *et al.*, 2009; Niklaus *et al.*, 2017*a*,*b*; Jiang *et al.*, 2018) can be used to interpolate these frames between, thereby increasing the video frame-rate. When a long exposure is used, a blurred frame is acquired which encodes the full motion information in the motion blur. Recently learning-based methods (Jin *et al.*, 2018; Purohit *et al.*, 2019) have been used to decode the motion information from a single blurred frame into multiple video frames.

Computational Imaging techniques For scenes with little to no depth variations, techniques using arrays of low-cost, low-frame-rate cameras have shown to be effective at computationally recovering the high frame rate video (Wilburn *et al.*, 2005; Shechtman *et al.*, 2005; Agrawal *et al.*, 2010). A hybrid imaging system of two cameras: one low-frame-rate but high spatial resolution sensor, and one high-frame-rate but low spatial resolution sensor, has been proposed for image deblurring (Nayar and Ben-Ezra, 2004) and high spatio-temporal resolution video recovery (Paliwal and Kalantari, 2020). Recently, a hybrid imaging system consisting of image and event sensor has been proposed for high speed image reconstruction (Shedligeri and Mitra, 2019; Wang *et al.*, 2019*c*, 2020).

Motivated from the compressive sensing theory, several imaging architectures have been proposed for video compressive sensing problem (Baraniuk *et al.*, 2017). Flutter shutter is a global exposure coding technique which was first introduced for motion deblurring (Raskar *et al.*, 2006) and then extended for video recovery from the compressed measurements (Holloway *et al.*, 2012). A pixel-wise coded exposure system was proposed in (Reddy *et al.*, 2011) which demonstrated the recovery of high temporal resolution video from measurements compressed using spatial light modulator. A per-pixel control of the exposure was shown in (Liu *et al.*, 2013), using only a commercially available CMOS image sensor without the need for any other hardware. The recently introduced multi-bucket sensors such as *Coded-2-Bucket* cameras (Sarhangnejad *et al.*, 2019; Wei *et al.*, 2018), have reduced the complexity of per-pixel exposure control to a great extent. As video recovery from the compressed measurements is an ill-posed problem, strong signal priors are necessary for solving the inverse problem. While analytical priors such as wavelet domain sparsity (Reddy *et al.*, 2011; Park and Wakin, 2009), TV-regularization (Yuan, 2016) have been used, learning based algorithms such as Gaussian mixture models (Yang *et al.*, 2014), dictionary learning (Liu *et al.*, 2013) and neural network based models (Iliadis *et al.*, 2018, 2020; Yoshida *et al.*, 2018) have shown better performance than analytical priors. While many of the deep learning based methods use fully connected networks for the signal recovery, a very recent paper (Li *et al.*, 2020) uses a fully convolutional network to learn a denoising prior to iteratively solve the inverse problem.

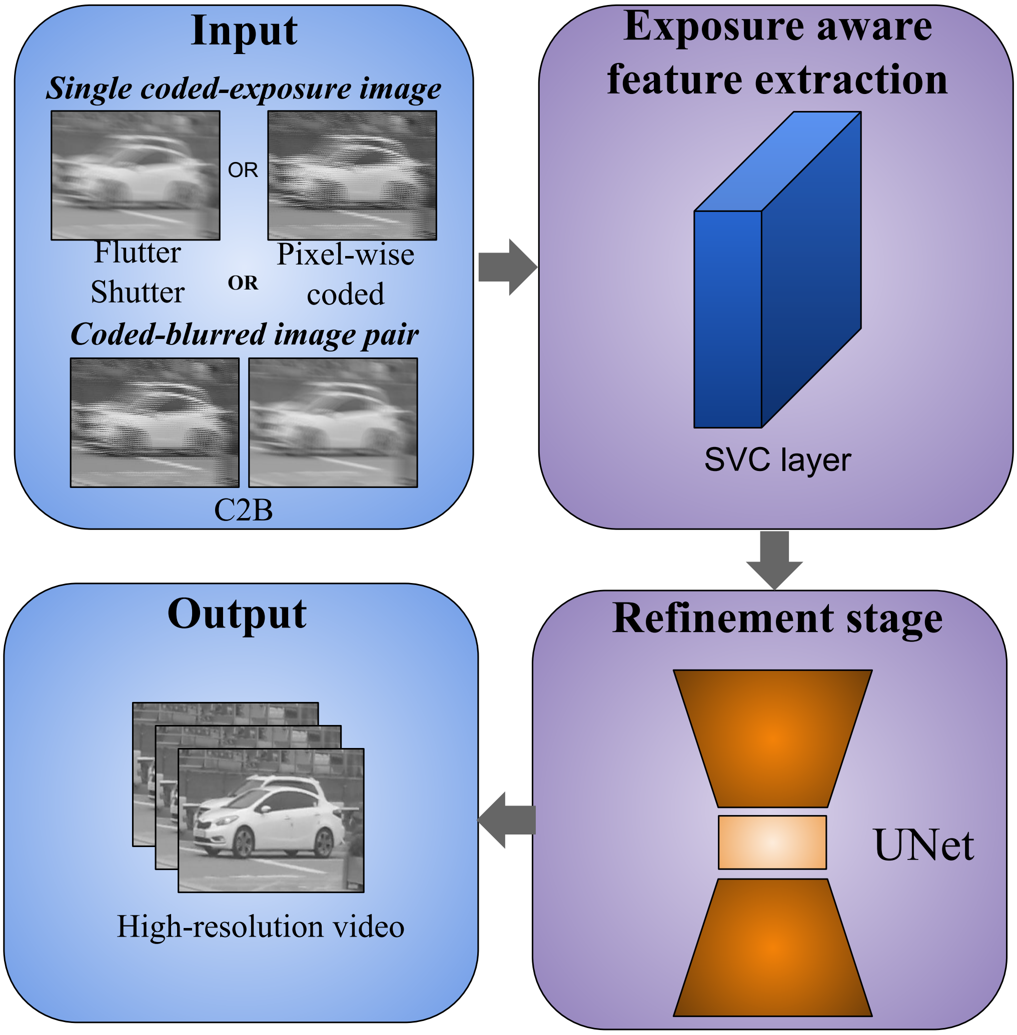


Fig. 3.2: Our proposed algorithm takes in compressed measurements from the different coded exposure techniques as input and output full spatial and temporal resolution video in a single forward pass. Our proposed algorithm is fully convolutional and consists of a feature extraction stage and a refinement stage. The feature extraction stage consists of a SVC layer where, unlike the standard convolutional layer (see Fig. 3.3), the weights of the SVC layer vary spatially (see Fig. 3.4).

3.2 A Unified Framework for Compressive Video Recovery Using Fully Convolutional Network

In this section, we elaborate on our proposed method to obtain the video signal from its compressed measurements. Our proposed algorithm takes in as input the compressed video measurements and outputs the video sequence at full spatial and temporal res-

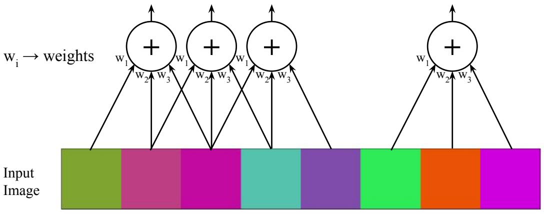


Fig. 3.3: **Standard Convolutional layer:** We show a convolutional layer in 1D where the weights $[w_1, w_2, w_3]$ act on the input image by sharing the same weights across different pixels. Unlike a fully-connected layer, a convolutional layer is locally connected and shares weights across the whole input image.

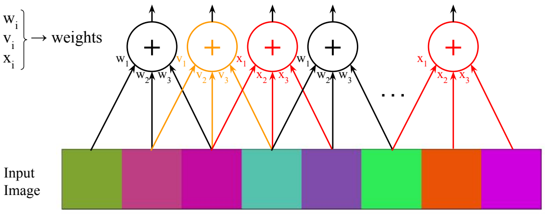


Fig. 3.4: **Shift-Variant Convolutional layer:** Like a convolutional layer, SVC layer is still locally connected where each location of the output feature map is affected by only a small subset of pixels neighboring to the current pixel. However, in contrast to a standard convolutional layer, SVC layer does not share the same weights across the whole image. Here we show 3 different sets of weights w_i, v_i, x_i, that operate on the input image. Note that, in the figure, the weights w_i, v_i, x_i are shared for every third pixel.

olution in a single forward pass. The proposed architecture consists of two stages, as shown in Fig. 3.2. First, features are extracted from the compressed measurements using an exposure aware feature extraction stage consisting of SVC layer. In the second stage, a deep neural network takes in the extracted features and outputs the full resolution video sequence. Our network architecture is flexible enough that it can be used for video reconstruction from all three coded exposure techniques considered here. All we need to do is train the network for these different inputs.

In Sec. 3.2.1, we provide motivation for using CNN for extracting relevant features from the compressed measurements. In Sec. 3.2.2, we elaborate on the use of SVC layer for handling pixel-wise coded exposure measurements and in Sec. 3.2.3 we specify the loss function used in the training our network.

3.2.1 Motivation for Using CNN

Several previous learning-based algorithms for compressive video recovery from coded exposure techniques have used fully connected networks (Yoshida *et al.*, 2018; Iliadis *et al.*, 2020). In (Martel *et al.*, 2020), it has been shown that a fully convolutional network provides better reconstruction than fully connected networks for compressive video sensing. This section shows that a fully convolutional network is a better choice for solving our problem than a fully connected network.

For coded exposure techniques, each pixel in the compressed measurement is a linear combination of the underlying video sequence at that pixel alone. As there is no spatial multiplexing involved, it is possible to recover the video sequence at each pixel independently of the neighboring pixels. However, by using the information in a small neighborhood of a pixel, we can exploit the spatio-temporal redundancy inherent in natural video signals. Fully connected networks that are used in previous works provide global connectivity at the cost of much larger computational complexity and learning parameters. Thus, they should be used for solving inverse problems where global multiplexing occurs in the forward model, such as FlatCam (Asif *et al.*, 2016). With a toy example and elementary mathematical operations, we demonstrate next that fully connected networks with global connectivity are an overkill for the task of video recovery from coded-exposure imaging. And fully convolutional networks with local spatial connectivity are a better design choice for our problem.

Toy example demonstration

Consider a video signal S of size $H \times W \times T$ with s_t representing each of the T frames of the video signal. A binary exposure sequence Φ of dimension $H \times W \times T$ is used for temporally multiplexing the signal S into the measurement Y. Mathematically, we can write the forward model as:

$$Y = \sum_{t=1}^{T} \phi_t \odot s_t \, , \tag{3.1}$$

where ϕ_t represents the code corresponding to each frame of Φ and $\odot$ represents element-wise multiplication.

The linear system in Eq. (3.1) can be represented in the matrix-vector form as follows:

$$\mathbf{Y} = \mathbf{\Phi}\mathbf{S} \,, \tag{3.2}$$

where $\mathbf{\Phi}$ is a matrix representation of Φ and S is a column vector obtained by vectorizing S. The minimum L_2-norm solution for the signal S can be obtained by:

$$\min_{\mathbf{S}} \|\mathbf{S}\|_2 \tag{3.3}$$

$$\text{s.t. } \mathbf{Y} = \mathbf{\Phi}\mathbf{S} \,. \tag{3.4}$$

Note that there are better reconstruction techniques such as dictionary learning which uses L_0 or L_1 norm on sparse transform coefficients of S (Liu *et al.*, 2013). But our objective here is to show that CNN is appropriate for solving our inverse problem and hence we only provide a justification with L_2-norm, that has a closed-form solution. The approximate solution $\tilde{\text{S}}$ for Eq. (3.3) is given by,

$$\tilde{\mathbf{S}} = \mathbf{\Phi}^{\dagger}\mathbf{Y} \,, \tag{3.5}$$

$$\mathbf{\Phi}^{\dagger} = \mathbf{\Phi}^{T}(\mathbf{\Phi}\mathbf{\Phi}^{T})^{-1}. \tag{3.6}$$

We notice that the matrix $\mathbf{\Phi}\mathbf{\Phi}^{T}$ is a diagonal matrix of dimension $HW \times HW$, and so is the matrix $(\mathbf{\Phi}\mathbf{\Phi}^{T})^{-1}$. As shown in Fig. 3.5, the matrix $\mathbf{\Phi}^{\dagger}$ is the matrix $\mathbf{\Phi}^{T}$ whose columns are scaled by the entries of the diagonal matrix $(\mathbf{\Phi}\mathbf{\Phi}^{T})^{-1}$. From the solution shown in Fig 3.5, it is clear that the temporal sequence at each pixel of the video is recovered only from the compressed measurement captured at that pixel. For example, the estimated temporal sequence at the j^{th} pixel depends only on the compressed measurement at the j^{th} pixel. Just with a minimum L2 norm solution, we observe that a fully-connected network that provides global connectivity at the cost of processing the image patch-by-patch is not a good architectural choice. A CNN while providing local connectivity is able to exploit the larger context information that comes from processing the whole image at once. The local connectivity does not hinder the reconstruction process as the compressed information at each pixel is available at that

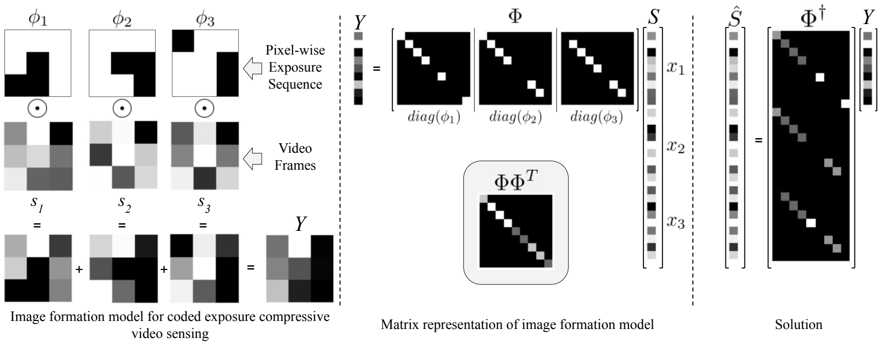


Fig. 3.5: We show a toy example of pixel-wise coded exposure technique for compressing a video sequence of size $3 \times 3 \times 3$. $\mathbf{\Phi}$ and $\mathbf{S}$ are the matrix and vector representation of the exposure sequence Φ and the video sequence S, respectively. From the pseudo-inverse solution we see that the temporal video sequence reconstruction at any pixel depends only on the measurement and the code at that pixel itself. This motivates our choice of a fully convolutional design.

pixel only.

3.2.2 Feature Extraction Using Shift-variant convolutional

In Sec. 3.2.1, we determined that to recover a video at a particular pixel, only that pixel's compressed measurements are necessary. Hence, the local connectivity offered by CNNs can be efficiently used for the task of recovering the underlying video signal. However, CNNs share the same weights across the whole input image. In pixel-wise coded exposure, the compressed measurement can be encoded using a different exposure sequence at each pixel. From Eq. (3.5) and Fig. 3.5, we see that the estimated video sequence at a particular pixel is dependent on the exposure sequence at that particular pixel. Hence, for pixels with different exposure sequence, using a different set of weights in the convolutional layer is desirable.

In FS video camera, each pixel in the image shares the same coded exposure sequence. Hence, identical weights can be used to recover the underlying video signal for all the pixels. Thus, for recovering video sequences from the FS camera, we build our inversion stage as a standard convolutional layer as it achieves the functions mentioned above: local connectivity and shared weights across the whole image.

In pixel-wise coded exposure and C2B architectures, the underlying coded exposure sequence can change from one pixel to the next. In practice, a predetermined code of size $m \times n \times T$ is repeated over the entire image with a stride of $m \times n$ pixels. Hence, a standard convolutional layer cannot be directly used as it shares the same set of weights across the whole image. Instead, a convolutional layer, which can share weights for every $m \times n^{th}$ pixel, is desirable. Such a convolutional layer whose weights *vary* in a local neighborhood of $m \times n$ pixels was proposed by Okawara *et al.* (2020) called SVC layer (see Fig. 3.4). This layer allows the network the freedom to learn different weights to invert the linear system when the underlying exposure sequence is different. Hence, we use this layer to extract adaptive features from the input compressed measurement. These extracted features are input to the next stage of the network, which predicts the full resolution video sequence.

3.2.3 Refinement Stage

The refinement stage takes as input the features extracted from the SVC layer and outputs a full resolution video sequence $\hat{S}$. Our refinement stage consists of a U-Net (Ronneberger *et al.*, 2015) like deep neural network. Our proposed U-Net model consists of 3 encoder stages followed by a bottleneck layer and 3 decoder stages. In each of the encoder stages, the feature maps are downsampled spatially by a factor of 2 and upsampled by the same factor in corresponding decoder stage. The output of this network is supervised using L_1 loss function. We also add a TV-smoothness loss on the final predicted video sequence. Our overall loss function then becomes,

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{ref} + \lambda_{tv}\mathcal{L}_{tv} \\ \mathcal{L}_{ref} &= \|\hat{S} - S\|_1 \\ \mathcal{L}_{tv} &= \|\nabla \hat{S}\|_1 \end{aligned} \tag{3.7}$$

where ∇ is the gradient operator in the x-y directions and λ_{tv} weights the smoothness term in the overall loss function.

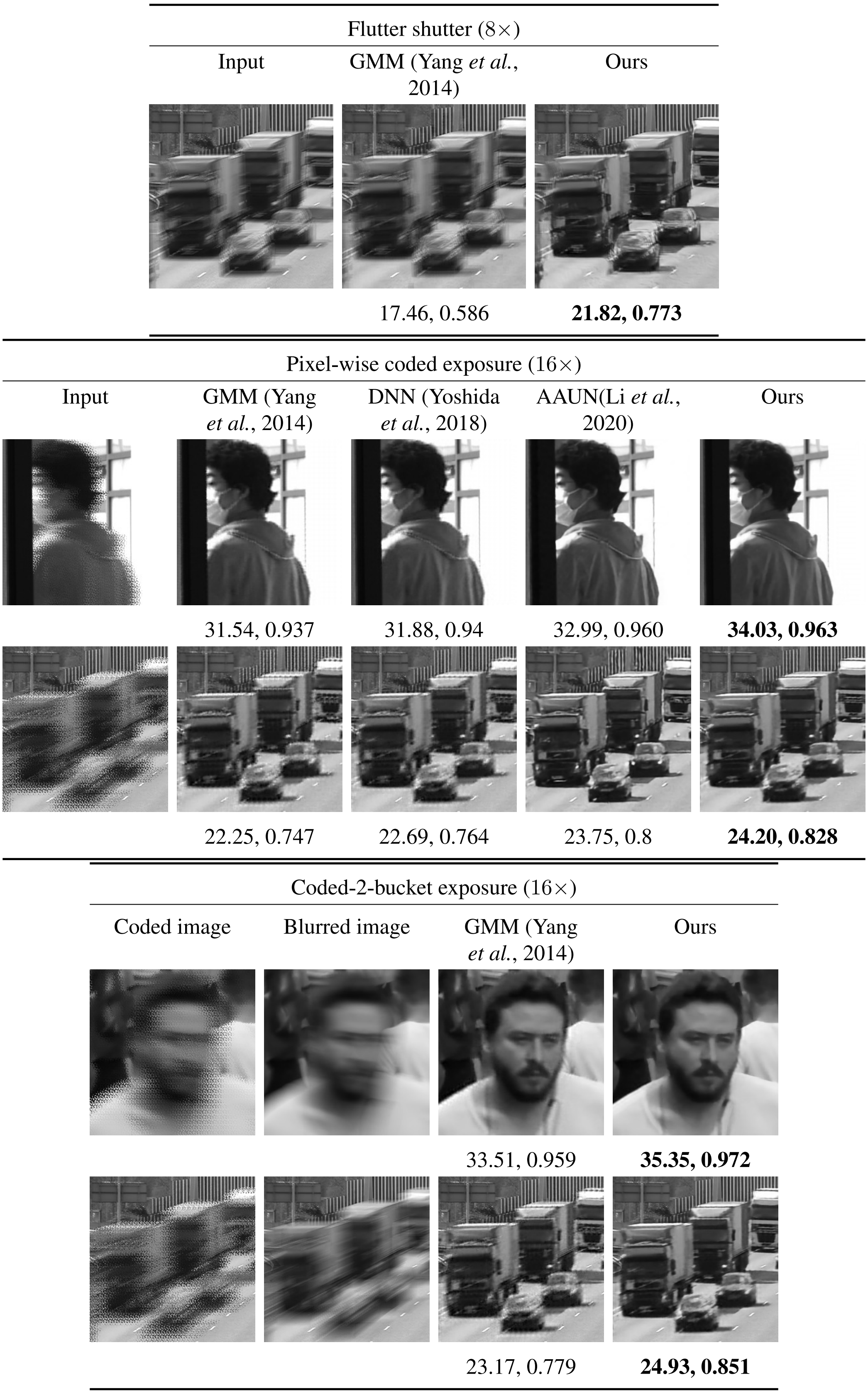


Fig. 3.6: Visual comparison of middle frame from the reconstructed video sequences from various reconstruction algorithms. Our proposed method performs better than the existing methods GMM (Yang *et al.*, 2014), DNN (Yoshida *et al.*, 2018), and also doesn't suffer from block artifacts caused by patch-wise reconstruction. As expected, C2B produces better results than pixel-wise coded imaging. FS lags far behind.

3.3 Experimental and Training Setup

Ground truth data preparation: We trained our proposed network using GoPro dataset (Nah *et al.*, 2017) consisting of 22 video sequences at a frame rate of 240 fps and spatial resolution of 720×1280. The first 512 frames from each of the 22 sequences are spatially downsampled by 2 for preparing the training data. Overlapping video patches of size $64 \times 64 \times 16$ (height×width×frames) are extracted from the video sequences by using a sliding 3D window of $(32, 32, 8)$ pixels resulting in $263{,}340$ training patches. Similarly, for 8-frame reconstruction, we extracted video patches of size $64 \times 64 \times 8$ and shifting the window by $(32, 32, 4)$ pixels. The network was trained in PyTorch (Paszke *et al.*, 2019*b*) using Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.0001, λ_{tv} of 0.1 and batch size of 50 for 500 epochs[1].

Network architecture for each sensing technique: We trained our network separately for each of the different coded exposure techniques - *FS, Pixel-wise coded exposure*, and *C2B*. For FS, we trained our proposed network for 16-frame reconstruction and 8-frame reconstruction. As FS uses global code, a standard convolutional layer is used as a feature extraction layer in place of the SVC layer. We use the SVC layer as described in Sec. 3.2.2 as a feature extraction stage for pixel-wise coded exposure and C2B.

Input to the network: In the case of FS, the input to the network is a single coded exposure image obtained by multiplexing with a global exposure code. We used the exposure code obtained by maximizing the minimum of the DFT values' magnitude and minimizing the variance of the DFT values (Raskar *et al.*, 2006), over all possible binary codes. For the case of pixel-wise coded exposure, the coded mask of size $8 \times 8 \times 16$ is repeated spatially to make it the same dimension as input, which is then used for multiplexing. We used the *optimized SBE mask* exposure code proposed in (Yoshida *et al.*, 2018) for this purpose (see Fig. 3.7). In the case of C2B exposure, the input to the network can either be a pair of coded and complement-coded images or a pair of coded and fully-exposed images. The output of the C2B sensor is two images that

[1] `https://github.com/asprasan/unified_framework`

Exposure	Algorithm	Test data	
		DNN set (Yoshida *et al.*, 2018)	GoPro set (Nah *et al.*, 2017)
FS 8×	GMM (Yang *et al.*, 2014)	23.90, 0.818	23.30, 0.766
	Ours	**24.06, 0.833**	**25.03, 0.811**
FS 16×	GMM (Yang *et al.*, 2014)	21.50, 0.738	21.45, 0.697
	Ours	**21.69, 0.752**	**21.61, 0.710**
Pixel-wise coded 16×	GMM (Yang *et al.*, 2014)	29.31, 0.898	29.94, 0.887
	DNN (Yoshida *et al.*, 2018)	30.21, 0.905	30.27, 0.890
	AAUN (Li *et al.*, 2020)	28.5, 0.882	31.6, 0.910
	Ours	**31.14, 0.925**	**31.76, 0.914**
C2B 16×	GMM (Yang *et al.*, 2014)	30.94, 0.914	30.84, 0.898
	Ours	**32.23, 0.935**	**32.34, 0.920**

Table 3.1: Quantitative results for different coded exposure techniques and reconstruction algorithms. The table lists average PSNR(dB) and SSIM of reconstructed videos from *DNN set* (Yoshida *et al.*, 2018) and *GoPro set* (Nah *et al.*, 2017).

are coded using complementary exposure sequences (i.e., Φ and $1 - \Phi$). We used the same exposure pattern *optimized SBE mask* from (Yoshida *et al.*, 2018) for C2B exposure as well. The fully-exposed or blurred image is obtained by adding the coded and complementary coded images. The image pair for the C2B sensor are stacked as two channels and provided as input to the proposed algorithm.

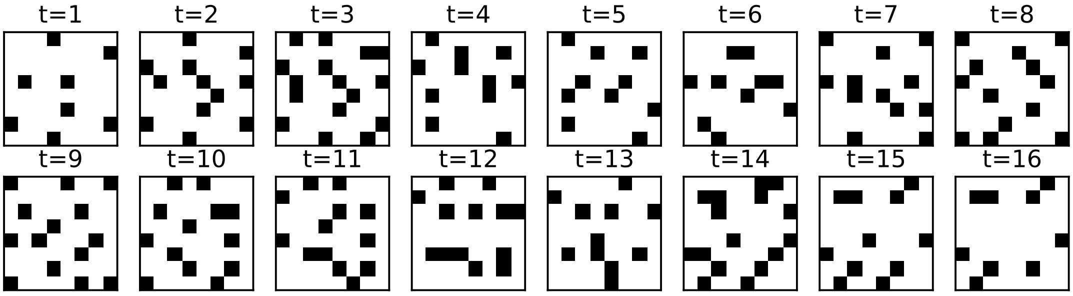


Fig. 3.7: Optimized SBE code from (Yoshida *et al.*, 2018) for multiplexing 16 frames.

Exposure	CPU run-time (GPU run-time) in seconds			
	GMM (Yang *et al.*, 2014)	DNN (Yoshida *et al.*, 2018)	AAUN (Li *et al.*, 2020)	Ours
Pixel-wise	78.7 (–)	4.6 (2.7)	11.1 (0.3)	3.6 (0.011)
C2B	96.4 (–)	–	–	4.1 (0.013)

Table 3.2: Run time for various algorithms to reconstruct a single $256 \times 256 \times 16$ frame sequence. For algorithms that are accelerated by GPU, the run times are provided in parentheses. The run times are for an Intel i7 CPU and Nvidia GeForce 2080 Ti GPU.

3.4 Experimental Results

3.4.1 Analysis of Video Reconstruction for Various Compressive Sensing Systems

In this section, we qualitatively and quantitatively assess video reconstruction from compressed measurements captured by different coded exposure techniques - *FS*, *pixel-wise coded exposure*, and *C2B*. We compared our proposed method with existing state-of-the-art algorithms for video reconstruction such as GMM-based inversion (Yang *et al.*, 2014), DNN (Yoshida *et al.*, 2018) and AAUN (Li *et al.*, 2020). We used two sets of test videos with a different spatial resolution to perform this analysis. First, we used the test set that was used for evaluation in *DNN (Yoshida* et al.*, 2018)*, consisting of 14 videos of spatial resolution 256×256 and 16 frames each. For the second set, we randomly selected 15 videos of resolution 720×1280 and 16 frames each, from the *GoPro test dataset (Nah* et al.*, 2017)*.

For FS, we compared our proposed method with the GMM-based video reconstruction method (Yang *et al.*, 2014) for 8-frame and 16-frame reconstructions. For single pixel-wise coded exposure sensing, we compare with GMM-based inversion (Yang *et al.*, 2014) and state-of-the-art deep learning based methods, DNN (Yoshida *et al.*, 2018) and AAUN (Li *et al.*, 2020), for 16-frame reconstruction. For C2B exposure, we compare with GMM-based inversion (Yang *et al.*, 2014) for 16-frame reconstruction from a pair of coded and blurred images. We trained the GMM (Yang *et al.*, 2014) model with 20 components using the same training dataset as described in Sec. 3.3. We used $8 \times 8 \times 8$ patches to train the GMM (Yang *et al.*, 2014) for 8-frame reconstruction and $8 \times 8 \times 16$ patches for 16-frame reconstruction. We used the pre-trained model for DNN proposed in (Yoshida *et al.*, 2018). We trained the AAUN (Li *et al.*, 2020) algorithm on the same training dataset as described in Sec. 3.3. The model was trained for 80 epochs on patches of size 128×128 for 16-frame reconstruction.

Comparison analysis Qualitative reconstruction results are shown in Fig. 3.6 and quantitative results are summarized in Table 3.1. FS produces satisfactory results for 8-frame reconstruction but struggles to reconstruct 16 frames. Pixel-wise coded exposure can perform 16-frame reconstruction with good fidelity. For natural images, the intensities in a small spatial neighborhood are correlated. Intuitively, using different exposure sequences for different pixels, is equivalent to making multiple measurements, which helps in recovering the information better. As our algorithm exploits the spatial correlation structure, the pixel-wise coded exposure technique will have an advantage over the global, FS imaging technique in the fidelity of the reconstructed video. The C2B exposure provides an additional advantage by capturing information that is lost by the pixel-wise coded exposure and hence produces better reconstruction than pixel-wise coded exposure. Overall, we observe a similar trend in the reconstruction performance of different sensing techniques in both GMM (Yang *et al.*, 2014) and our proposed model. We see that, overall, C2B provides the best reconstruction and FS performs the worst, while there is only a slight quantitative advantage for C2B when compared to pixel-wise exposure. We further compare the performance of pixel-wise coded exposure with C2B exposure in the following section.

Our proposed fully-convolutional model performs better than the existing methods, GMM (Yang *et al.*, 2014), DNN (Yoshida *et al.*, 2018) and AAUN (Li *et al.*, 2020), for all the sensing techniques. Since we reconstruct the full video, our proposed method doesn't suffer from block artifacts, which is seen in patch-wise reconstruction methods such as GMM and DNN. A comparison of run times of various algorithms on CPU as well as GPU has also been provided in Table 3.2. Patch-based reconstruction methods such as GMM and DNN require a significantly longer time to reconstruct a single video sequence compared to AAUN (Li *et al.*, 2020) and our algorithm. Being an iterative deep learning algorithm, AAUN (Li *et al.*, 2020) takes $3\times$ and $10\times$ longer time than

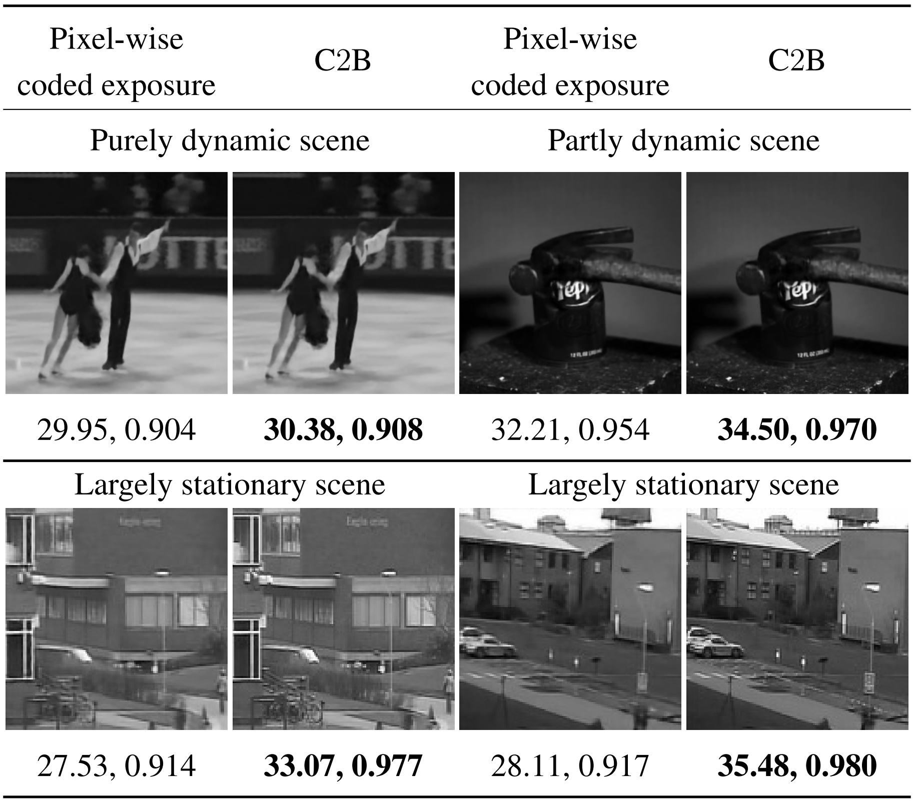


Fig. 3.8: Qualitative comparison of cropped middle frames from the reconstructed video sequences. When majority of the pixels do not see any motion C2B has a significant advantage, while being only marginally beneficial in the case where majority of the pixels see motion.

our proposed algorithm on CPU and GPU, respectively.

3.4.2 When Does C2B Have a Significant Advantage over Pixel-wise Coded Exposure?

In Sec. 3.4.1, we observe that C2B based sensing provides only a slight advantage compared to pixel-wise coded exposure technique. To analyze and identify the cases where C2B provides a significant advantage over pixel-wise coded exposure, we conduct experiments on different kinds of videos: purely dynamic sequences, partly-dynamic-partly-static sequences, and largely static sequences. We use our proposed method to compare video reconstruction from a pixel-wise coded exposure image and from a coded-blurred image pair obtained from C2B. We explain why we use a blurred image with the coded image as input through an ablation study in Sec 3.5. Fig. 3.8 shows reconstructed results for the different cases of video sequences mentioned above. For

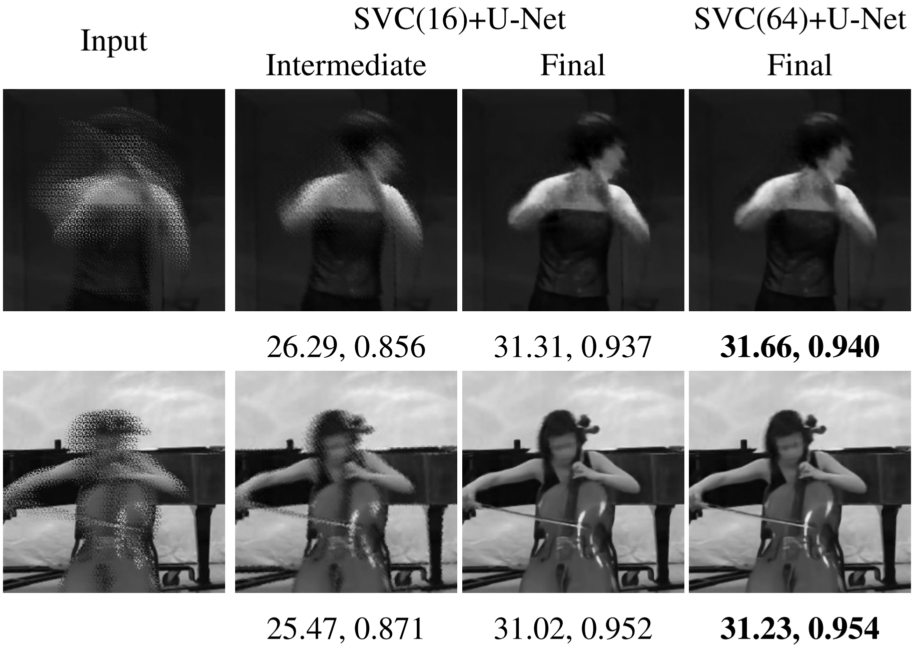


Fig. 3.9: The figure compares the middle frames from the reconstructed video sequences from two different architectural choices. It can be seen that SVC(64)+U-Net performs better than SVC(16)+U-Net in terms of the PSNR and SSIM.

purely dynamic scenes, C2B does not show a notable performance improvement over pixel-wise coded exposure. However, for videos containing significant static regions, C2B produces much better reconstruction results than pixel-wise coded exposure. If we consider a scene composed of both stationary and dynamic regions, the dynamic regions are better captured by the coded exposure image, while the stationary regions are better captured by the fully-exposed image. Therefore, it follows that videos containing stationary regions can be better recovered by using the additional information captured by C2B.

3.5 Ablation Study

3.5.1 Ablation study on proposed architecture

We explain some of the architectural choices that we made in developing our proposed network. We experimented with two different architectures for pixel-wise coded exposure - U-Net only, SVC(16) + U-Net, and SVC(64) + U-Net. SVC denotes the Shift-variant convolutional layer (Okawara *et al.*, 2020), and the following value in bracket specifies the number of output channels of the SVC layer. In U-Net only framework, we input the coded image directly to the standard U-Net architecture, which learns the mapping to the full resolution video sequence. In SVC(16)+U-Net, we implemented the SVC layer to produce an intermediate reconstruction from the input, followed by U-Net (Ronneberger *et al.*, 2015) to refine the intermediate reconstruction and produce the final high-quality video. While training the network, we supervise both the intermediate and final reconstructions using ground truth with a 0.5 weightage for intermediate reconstruction. In SVC(64)+U-Net, we modified the number of output channels of the SVC layer from 16 to 64. Therefore, instead of producing an intermediate reconstruction, the SVC layer extracts the features required to reconstruct the video. Here, we supervise the final reconstruction using ground truth while training. From Table 3.3, we observe that using SVC(64)+U-Net gives the best reconstruction results. It can also be observed that using an SVC layer instead of a standard convolutional layer provides a significant improvement in performance. The SVC layer also does not add significantly to the computational overhead. While, SVC(64)+U-Net model takes 0.011s, Unet-only model takes 0.009s per forward pass on a GPU for a $256 \times 256 \times 16$ video sequence. Therefore, we choose SVC(64)+U-Net architecture as our proposed method.

3.5.2 Ablation Study on C2B Input

The advantage of using C2B exposure is that it captures the complementary information otherwise lost in pixel-wise coded exposure. C2B captures two coded exposure images: coded image and complement-coded image. We can obtain a fully-exposed or blurred image by adding the coded and complementary coded images. There are two ways of representing the C2B input: a coded-complement image pair (see Sec. 2.1.2 and Figs. 2.2a and 2.2b) or coded-blurred image pair. In the coded-blurred image pair, one of the images is the coded image as shown in Fig. 2.2a, while the blurred image is defined to be the sum of both coded (Fig. 2.2a) and complementary coded images (Fig. 2.2b). We evaluated both the cases and determined that video reconstruction from a coded-blurred image pair performs marginally better than reconstruction from a coded-complement pair. The results are summarized in Table 3.3.

Exposure		DNN set (Yoshida *et al.*, 2018)		GoPro set (Nah *et al.*, 2017)	
		PSNR	SSIM	PSNR	SSIM
Pixel-wise coded	U-Net only	30.68	0.919	31.27	0.902
	SVC(16)+U-Net	30.89	0.921	31.56	0.910
	SVC(64)+U-Net	**31.14**	**0.925**	**31.76**	**0.914**
C2B	coded+complement	32.19	0.935	32.31	0.919
	coded+blurred	**32.23**	**0.935**	**32.34**	**0.920**

Table 3.3: Ablation studies on proposed architecture and C2B input. The table lists average PSNR(dB) and SSIM of reconstructed videos from *DNN set (Yoshida* et al.*, 2018)* and *GoPro set (Nah* et al.*, 2017)*.

Model	Noiseless		Noisy($\sigma = 0.01$)	
	PSNR	SSIM	PSNR	SSIM
FS (fixed)	21.61	0.752	21.28	0.707
FS (optimized)	**21.72**	**0.756**	**21.42**	**0.722**
Pixel-wise(fixed)	31.76	0.914	27.58	0.845
Pixel-wise(optimized)	**32.13**	**0.953**	**29.58**	**0.912**
C2B(fixed)	32.34	0.920	28.22	0.860
C2B(optimized)	**32.59**	**0.961**	**30.06**	**0.912**

Table 3.4: PSNR, SSIM comparison of reconstructed videos for FS, pixel-wise and C2B for *fixed* and *optimized* coded mask Φ. We observe better reconstruction performance for *optimized* mask for both the noisy and noiseless cases.

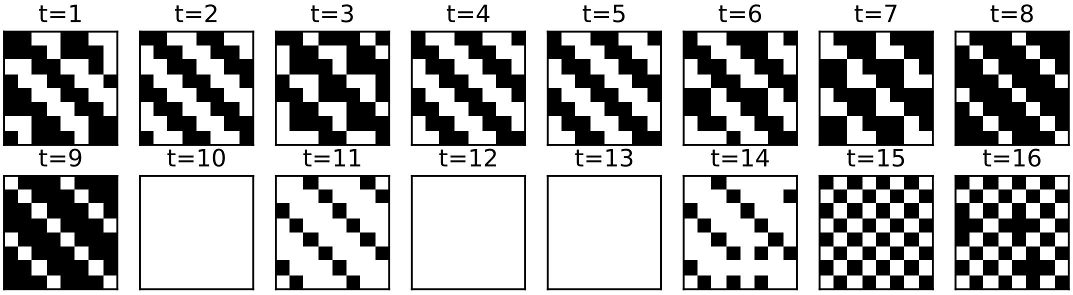


Fig. 3.10: Optimized C2B code from our algorithm for multiplexing 16 frames.

3.6 Learning the mask

Jointly learning the coded mask Φ and the reconstruction algorithm has been shown to provide better reconstruction results (Li *et al.*, 2020; Okawara *et al.*, 2020; Iliadis *et al.*, 2020). To demonstrate this, we jointly learn the coded mask Φ along with our proposed learning-based reconstruction algorithm. We add the weights of the mask Φ also as trainable parameters along with the other trainable network parameters. As the hardware sensors can use only binary mask patterns, we restrict the mask weights to be binary. Binarization is done via thresholding the weights before each forward pass through the network. As thresholding is non-differentiable, we follow (Hubara *et al.*, 2016) and use the *straight-through estimator* for computing gradients. In the straight-through estimator, to keep the weights Φ binary, we maintain a real-valued variable Φ_R following Hubara *et al.* (2016). In the forward-pass Φ_R is binarized by first applying the 'signum' function and then setting the negative values to 0. During the backward-pass, the loss gradients g_Φ are computed with respect to Φ. Now, updating Φ with these gradients will very likely make Φ a real-valued variable instead of a binary variable. And passing these gradients through the signum function to update Φ_R will cause the gradients to become zero. Hence, in (Hubara *et al.*, 2016), the gradients g_Φ are passed 'straight through' the signum function as is and the real-valued variable Φ_R is updated. Again, during the forward pass, Φ_R is binarized to Φ using signum function, continuing the training loop.

We use an identical training scheme and dataset as described in Sec. 3.3 for training the network with optimized sensor mask Φ. The mask Φ and the network are jointly trained for 16x reconstruction for the case of FS, pixel-wise exposure, and C2B. The trained network is evaluated on the GoPro test set, and the results are summarized in Table 3.4. We observe that for both the noiseless and the noisy cases, joint optimization of the coded mask and the reconstruction algorithm provides better performance. The gap between the fixed and optimized code is bigger for the noisy case.

3.7 Conclusion

We propose a unified deep learning-based framework to make a fair comparison of the video reconstruction performance of various coded exposure techniques. We make a mathematically informed choice for our framework that leads to the use of fully convolutional architecture over a fully connected one. Extensive experiments show that the proposed algorithm performs better than previous video reconstruction algorithms across all coded exposure techniques. The proposed unified learning framework is used to make an extensive quantitative and qualitative evaluation of the different coded exposure techniques. From this, we observe that C2B provides the best reconstruction performance, closely followed by the single pixel-wise coded exposure technique, while FS lags far behind. Our further analysis of C2B shows that a significant advantage is gained over pixel-wise coded exposure only when the scenes are largely static. However, when the majority of scene points undergo motion, C2B shows only a marginal benefit over acquiring a single pixel-wise coded exposure measurement.

CHAPTER 4

Photorealistic Image Reconstruction from Hybrid Intensity and Event based sensor

4.1 Introduction

In Chapter 3, we discussed a coded-exposure sensor based system that achieved a temporal upsampling of up to $16\times$ over the sensor's original frame-rate. For a commercial image sensor acquiring videos at 30 fps, the high-speed video is reconstructed at a frame-rate of 480 fps. This still falls short of the typical frame-rates of thousands of frames per second for a specialized high-speed video camera. Naively increasing the sensor's base frame-rate from 30 fps will again lead to an increase in the bandwidth requirement. Compressing more high-speed video frames into a single measurement is not viable due to the limitations of the current video recovery techniques. Hence, in this chapter we discuss a novel system consisting of a neuromorphic event sensor that promises video reconstruction at thousands of frames per second.

Neuromorphic event-based sensors (Lichtsteiner *et al.*, 2008) are a new generation of sensors that capture only the brightness changes at pixel-level. Based on whether the brightness has increased or decreased, the sensor outputs either a positive or a negative event. While traditional frame-based sensors output full-resolution frames at fixed frame-rate, event sensors output only these brightness changes as a sequence of events. In most natural scenes, these brightness changes are spatially sparse and hence the event sensor has to output only those sparse measurements. This allows the event sensors to operate at much lower bandwidth than any frame-based sensor. While these events may be spatially sparse, they can be temporally dense due to the microsecond temporal resolution of the event sensors. This can result in a frame-rate at an order of magnitude higher than that achieved with coded-exposure sensor in Chapter 3. Event sensors also possess several other advantages such as low power requirement and a high dynamic range of ~120 dB.

Event-based sensors convert the high-bandwidth high-frame rate video signal to a low-bandwidth stream of binary event measurements. However, the event stream cannot be directly visualized like a normal video, with which we as human beings are familiar with. This calls for an algorithm that can convert this stream of event data to a more familiar version of image frames. These reconstructed intensity frames could also be used as an input for traditional frame-based computer vision algorithms like multi-view stereo, object detection etc. Previous attempts (Munda *et al.*, 2018; Bardow *et al.*, 2016; Barua *et al.*, 2016; Scheerlinck *et al.*, 2018*a*) at converting the event stream into images have heavily relied on event data. Although these methods do a good job of recovering the intensity frames they suffer from two major disadvantages: a) The intensity frames don't look photorealistic and b) some of the objects in the scene can go missing in the recovered frames because they are not producing any events (edges parallel to the sensor motion do not trigger any events).

In this chapter, a method is proposed to reconstruct photorealistic intensity images at a high frame rate. As the absolute intensity and fine texture information is lost during the encoding of events, the information from the frames of the conventional image sensor is used to reconstruct photorealistic intensity images. The conventional image sensor will compensate for the spatial information lost due to encoding of events. The event sensor will compensate for the motion information lost due to fixed frame-rate sampling of the conventional image sensor. There exists a commercially available hybrid sensor consisting of a co-located low-frame rate intensity sensor and an event-based sensor called DAVIS (Berner *et al.*, 2013). Fig. 4.2 summarizes the overall approach to reconstruct the temporally dense photorealistic intensity images using the hybrid sensor. The proposed method has mainly four steps. In the first step, a dense depth map is estimated using successive intensity frames obtained from the traditional image sensor. For estimating depth, a traditional iterative optimization scheme is utilized, which is initialized by a depth map obtained from a deep learning based optical flow estimation algorithm. In the second step, the event data between successive intensity frames is mapped to multiple pseudo-intensity frames using (Munda *et al.*, 2018). Next, the pseudo-intensity frames and the dense depth maps obtained from the first step are used to estimate temporally dense camera ego-motion by direct visual odometry. And finally,

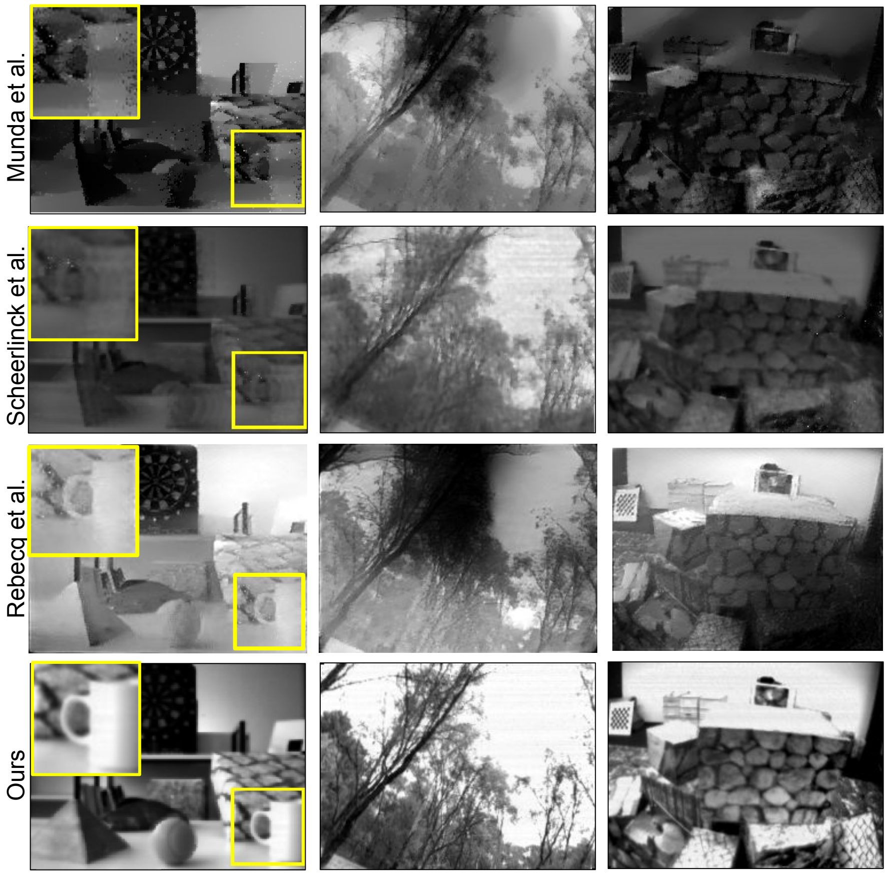


Fig. 4.1: Comparing the reconstruction from the proposed algorithm using a hybrid sensor data (such as DAVIS) with that of Complementary Filter (CF) (Scheerlinck *et al.*, 2018*a*), Manifold Reconstruction (MR) (Munda *et al.*, 2018) and E2Vid (Rebecq *et al.*, 2019*a*). Note that, MR only uses events for reconstruction. In column (a) inset the zoomed-in version of an image region is shown. We can clearly see that our proposed reconstruction method is able to recover the image region well compared to other state-of-the-art methods.

in the fourth step, warp the successive intensity frames are warped to intermediate temporal locations of the pseudo-intensity frames to obtain photo-realistic reconstruction. With extensive experiments, it is shown that our proposed method is able to reconstruct photorealistic intensity images at a high frame rate and is also robust to noisy events in the event stream. To summarize, the contributions of this work are:

- A pipeline is proposed which uses a hybrid event and low frame rate intensity sensor to reconstruct *temporally dense photorealistic intensity images*. This would be difficult to achieve with only either the conventional image sensor or the event sensor.
- Event data are used for estimating temporally dense sensor ego-motion and the low-frame rate intensity frames are used in estimating spatially dense depth map.

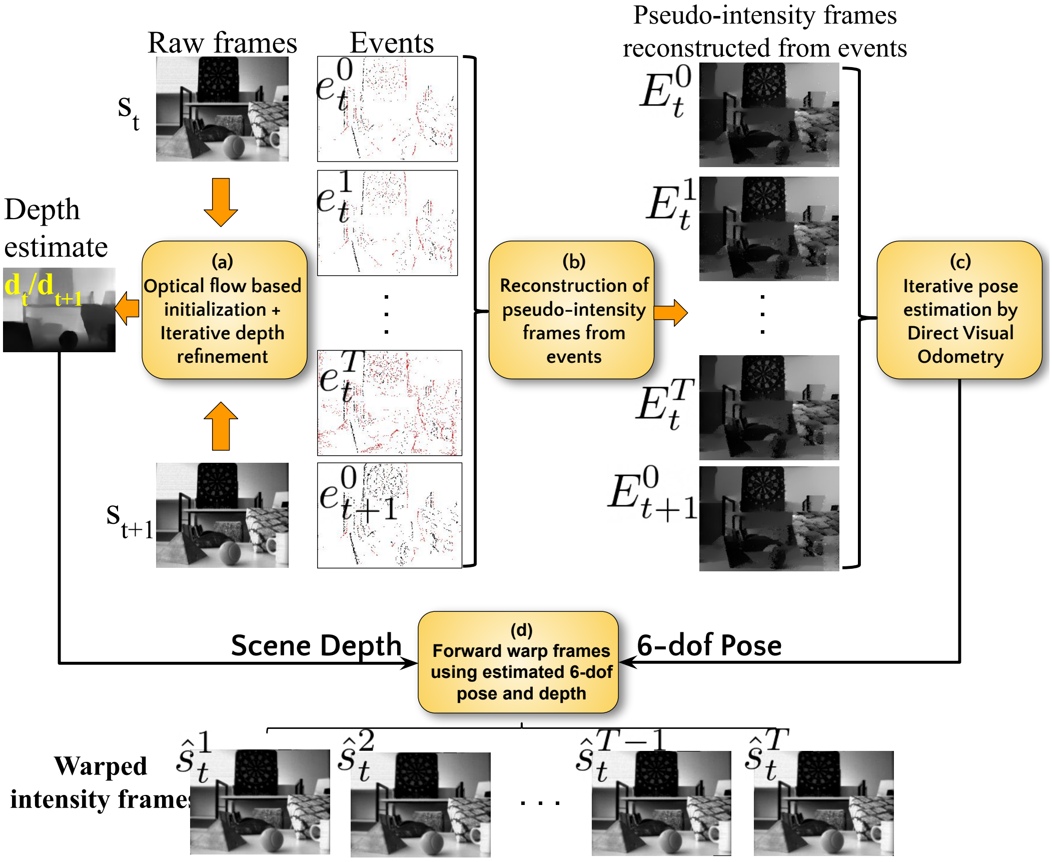

Raw frames
Events
Pseudo-intensity frames reconstructed from events
S_t
e_t^0
e_t^1
e_t^T
e_{t+1}^0
S_{t+1}
Depth estimate
d_t/d_{t+1}
(a) Optical flow based initialization + Iterative depth refinement
(b) Reconstruction of pseudo-intensity frames from events
E_t^0
E_t^1
E_t^T
E_{t+1}^0
(c) Iterative pose estimation by Direct Visual Odometry
Scene Depth
(d) Forward warp frames using estimated 6-dof pose and depth
6-dof Pose
Warped intensity frames
$\hat{s}_t^1$
$\hat{s}_t^2$
$\hat{s}_t^{T-1}$
$\hat{s}_t^T$

Fig. 4.2: Overview of the approach: The main blocks of the algorithm are a) an iterative depth and camera pose estimation technique for successive intensity frames, b) mapping event data into pseudo-intensity frames using Munda *et al.* (2018), c) direct visual odometry based sensor ego-motion estimation for intermediate event frame locations and d) a warping module for warping intensity images to intermediate locations.

- A high quality temporally dense photorealistic reconstructions is demonstrated using the proposed method on real data captured from DAVIS.
- The algorithm's robustness to abrupt camera motion and noisy events in the event sensor data is also demonstrated

4.1.1 Related work

There has been increased interest in visual odometry and simultaneous localization and mapping (SLAM) (Kim *et al.*, 2016; Rebecq *et al.*, 2016*a*; Mueggler *et al.*, 2017; Gallego *et al.*, 2018, 2019*b*), ego-motion estimation (Nguyen *et al.*, 2019; Bryner *et al.*, 2019; Zhu *et al.*, 2019), and 3D reconstruction (Kim *et al.*, 2016; Gallego *et al.*, 2018; Zhu *et al.*, 2019; Zihao Zhu *et al.*, 2018; Zhou *et al.*, 2018*b*; Andreopoulos *et al.*, 2018) with the help of event sensors. Event sensors have also shown to be performing well in mainstream vision tasks, such as image classification, corner detection etc. after reconstructing intensity images (Rebecq *et al.*, 2019*a*). We refer the reader to Gallego *et al.* (2019*a*) for further research interests in event-based vision algorithms.

Intensity image reconstruction from events The proposed work is very closely related to other previous works which reconstruct intensity images from events (Munda *et al.*, 2018; Bardow *et al.*, 2016; Barua *et al.*, 2016; Scheerlinck *et al.*, 2018*a*). Most previous works (Munda *et al.*, 2018; Bardow *et al.*, 2016; Barua *et al.*, 2016) cannot recover the true intensity information of the scene as they use only the events to estimate the intensity images. Some works (Kim *et al.*, 2016; Rebecq *et al.*, 2016*b*) reconstruct intensity images as a by-product of sensor tracking from event data over 3D scenes but are not able to recover the true intensity information. Recently, Scheerlinck *et al.* (2018*a*) demonstrated that event data and the intensity image data can be used in a complementary filter to reconstruct intensity frames at a higher frame rate. Although Scheerlinck *et al.* (2018*a*) make use of the intensity images, the reconstructed images tend to be blurry and are adversely affected by noisy events due to lack of any regularization in their proposed method. Wang *et al.* (2019*c*) proposes to use a learning based denoising algorithm to fuse event sensor and image sensor data to reconstruct images at high frame rate. Although the paper shows promising results on the synthetic data, it fails to show similar quality results on real data captured using DAVIS. Another recent work on reconstructing intensity images from event sensors is E2Vid (Rebecq *et al.*, 2019*a*). Rebecq *et al.* (2019*a*) propose a deep-learning algorithm that overcomes trailing edge issues and provide high-quality reconstructions. However, they do not achieve photorealistic reconstructions and also cannot reconstruct regions that are static with respect to the event sensor. By using raw intensity images along with events as input, our algorithm can reconstruct both static as well as dynamic regions.

4.2 Photorealistic image reconstruction

Here, a method is proposed to reconstruct photorealistic intensity images using the event stream obtained from an event sensor. The conventional image sensor will compensate for the fine texture and the absolute intensity information which is lost in the event stream. As can be seen from Fig. 4.2, the proposed algorithm has four major steps to reconstruct the temporally dense photorealistic intensity frames: (a) Estimate dense depth maps d_t and d_{t+1} corresponding to the successive intensity frames s_t and s_{t+1} and the relative pose $\xi \in \mathbb{R}^7$ between them (Sec. 4.2.1); (b) Reconstruct pseudo-intensity frames ${E_t}^j$ at uniformly spaced temporally dense locations $j = 1, 2, \ldots N$ between every successive intensity frame s_t and s_{t+1}; (c) Estimate temporally dense sensor ego-motion estimates ξ_t^j and ξ_{t+1}^j for each intermediate pseudo-intensity frame with respect to the intensity frames s_t and s_{t+1} (Sec. 4.2.2) and (d) Forward warp the intensity frames s_t and s_{t+1} to the intermediate location of each of the pseudo-intensity frames ${E_t}^j$ and blend them (Sec. 4.2.3).

The relative pose $\xi = [q_1, q_2, q_3, q_4, t_1, t_2, t_3]$ where the vector $[q_1, q_2, q_3, q_4]$ represents rotation in quaternion and the vector $[t_1, t_2, t_3]$ represents the translation. Both the quaternion and translation vectors define the three dimensional co-ordinate transformation between the camera positions in the world. The dense ego-motion estimates $\xi_t^j \in \mathbb{R}^7$ and $\xi_{t+1}^j \in \mathbb{R}^7$ are also relative poses (just like ξ) between the j^{th} pseudo-intensity frame and the raw intensity frames at t and $t + 1$ respectively. For, ξ_t^j and ξ_{t+1}^j the frame at time t and $t + 1$ respectively act as the reference world-coordinate system from which the transformation to frame j is defined.

4.2.1 Depth estimation from two successive intensity images

One of the important steps in the proposed algorithm is forward warping the intensity images to multiple intermediate temporal locations between successive intensity frames. However, forward warping can introduce undesired holes in the final reconstructed images at regions of disocclusion. This can be solved by warping both the successive intensity frames, s_t and s_{t+1}, to the intermediate locations. This requires the estimation of two dense depth maps d_t and d_{t+1} corresponding to the images s_t and

s_{t+1}, respectively. Fig. 4.3 shows the overall scheme of estimating dense depth maps from successive intensity frames. We initialize the depth estimates d_t and d_{t+1} from optical flow, and the 6-DoF camera pose ξ with zero rotation and translation. Here, ξ is the 6-DoF relative camera pose at s_{t+1} with respect to s_t. The intensity image s_{t+1} is warped to the location of s_t with the current estimate of d_t and ξ, to obtain $\hat{s}_t$. Similarly, image s_t is warped to the location of s_{t+1} to obtain $\hat{s}_{t+1}$. The photometric reconstruction loss $\mathcal{L}_{ph}$ is defined as,

$$\mathcal{L}_{ph}(d_t, d_{t+1}, \xi) = \|(\hat{s}_t - s_t)\|_1 + \|(\hat{s}_{t+1} - s_{t+1})\|_1 \tag{4.1}$$

By minimizing the above reconstruction loss, $\mathcal{L}_{ph}$, it is possible to estimate the depth maps d_t and d_{t+1} and 6-DoF relative pose ξ. An edge aware Laplacian smoothness prior is enforced on the estimated depth maps d_t and d_{t+1}, by taking inspiration from Mahjourian *et al.* (2018). The smoothness loss $\mathcal{L}_{sm}$ is defined as,

$$\mathcal{L}_{sm}\left(d_t\right) = \sum |\nabla_x d_t| \exp\left(-\beta|\nabla_x s_t|\right) + |\nabla_y d_t| \exp\left(-\beta|\nabla_y s_t|\right) \tag{4.2}$$

where I is the intensity image, d is the corresponding dense depth map and ∇_x and ∇_y are the x and y-gradient operators, respectively. Overall, the dense depth estimate d_t, d_{t+1} and the relative pose ξ is estimated by,

$$\xi, d_t, d_{t+1} = \underset{\hat{\xi}, \hat{d}_t, \hat{d}_{t+1}}{\arg\min}\, \mathcal{L}_{ph}\left(\hat{d}_t, \hat{d}_{t+1}, \hat{\xi}\right) + \lambda_{sm}\left(\mathcal{L}_{sm}\left(\hat{d}_t\right) + \mathcal{L}_{sm}\left(\hat{d}_{t+1}\right)\right) \tag{4.3}$$

Eq. (4.3) is a non-convex optimization problem and hence a good initialization of depth and pose is essential to avoid local minima. Here, we use optical flow between the successive intensity frames obtained from PWC-Net (Sun *et al.*, 2018) as an initial estimate of the depth (Heeger, 1996).

4.2.2 6-DoF relative pose estimation by direct matching

To achieve the goal of photorealistic reconstruction, the successive intensity frames captured by the image sensor are warped to the intermediate temporal location of an event frame. For warping, the 6-DoF camera pose between the temporal locations of the suc-

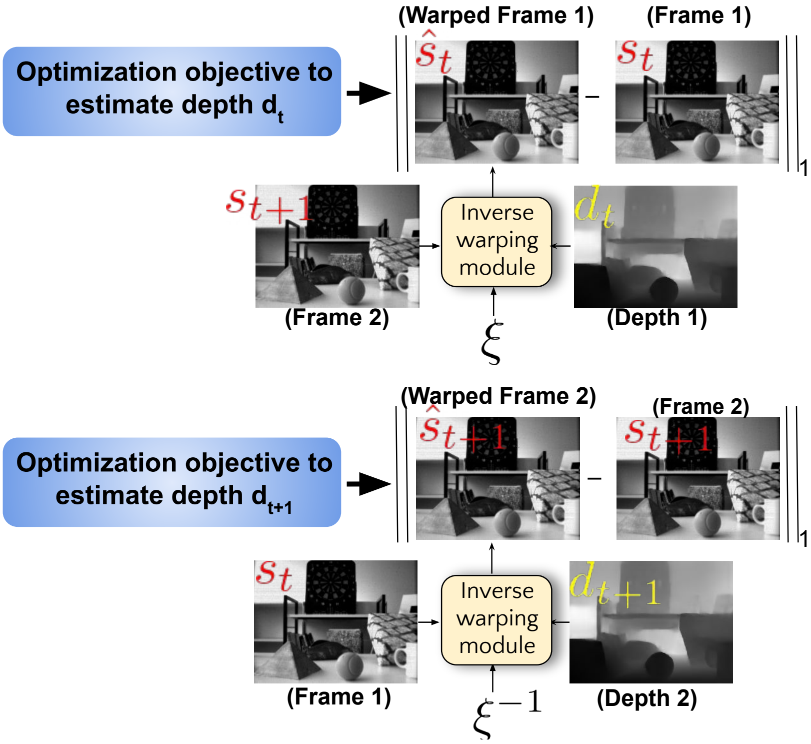


Fig. 4.3: *Estimating dense depth maps and relative pose of two successive intensity images:* The optical flow estimated from PWC-Net (Sun *et al.*, 2018) is used to obtain an initial depth estimate and initialize the relative pose to zero rotation and translation. The photometric error is iteratively minimized over the depth maps d_t and d_{t+1} and the relative pose ξ.

cessive intensity frames and that of the intermediate event frames are to be determined. We reconstruct pseudo-intensity images from events using (Munda *et al.*, 2018) at the temporal locations of the intermediate event frames as well as the successive intensity frames. A brief explanation of the algorithm used to reconstruct pseudo-intensity images is provided in the following paragraph.

The pseudo-intensity image reconstruction from events in (Munda *et al.*, 2018) is cast as a regularized integration of events. Each incoming event is added to a previously reconstructed log-intensity frame to produce an intermediate frame. This intermediate frame is then assumed to be generated from a Poisson likelihood model whose mean and variance is the final reconstructed image. Hence, the task is to estimate the mean and variance of a Poisson distribution given only a single observation. As this is an ill-posed estimation problem, the authors use a regularization term on the final reconstructed image. This regularization term is defined on a manifold of event timestamps called the surface of active events. This manifold ensures that the regularization between pixels

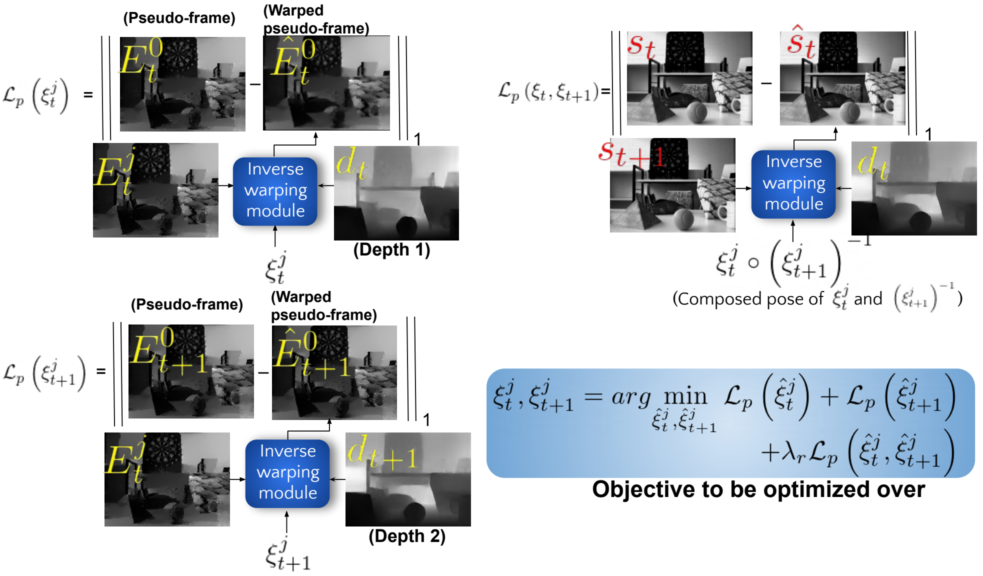

(Pseudo-frame)
(Warped pseudo-frame)
$\mathcal{L}_p\left(\xi_t^j\right)$ =
E_t^0
$\hat{E}_t^0$
1
E_t^j
Inverse warping module
d_t
(Depth 1)
ξ_t^j
(Pseudo-frame)
(Warped pseudo-frame)
$\mathcal{L}_p\left(\xi_{t+1}^j\right)$ =
E_{t+1}^0
$\hat{E}_{t+1}^0$
1
E_t^j
Inverse warping module
d_{t+1}
(Depth 2)
ξ_{t+1}^j
$\mathcal{L}_p\left(\xi_t, \xi_{t+1}\right)$=
s_t
$\hat{s}_t$
1
s_{t+1}
Inverse warping module
d_t
$\xi_t^j \circ \left(\xi_{t+1}^j\right)^{-1}$
(Composed pose of ξ_t^j and $\left(\xi_{t+1}^j\right)^{-1}$)
$\xi_t^j, \xi_{t+1}^j = arg \min_{\hat{\xi}_t^j, \hat{\xi}_{t+1}^j} \mathcal{L}_p\left(\hat{\xi}_t^j\right) + \mathcal{L}_p\left(\hat{\xi}_{t+1}^j\right) + \lambda_r \mathcal{L}_p\left(\hat{\xi}_t^j, \hat{\xi}_{t+1}^j\right)$
Objective to be optimized over

Fig. 4.4: *Estimating relative pose of intermediate pseudo-intensity images:* The 6-DoF camera pose of ${E_t}^j$ w.r.t. ${E_t}^0$ and ${E_{t+1}}^0$ is estimated by iteratively minimizing the photometric error between the warped image $\hat{E}_t^j$ and the target image ${E_t}^j$. The photometric error over the relative poses ξ_t^j and ξ_{t+1}^j is minimized using the known depth estimates d_t and d_{t+1}.

with different timestamps is reduced while pixels having similar timestamps have higher regularization.

As shown in Fig. 4.4, the objective here is to estimate the relative camera pose between ${E_t}^0$, ${E_{t+1}}^0$ and the pseudo-intensity images ${E_t}^j$ ($j = 1, 2, \ldots N$). This relative pose is used to warp the successive intensity frames to the intermediate locations specified by the event frames (${E_t}^j$) and hence reconstruct photorealistic intensity images.

Let ξ_t^j represent the 6-DoF camera pose of the intermediate pseudo-intensity image E_t^j with respect to E_t^0 and ξ_{t+1}^j be the 6-DoF camera pose of E_t^j with respect to ${E_{t+1}}^0$. The current estimate of relative camera pose ξ_t^j and the known depth estimate d_t is used to inverse warp the pseudo-intensity frame E_t^j to the location of E_t^0 to obtain $\hat{E}_t^0$. Similarly, the pseudo-intensity frame E_t^j is inverse warped to the location of E_{t+1}^0 to obtain $\hat{E}_{t+1}^0$ using the current estimate of relative pose ξ_{t+1}^j and the known depth d_{t+1}. The photometric loss $\mathcal{L}_p$ is defined as MAE between the warped intensity frame and the ground truth frame.

$$\mathcal{L}_p\left(\xi_t^j\right) = \|E_t^0 - \hat{E}_t^0\|_1 \tag{4.4}$$

$$\mathcal{L}_p\left(\xi_{t+1}^j\right) = \|E_{t+1}^0 - \hat{E}_{t+1}^0\|_1 \tag{4.5}$$

By composing the relative pose estimates, ξ_t^j and $(\xi_{t+1}^j)^{-1}$ the overall pose between s_t and s_{t+1} is obtained. This knowledge is used to regularize the relative camera pose estimates ξ_t^j and ξ_{t+1}^j with $\mathcal{L}_p(\xi_t^j, \xi_{t+1}^j) = \|s_t - \hat{s}_t\|_1$. Overall,

$$\xi_t^j, \xi_{t+1}^j = \underset{\hat{\xi}_t^j, \hat{\xi}_{t+1}^j}{\arg\min}\, \mathcal{L}_p\left(\hat{\xi}_t^j\right) + \mathcal{L}_p\left(\hat{\xi}_{t+1}^j\right) + \lambda_r \mathcal{L}_p\left(\hat{\xi}_t^j, \hat{\xi}_{t+1}^j\right) \tag{4.6}$$

where λ_r is the regularization parameter.

4.2.3 Forward Warping and Blending

At this stage, depth maps d_t and d_{t+1} are obtained corresponding to intensity images s_t and s_{t+1} respectively. A source-target mapping (forward warping) is done from two images s_t and s_{t+1} using the estimated relative pose ξ_t^j and ξ_{t+1}^j to the latent image ${s_t}^j$ and alpha-blend them. In forward warping the pixel (x, y) of the image s_t is mapped to the pixel (x', y') as follows,

$$(x', y', 1)^T = \frac{1}{h} K\,[R|p]\,K^{-1} d_t\,(x, y) \times (x, y, 1)^T \tag{4.7}$$

where K is the intrinsic matrix of the camera, R and p are the rotation, translation parameters given by ξ_t^j. Note that, while the values (x, y) lie on a regular grid, the transformed values (x', y') need not necessarily lie on the regular rectangular grid. To avoid any holes in the resultant warped image we splat the intensity values which are transformed from the source image s_t to the target image at position ${s_t}^j$. Similarly, we also transform the source image s_{t+1} to the target image position ${s_t}^j$ using the depth estimate d_{t+1} and the pose estimate ξ_{t+1}^j.

Now, there are two images warped from two different source images at a single target image location. In order to combine the two frames into a single frame, the simple technique of alpha blending (Szeliski, 2010) is used. Alpha blending performs a convex combination of two images where the parameter α determines of the weight assigned to each image. The value of α is set to 0.5 for overlapping image regions, and is 1.0 for regions where one of the images has non-zero value. The value of α linearly increases in the transition between 0.5 and 1.0.

4.3 Experiments

For all the experiments DAVIS240 (Berner *et al.*, 2013) sensor is used, which is commercially available and has a conventional image sensor and an event sensor bundled together. We used the recently proposed dataset by Mueggler *et al.* (2017) and Scheerlinck *et al.* (2018*a*) which consists of several video sequences captured using DAVIS240. Spatially dense depth maps at the locations of low frame rate intensity frames and temporally dense sensor ego-motion using the event sensor data are obtained to warp the low frame-rate intensity frames to intermediate camera locations. The proposed optimization technique estimates depth with blurred edges. To enhance the sharpness of the estimated depth maps, we use a fast bilateral solver (Barron and Poole, 2016) which takes estimated depth and the raw image as input. The output of this bilateral solver is then used as an initialization for the iterative depth refinement scheme.

Using the event stream from each sequence in the dataset, pseudo-intensity estimates are generated using the algorithm proposed in (Munda *et al.*, 2018). Non-overlapping blocks of 2000 events are stacked into a frame to generate a corresponding pseudo-intensity frame using (Munda *et al.*, 2018). These pseudo-intensity frames are then used for estimating the temporally dense sensor ego-motion.

A hyper-parameter search is conducted for different values of β, λ_{sm} and λ_r in Eqs. (4.2), (4.3), and (4.6). For β, the search was done between 1 and 20, with steps of 5, essentially searching over 5 values $\{1, 5, 10, 15, 20\}$. For λ_r, we searched over $\{0.003, 0.01, 0.03, 0.1\}$ and for λ_{sm} we searched over $\{0.1, 0.3, 1.0, 3.0\}$. The values which gave us the visually best results are used. Finally, we had $\beta = 10$, $\lambda_r = 0.01$ and $\lambda_{sm} = 1.0$. For pose estimation, $\lambda_r = 0.01$ in Eq. (4.6) is used. We use the Adam optimizer (Kingma and Ba, 2014) to solve Eq. (4.3) and Eq. (4.6).

The number of intensity frames interpolated between successive intensity frames is adaptive to the event rate produced. This is because events are binned into a frame, based on number of events instead of binning all the events occurring in a particular time interval. This is also known as Stacking By Number (SBN) in (Wang *et al.*, 2019*c*). The typical event rates in an event sensor can range from 2×10^5 to 1×10^6 events per second. As successive 2000 events are being binned into a single frame, the frame rate can range from 100 fps to 500 fps. Theoretically, this frame rate can be further increased by binning overlapping events into a frame. For the video sequences shown in this manuscript we have mentioned the frame rate of the video wherever appropriate.

Metrics	Ummenhofer *et al.* (2017) initialization		Sun *et al.* (2018) initialization	
	brown_bm_1	brown_bm_2	brown_bm_1	brown_bm_2
MAE	0.55	0.61	0.45	0.3
RMSE	0.71	0.8	0.58	0.45
$\delta < 1.25$	0.27	0.33	0.24	0.58
$\delta < 1.25^2$	0.46	0.56	0.58	0.7
$\delta < 1.25^3$	0.62	0.66	0.73	0.81

Table 4.1: Predicted depth accuracy for different depth initialization schemes. Lower values of ‘MAE’ and ‘RMSE’ are preferred while higher values of ‘$\delta < th$’ are preferred.

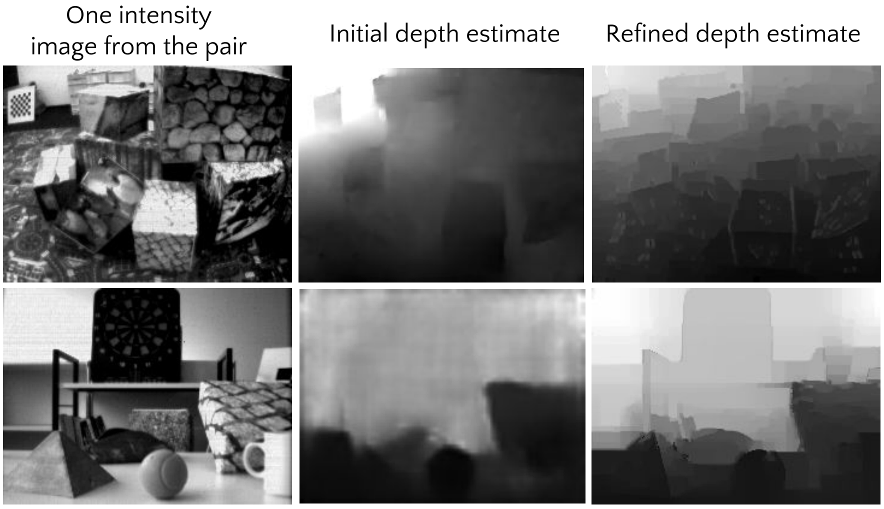


Fig. 4.5: Estimated depth maps from our proposed method

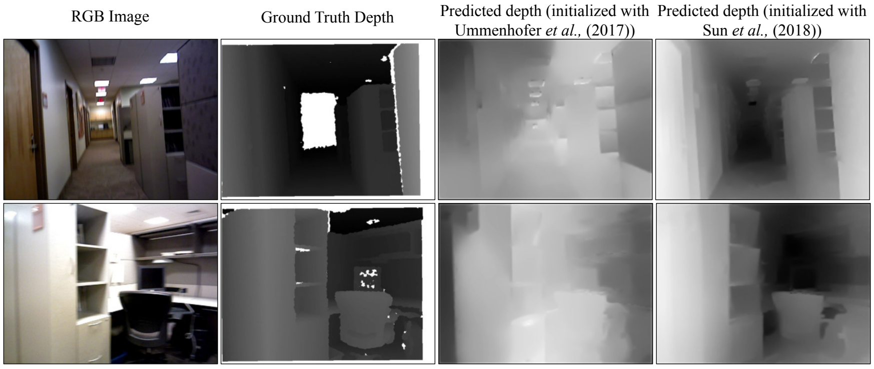

RGB Image
Ground Truth Depth
Predicted depth (initialized with Ummenhofer *et al.,* (2017))
Predicted depth (initialized with Sun *et al.,* (2018))

Fig. 4.6: Effect of two different initialization schemes on depth estimation. Each of the depth images shown above have been normalized between 0 and 1 for visualization.

4.3.1 Depth estimation

In Fig. 4.5 the effectiveness of our proposed method is demonstrated for estimating depth. An initial estimate of depth from a deep learning method is used and iteratively refined. We empirically found that using PWC-Net (Sun *et al.*, 2018) to initialize the depth estimate for the iterative optimization scheme gave consistently good results. In Table 4.1 quantitative results are provided which compares the accuracy of two different initialization schemes (Sun *et al.*, 2018; Ummenhofer *et al.*, 2017). As there are no real event-based datasets with ground truth depth maps, an RGBD dataset (Xiao *et al.*, 2013) captured using Kinect sensor is used. As the algorithm requires only RGB image sequence to compute depth map a real dataset is used instead of a synthetic one containing events. These datasets contain multiple video sequences with RGB video and corresponding depth maps obtained from the Kinect time-of-flight sensor. Two sequences ‘brown_bm_1’ and ‘brown_bm_2’ are used here for quantitative evaluation. The depth maps from this dataset contain holes, pixels where depth values are not acquired. These invalid pixels are masked while computing the error metrics. The different metrics used for quantifying depth accuracy are:

- MAE: $\frac{1}{n}\Sigma|d_i - \hat{d}_i|$
- RMSE: $\sqrt{\frac{1}{n}\Sigma(d_i - \hat{d}_i)^2}$
- Accuracy: $\%$ of d_i s.t. $max(\frac{d_i}{\hat{d}_i}, \frac{\hat{d}_i}{d_i}) = \delta < th$

From Table 4.1 it can be seen that the predicted depth is not as accurate as some of the relevant state-of-the art methods. However, there is clearly an advantage by using PWC-Net (Sun *et al.*, 2018) to initialize the depth estimate over DeMoN (Ummenhofer *et al.*, 2017).

4.3.2 Photorealistic intensity image reconstruction

In Fig. 4.1 and Fig. 4.7 the intensity images reconstructed using the proposed method and the one proposed by Munda *et al.* (2018); Scheerlinck *et al.* (2018*a*) are compared. While MR (Munda *et al.*, 2018) utilizes only event sensor data, CF (Scheerlinck *et al.*, 2018*a*) uses both event sensor data as well as information from intensity images. For fairness in comparison, intensity images are generated from (Munda *et al.*, 2018; Scheerlinck *et al.*, 2018*a*) for every 2000 events in the sequence. In (Scheerlinck *et al.*, 2018*a*), the cut-off frequency was initialized to 6.28rad/s and other parameters were updated dynamically to yield the best results. More reconstruction results obtained using the proposed technique are shown on the data captured by us in Fig. 4.8.

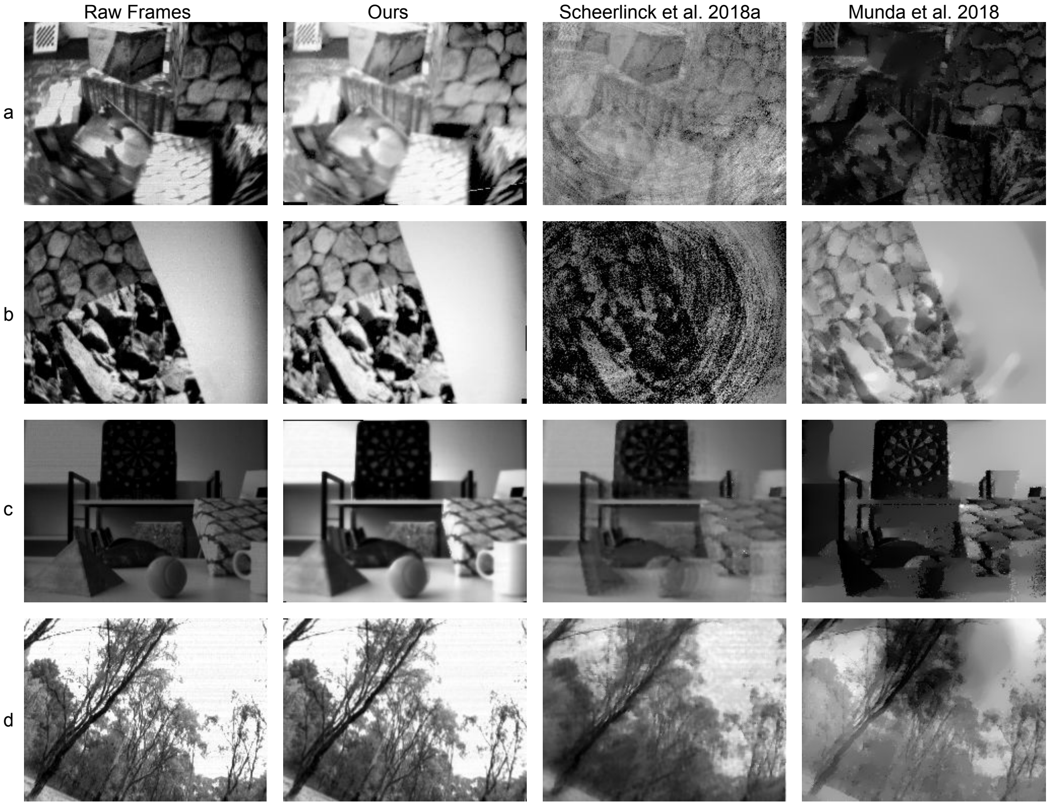


Fig. 4.7: Qualitative comparisons of reconstructions. The frame upconversion rate for the sequences from top-bottom are respectively $150\times$, $30\times$, $7\times$ and $15\times$.

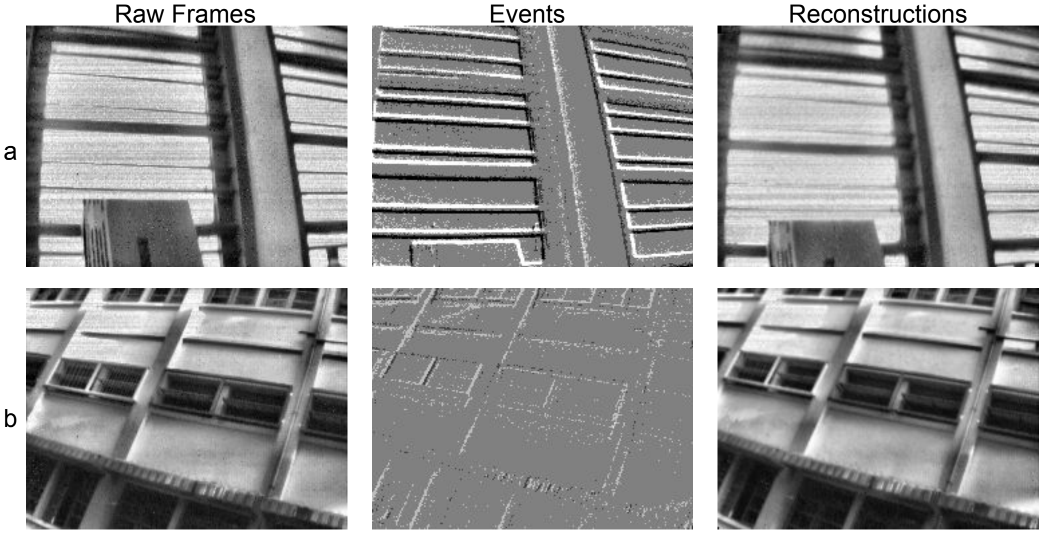


Fig. 4.8: More results obtained using the proposed technique on our data. The frame upconversion rate for the sequences from top-bottom are respectively $32\times$ and $4\times$.

PSNR is also computed for the reconstructed intermediate intensity frames and compare with the state-of-the-art algorithms in Table 4.2. Note that for real hybrid sensor data, we don't have access to the ground-truth intermediate frames. In order to overcome this, synthetic data is generated by considering every fifth frame in a video sequence (Mueggler *et al.*, 2017) as the successive intensity frames. By doing this, we now have the four ground-truth intermediate frames. It is also made sure that the successive frames in the generated synthetic sequence have enough overlap between them. The four intermediate frames are interpolated using the proposed method as well as methods proposed in (Munda *et al.*, 2018; Scheerlinck *et al.*, 2018*a*). Using the ground truth intermediate frames, the PSNR for each of the methods is computed on three different sequences. From Table 4.2, it is clear that the proposed algorithm out-performs other state-of-the-art methods. MR (Munda *et al.*, 2018) performs the worst in the metric of PSNR as it doesn't include the intensity image information. Although it is not fair to compare MR (Munda *et al.*, 2018) with methods which use intensity image information such as ours and CF (Scheerlinck *et al.*, 2018*a*), the results are included for completeness.

MR (Munda *et al.*, 2018) uses only event information and are hence unable to recover the true intensity information present in the scene. CF (Scheerlinck *et al.*, 2018*a*)

Sequence	MR (Munda *et al.*, 2018)	CF (Scheerlinck *et al.*, 2018*a*)	Ours
slider_depth	25.23	33.8	38.4
poster_6dof	26.11	34.59	39.73
boxes_6dof	23.29	31.44	36.87

Table 4.2: Comparing PSNR for different sequences

does not use any kind of spatial regularization and hence the reconstructed images are noisy and blurry even though it has access to the intensity images. Though the proposed algorithm takes more time to run compared to previous works (Scheerlinck *et al.*, 2018*a*; Munda *et al.*, 2018), it is able to produce much better results. The proposed algorithm takes about two minutes to estimate the dense depth maps and about 40 seconds to render each intermediate frame. However, with recent advances in stereo depth estimation methods, it is expected that in future, the need for an iterative depth refinement scheme can be eliminated and the output of a state-of-the-art stereo depth estimation algorithm can be directly used. This will greatly reduce the computation time.

4.3.3 Robustness to abrupt camera motion

In the case of abrupt motion of the sensor, the intensity images get blurred and the rate at which events are generated becomes high. In this case the intensity images are deblurred using an existing deblurring technique (in our experiments we used (Nah *et al.*, 2017)). These deblurred images are then used as an input to the reconstruction pipeline. Abrupt motion results in a high event rate and also produces many noisy events. These noisy events affect the reconstructions in (Scheerlinck *et al.*, 2018*a*) as their trust on events increases exponentially with the rise in the event rate. As can be seen in Fig. 4.9, our method is robust to such abrupt motions as can be seen from the results shown in columns (b) and (c).

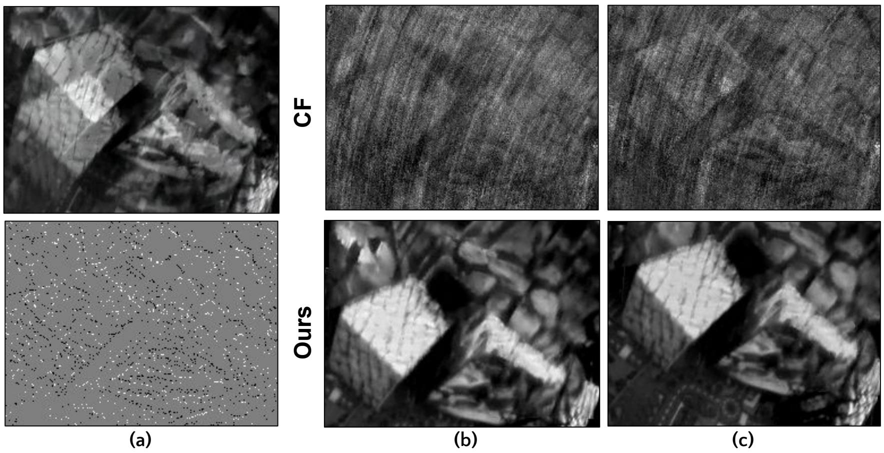


Fig. 4.9: *Abrupt camera motion:* (a)Top row shows the two successive images blended into one where we can see the abrupt camera motion. The bottom row shows an intermediate event frame affected by noise. In (b) and (c) two intermediate reconstructed frames using CF (Scheerlinck *et al.*, 2018*a*) and our proposed algorithm are shown. We can clearly see that the proposed method performs much better even during abrupt camera motions.

4.4 Conclusion

The strength of texture-rich low frame rate intensity frames is combined with high temporal rate event data to obtain temporally dense photo-realistic images. This is achieved by warping the low frame rate intensity frames from the conventional image sensor to intermediate locations. With extensive experiments, it has been demonstrated that the images reconstructed from the proposed algorithm are photorealistic compared to any of the previous methods. The robustness of our algorithm to abrupt camera motion has also been shown. Currently, the proposed algorithm assumes a static scene. A future direction would be to build a generalized algorithm which can reconstruct photorealistic images for dynamic scenes as well.

CHAPTER 5

High Frame-rate and High Dynamic Range Intensity Reconstruction from Event Sensors

5.1 Introduction

Event-based sensors are a novel generation of neuromorphic sensors which asynchronously sense only the pixel-level brightness changes with a temporal resolution of the order of microseconds (Delbrück *et al.*, 2010). Chapter 4 proposed a hybrid setup of event and intensity sensor for photorealistic high frame-rate video reconstruction. The use of additional intensity sensor proved helpful in overcoming challenges with event-based image reconstruction such as trailing edges, sensitivity to event-noise, *etc.* In Chapter 4, event sensor data is used to estimate only the dense camera motion in the form of 6-DoF relative pose estimate. The input images from the intensity sensor are then warped to the temporally dense location of events producing a high frame-rate video. However, this also results in the reconstructed video to have the same low dynamic range as that of the input video from the intensity sensor. The high dynamic range nature of the event-based sensor output is not used effectively for video reconstruction. Hence, in this chapter we discuss a technique for reconstruction of high dynamic range and high frame-rate video from event-based sensors.

In Chapter 4, the use of additional intensity sensor was justified because of the trailing edge artifacts (Bardow *et al.*, 2016; Reinbacher *et al.*, 2016*a*) and event-noise sensitivity (Reinbacher *et al.*, 2016*a*) in videos generally reconstructed from only event-sensor data. However, Rebecq *et al.* (2019*b*,*a*) showed promising results for video reconstruction using only the event sensor data. The reconstructed high frame-rate videos had a high dynamic range and did not show any of the artifacts present in traditional techniques (Bardow *et al.*, 2016; Reinbacher *et al.*, 2016*a*). This was achieved by training a deep neural network on a large number of pairs of event-sensor data and corresponding intensity frames. As the event sensor data was synthetically generated, these

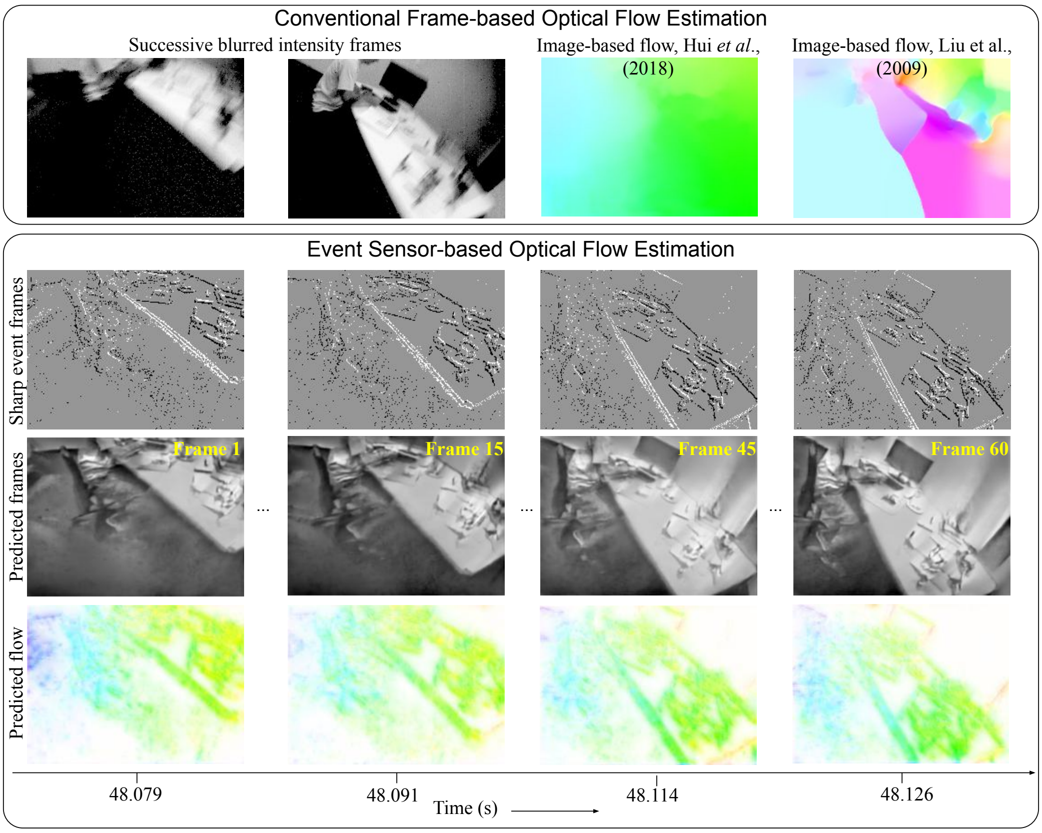

Conventional Frame-based Optical Flow Estimation
Successive blurred intensity frames
Image-based flow, Hui *et al.*, (2018)
Image-based flow, Liu et al., (2009)
Event Sensor-based Optical Flow Estimation
Sharp event frames
Predicted frames
Predicted flow
Frame 1
Frame 15
Frame 45
Frame 60
...
48.079
48.091
48.114
48.126
Time (s)

Fig. 5.1: Conventional frame-based optical flow algorithms suffer when the input images are degraded with motion blur as shown in the top row. Event sensors on the other hand operate at much higher temporal resolution and can sense much higher dynamic range than the frame-based sensors. We accumulate the events triggered between the two successive intensity images as event frames and show some of them in the second row. Our proposed algorithm takes these intermediate event frames as input and predicts corresponding intensity images and optical flow. In this example, optical flow and intensity images are predicted at 60 intermediate temporal locations corresponding to a 60× temporal super-resolution.

techniques were sensitive to those scenes where event sensor noise became dominant. The synthetic event generation pipeline could not realistically simulate noise for every possible real-world scenario. Rebecq *et al.* (2019*a*,*b*) also relied on hard-to-acquire ground truth optical flow data to impose temporal consistency between successive predicted intensity frames. To overcome these drawbacks, we propose a semi-supervised learning-based technique to predict high frame-rate and high dynamic range videos and optical flow simultaneously from event sensor data.

In our proposed method, intensity frames and a sparse optical flow are simultaneously predicted from the input event sensor data. The event sensor data is first converted to a series of event frames by stacking a fixed number of events per frame following the stacking by number (SBN) principle of (Wang *et al.*, 2019*c*). A sequence of event frames are given as input one-by-one to the neural network which predicts the corresponding intensity frame and optical flow. The intensity frame prediction is supervised using the temporally sparse ground truth intensity frames. While our proposed algorithm predicts intensity frame at a very high temporal resolution (at the rate of incoming events) the intensity frames acquired from hybrid intensity and event based sensors (Brandli *et al.*, 2014) are at a much lower temporal resolution. Thus, it is not possible for us to have a supervised loss for every predicted intensity frame. We overcome this challenge by using recurrent neural network architecture that makes it possible to use supervision only at a few time-steps by sharing weights across all the time-steps. Recurrent neural networks have already been used in (Rebecq *et al.*, 2019*b*) to predict high frame rate intensity frames. We adapt this network to simultaneously predict intensity frames and optical flow. As demonstrated for optical flow prediction from conventional image sensors (Jason *et al.*, 2016; Ren *et al.*, 2017; Meister *et al.*, 2018), we use the brightness constancy constraint as a supervisory signal for optical flow prediction from event sensors.

In summary, we make the following contributions:

- We propose a semi-supervised learning algorithm to predict high frame-rate and high dynamic range videos from event-based sensors. The algorithm also simultaneously predicts temporally dense and spatially sparse optical flow from the input.
- Optical flow prediction is self-supervised using the high frame rate and high dynamic range intensity frames predicted directly from the event sensor data. Thus, ground truth optical flow is not necessary for training our proposed algorithm.
- We also demonstrate the generalizability of our proposed algorithm on a wide variety of open source event datasets captured with different sensors and in different environments.

5.1.1 Related Work

Motion estimation from event sensors Although it's a challenging task to estimate optical flow from event sensors, several algorithms have been proposed (Liu and Del-

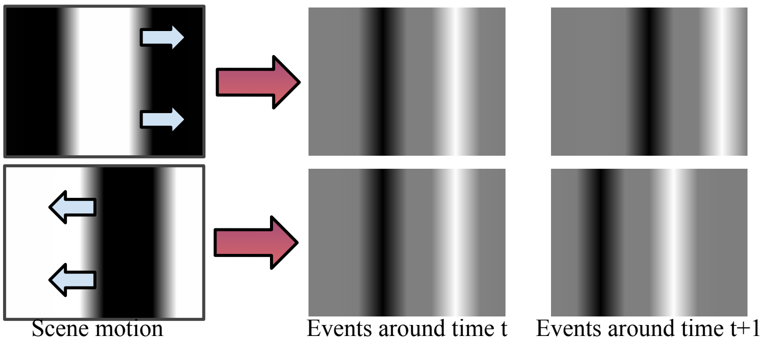


Fig. 5.2: Ambiguity in intensity image prediction from a single event frame. The first column shows two different scenes which have opposite motion with respect to the camera. These two scenes produce the same event frame at time t making it ambiguous to predict the corresponding scene intensity from the single event frame. However, when we consider the next event frame at time $t + 1$, we clearly see the motion in the scene. Modeling this temporal information using recurrent neural network helps in predicting the intensity frames unambiguously from event data alone.

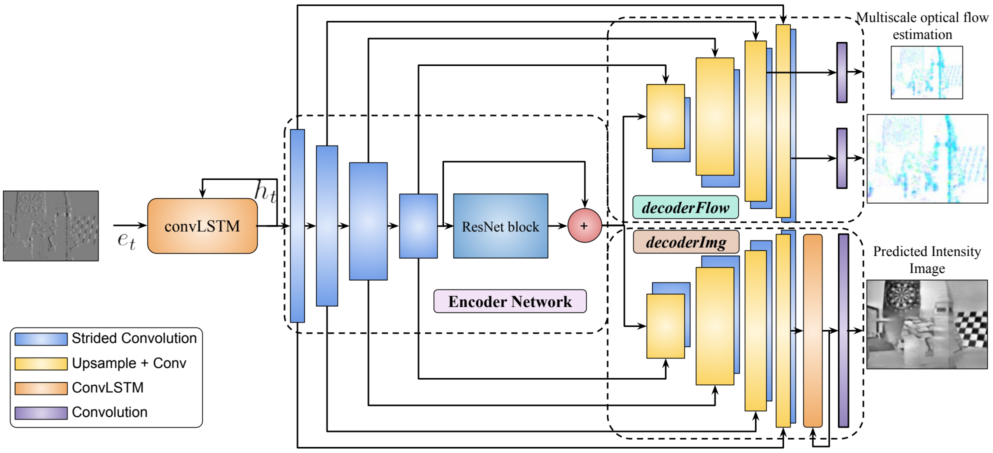


Fig. 5.3: Overall flow of our proposed method: Our proposed methods takes in a single event frame at each time-step, which is then input to a ConvLSTM network. The updated hidden state from the ConvLSTM network is input to an encoder network consisting of four strided convolutional layers followed by a ResNet block. The hidden representation from the encoder network is then fed as input to two decoder networks, *decoderImg* and *decoderFlow*, which predict the intensity image and the optical flow, respectively.

bruck, 2018; Nagata *et al.*, 2019; Paredes-Vallés *et al.*, 2019; Khoei *et al.*, 2019; Bardow *et al.*, 2016; Zhu *et al.*, 2018*a*, 2019, 2018*c*; Haessig *et al.*, 2018; Gallego *et al.*, 2018). Works such as (Gallego *et al.*, 2018; Zhu *et al.*, 2018*c*, 2019) use motion compensation on the space-time volume of events to estimate optical flow. In (Haessig *et al.*, 2018), the authors design a spiking neural network to estimate optical flow and demonstrate their proposed algorithm on IBM's neuromorphic chip. A few learning based methods have also been proposed for estimating optical flow from event sensors (Zhu *et al.*, 2019, 2018*a*).

Intensity image reconstruction Previously researchers have attempted to estimate intensity frames from event sensor data (Reinbacher *et al.*, 2016*a*; Scheerlinck *et al.*, 2018*b*; Bardow *et al.*, 2016; Shedligeri and Mitra, 2018; Rebecq *et al.*, 2019*a*; Wang *et al.*, 2019*c*), so that the intensity frames could be used as an input to off-the-shelf frame-based computer vision algorithms. Recent learning based algorithms (Rebecq *et al.*, 2019*a*; Wang *et al.*, 2019*c*) have shown a great improvement in reconstructed intensity image quality compared to traditional methods. The closest work to ours is (Bardow *et al.*, 2016), where the authors propose a framework to simultaneously estimate intensity and optical flow directly from the event sensor data. Bardow *et al.* (2016) use a sliding window approach on incoming event sensor data where a handcrafted variational loss function is defined on each event fired from the sensor. This can lead to noisy estimates as the algorithm does not differentiate between noisy and actual event data. The algorithm also relies on the approximate forward model relating true intensity values and the generated events. A noisy estimate of intensity frames affects the optical flow estimation, as Bardow *et al.* (2016) rely on the intensity images for this task. However, in our approach we make use of ground-truth raw intensity frames from a conventional image sensor to train a deep neural network for estimating clean intensity frames. This leads to more accurate optical flow estimation as shown in the experiments section.

5.2 Optical Flow Estimation from Event Sensors

5.2.1 Modeling events as sequential data

The output of an event sensor is a 4-tuple (x, y, t, p) where x and y represent the spatial location, t represents the time instant and p denotes the polarity ($+1$ or -1) of the triggered event. Following (Wang *et al.*, 2019*c*), we stack these events into a sequence of event frames to form the input to our algorithm. The temporal information is obviously lost due to this projection of spatio-temporal data as a spatial frame. In Fig. 5.2, we show a toy example where two different video sequences are used to generate an event frame at time t. Both the event frames look identical as they lack any temporal information about the events, leading to ambiguity in prediction of intensity frames.

To tackle this loss of temporal information we use a sequence of event frames akin to a sequence of image frames forming a temporal video. The effectiveness of this simple representation can be seen from Fig. 5.2 where a clear distinction emerges between the two cases of scene motion when considering a video sequence instead of looking at each frame independently. It's imperative for us to design a neural network that can effectively incorporate this temporal information so as to unambiguously predict the intensity images. Long Short-Term Memory (LSTM) (Gers *et al.*, 1999) networks have been shown to be effective for such tasks and we use them to model the long-term temporal dependency in the sequence of event frames. Although the input to the algorithm at each timestep is a single event frame, the intensity frame is still unambiguously predicted, demonstrating the effectiveness of the proposed LSTM network to model sequential information.

5.2.2 Joint estimation of intensity image and optical flow

Fig. 5.3 shows our overall algorithm to predict the intensity frames and optical flow from input event sensor data. The intensity frame prediction is supervised using temporally sparse raw intensity images acquired from the conventional image sensor present in DAVIS (Brandli *et al.*, 2014). DAVIS is a hybrid sensor consisting of co-located intensity and event based sensors. The input frames are formed by accumulating events occurring in T non-overlapping sub-intervals between successive intensity frames. Each of these sub-intervals contain a fixed, predetermined number of events. These T event frames are given as input and at the output we obtain the T intensity frames and corresponding $T - 1$ optical flow estimates. In the following sections we elaborate on the training algorithm for intensity and the optical flow estimation.

Intensity image prediction

We obtain the dataset to train our network from a hybrid intensity and event based sensor where the event data and intensity images are perfectly registered. Such hybrid sensors can acquire intensity frames at the rate of $25 - 30$ fps and the event data at the temporal resolution of the order of microseconds. We first elaborate the process of predicting and supervising intensity image prediction considering two arbitrary intensity frames s_t and s_{t+1} and the T event frames between them. This process can be generalized to any number of successive raw intensity frames from a given video sequence.

For ease of training, we divide the interval between s_t and s_{t+1} into T sub-intervals based on equal time, instead of equal number of events in each interval. While training, we use the SBT strategy and while testing, we use the SBN (Wang *et al.*, 2019*c*) strategy for creating event frames. The events occurring in each of these sub-intervals are accumulated into separate event frames forming T event frames. At each time-step, the ConvLSTM network named *inLSTM* takes one event frame as input and updates its hidden state $\mathbf{h}_t$, as shown in Fig. 5.3. This hidden state $\mathbf{h}_t$ is then fed to an encoder network which outputs a hidden representation ϕ_e. The hidden representation is then fed to a decoder network, *decoderImg*, which outputs the intensity image corresponding to the event frame at time-step t. We denote the T intermediate frames predicted between raw frames s_t and s_{t+1} as $\hat{s}_t^1, \hat{s}_t^2, \hat{s}_t^3 \dots \hat{s}_t^T$. As we have obtained T event frames between two successive intensity frames s_t and s_{t+1}, we can have supervision for only one of those T predicted frames. Due to the way we have formed event frames only the T^{th} interval has the corresponding ground truth intensity frame, s_{t+1}, for supervision. Hence the network can be supervised for intensity image prediction at every T time-steps only. As the proposed recurrent network shares weights at each time-step, the network is able to predict intensity frames without being supervised at every time-step.

We supervise the intensity image predicted at T^{th} interval s_t^T with the loss $\mathcal{L}_{im}$ defined as,

$$\mathcal{L}_{im}(\hat{s}_t^T) = dist\left(\hat{s}_t^T, s_{t+1}\right) \tag{5.1}$$

where $dist(\cdot)$ is an appropriate distance metric. $L1$ distance metric has been popularly used in supervising learning based methods due to their ability to preserve edge sharpness. This distance metric is unsuitable for our problem as the event sensor data has lost the absolute scene intensity information. So, by using a naive $L1$ metric, we are penalizing the network for not predicting something that it theoretically cannot predict with just events as input. To reflect this knowledge, we define our distance metric as,

$$dist(\hat{s}_t^T, s_{t+1}) = \frac{1}{Z} \sum \| \left(\nabla_x \hat{s}_t^T - \nabla_x s_{t+1} \right) \odot b \|_2 + \| \left(\nabla_y \hat{s}_t^T - \nabla_y s_{t+1} \right) \odot b \|_2 \quad (5.2)$$

where Z is the total number of pixels, ∇_x and ∇_y respectively are x and y-gradient operators. The gradient operator ∇ cancels out any absolute scene intensity information at each pixel of the image. We use a binary mask b which masks the saturated and low-intensity noisy image regions and is defined as,

$$b = \begin{cases} 1, & 50 < I < 200 \\ 0, & \text{otherwise} \end{cases}, \quad (5.3)$$

where the image intensity $I \in [0, 255]$. We also do not penalize the network at saturated or the low-intensity noisy regions as the dynamic range of the intensity images is much lower than that of the event sensor data. We later show the effect of using the naive $L1$ loss as a distance metric on the performance of intensity frame and optical flow prediction.

Optical flow prediction

To predict the optical flow between the current and the previous time-steps, we feed the hidden representation ϕ_e obtained at the current time-step to the decoder network, *decoderFlow*. For image-based optical flow estimation, obtaining ground truth optical flow for a real dataset is a challenging task. Hence, several self-supervised learning-based methods for optical flow estimation have been proposed (Jason *et al.*, 2016; Ren *et al.*, 2017; Meister *et al.*, 2018). We make use of these techniques to supervise optical flow prediction with the help of the intensity images predicted from the event sensor data. We define our self-supervised loss for optical flow as,

$$\mathcal{L}_{flow}\left(\hat{\mathbf{o}}_t^{j,\pi}\right) = \sum_{j=2}^{T} \|\hat{s}_t^j - \mathcal{W}\left(\hat{s}_t^{j-1}; \hat{\mathbf{o}}_t^{j,\pi}\right)\|_1 \tag{5.4}$$

where $\hat{\mathbf{o}}_t^{j,\pi}$ is the predicted optical flow at time-step j and $\hat{s}_t^j, \hat{s}_t^{j-1}$ are respectively the predicted intensity images at timestep $j, j-1$. The superscript $^\pi$ in $\hat{\mathbf{o}}_t^{j,\pi}$ denotes the scale of the predicted optical flow. To overcome gradient locality (Godard *et al.*, 2019*a*; Zhou *et al.*, 2017) of the bilinear sampler during image warping, optical flow is predicted at 2 different scales as can be seen in Fig. 5.3. Following (Godard *et al.*, 2019*a*), the optical flow at coarser scales is upsampled to the resolution of predicted intensity frame and the cost function in Eq. (5.4) is imposed. The final loss is the sum of costs at individual scales.

Overall cost function

Apart from $\mathcal{L}_{flow}$ and $\mathcal{L}_{im}$, we also impose the piece-wise smoothness constraint on the predicted intensity images and the optical flow as

$$\mathcal{L}_{im\_sm} = \frac{1}{Z}\|\nabla_x \hat{s}_t\|_2 + \|\nabla_y \hat{s}_t\|_2 \tag{5.5}$$

$$\mathcal{L}_{flow\_sm}(\hat{\mathbf{o}}_t^{j,\pi}) = \frac{1}{Z}\sum_{t=1}^{T} \|\nabla_x \hat{\mathbf{o}}_t^{j,\pi}\|_2 + \|\nabla_y \hat{\mathbf{o}}_t^{j,\pi}\|_2 \tag{5.6}$$

Overall, our training loss becomes,

$$\mathcal{L} = \sum_t \left(\lambda_1 \mathcal{L}_{im} + \lambda_2 \sum_{\pi=1}^{2} \mathcal{L}_{flow}(\hat{\mathbf{o}}_t^{j,\pi}) + \lambda_3 \mathcal{L}_{im\_sm} + \lambda_4 \sum_{s=1}^{2} \mathcal{L}_{flow\_sm}(\hat{\mathbf{o}}_t^{j,\pi})\right) \tag{5.7}$$

where λ_i with $i = 1, 2, 3, 4$ are hyperparameters which weigh each of the loss terms for optimal performance. In the second and fourth term $\pi = 1, 2$ represents the coarse and fine scale of the predicted optical flow. The optical flow at coarser scale is first upsampled to the resolution of the predicted intensity image before applying the loss function.

5.3 Architectural and implementation details

5.3.1 Architectural details

As shown in Fig. 5.3 our proposed model consists of 4 major components, a LSTM network named *inLSTM*, an encoder network and two decoder networks named *decoderImg* and *decoderFlow*. The detailed description of architecture is shown in Table 5.1. The ConvLSTM network, *inLSTM*, consists of three 2D convolutional layers and has a hidden and cell state of size 32 channels. The ConvLSTM network used at the output of the *decoderImg* has the same architecture as inLSTM. The inLSTM network is then followed by an encoder network and a ResNet block (He *et al.*, 2016) as described in Table 5.1. The ResNet block is then followed by two decoder networks *decoderImg* and *decoderFlow*. Both the decoder networks mirror the encoder network with 4 convolutional layers. Each of the convolutional layers in the decoder block are preceded by a bilinear upsampling layer that upsamples the feature maps by a factor of 2. As shown in Fig. 5.3, the network also consists of skip connections between the encoder and the decoder networks, much like a U-Net (Ronneberger *et al.*, 2015). The decoder network *decoderImg* outputs an intensity image at the same spatial resolution as the input event frame. We use the *decoderFlow* network to predict optical flow at 2 scales, as shown in Fig. 5.3. The feature maps from the final 2 layers of *decoderFlow* are input to separate 2D convolutional layers to predict the optical flow at 2 scales.

5.3.2 Dataset

To train our proposed algorithm we use the real-world dataset proposed by Mueggler *et al.* (2017). This dataset consists of multiple video sequences collected from a commercially available hybrid intensity and event-based sensor named DAVIS240 (Brandli *et al.*, 2014). Each of the video sequences are approximately 60s in length and contain raw image sensor data acquired at about $20 - 25$ fps and the event sensor data acquired from the event sensor. Both the intensity and the event data have a spatial resolution of 180×240. The video sequences are collected in variety of environments such as indoor, outdoor, planar scenes and with varying motion patterns such as pure rotation, pure translation, 6-DoF rapid motion, *etc.* We used 20 different sequences from the dataset where 13 sequences were used for training and the rest for testing. The training sequences were *boxes_6dof, boxes_translation, boxes_rotation, office_spiral, office_zigzag, outdoors_running, outdoors_walking, poster_6dof, poster_rotation, poster_translation, shapes_6dof, shapes_translation, shapes_rotation*. The validation and testing sequences were *dynamic_6dof, dynamic_translation, dynamic_rotation, slider_depth, slider_close*. Video sequences that are captured in similar environ-

Block	Layer	IC	OC	K	S	P	Remarks
inLSTM	conv2d	1	64	3	1	1	tanh
	conv2d	64	64	3	1	1	tanh
	conv2d	64	128	3	1	1	tanh; hidden size of 32
Encoder	conv2d	32	32	3	2	1	ReLU
	conv2d	32	64	3	2	1	ReLU
	conv2d	64	128	3	2	1	ReLU
	conv2d	128	256	3	2	1	ReLU
ResBlock	conv2d	256	128	3	1	1	ReLU
	conv2d	128	128	3	1	1	ReLU
	conv2d	128	256	3	1	1	ReLU
decoderFlow	conv2d	256	128	3	1	1	ReLU
	conv2d	256	64	3	1	1	skip connection; upsampling; ReLU
	conv2d	128	32	3	1	1	skip connection; upsampling; ReLU
	conv2d	64	32	3	1	1	skip connection; upsampling
Coarse flow	conv2d	32	2	3	1	1	tanh
Fine flow	conv2d	32	2	3	1	1	tanh
decoderImg	conv2d	256	128	3	1	1	ReLU
	conv2d	256	64	3	1	1	skip connection; upsampling; ReLU
	conv2d	128	32	3	1	1	skip connection; upsampling; ReLU
	conv2d	64	32	3	1	1	skip connection; upsampling; ReLU
outLSTM	conv2d	32	64	3	1	1	tanh
	conv2d	64	64	3	1	1	tanh
	conv2d	64	128	3	1	1	tanh; hidden size of 32
Out layer	conv2d	32	1	3	1	1	sigmoid; image output

Table 5.1: The number and type of layers constituting each block of the network is shown. We only use 2D convolutional layers with input channels *IC*, output channels *OC*, kernel size *K*, stride *S* and a padding of *P* pixels to the input. Non-linearity used for each layer has also been shown. The decoder blocks consist of bilinear upsampling layer and skip connections from the corresponding encoder block. More details are provided in Sec. 5.3.1 of the manuscript.

ments were put in either the training set or the test set, but not both. E.g. sequences such as *boxes* or *dynamic* appear either in the training data or in the test data, but not in both.

We quantitatively evaluate our proposed optical flow algorithm with the ground truth optical flow available in the dataset proposed by Zhu *et al.* (2018*b*). This dataset, also known as *MVSEC* dataset (Zhu *et al.*, 2018*b*), consists of event sequences captured using a commercially available hybrid sensor named DAVIS346. This sensor has a slightly higher spatial resolution of 260×346 than DAVIS240. The dataset also contains ground truth depth maps captured using a Light Detection and Ranging (LiDAR) and the relative 6-DoF pose captured using a motion-capture system. For quantitative evaluation, the ground truth optical flow is also provided in the dataset. The ground truth optical flow is actually computed using the ground truth depth maps and the relative 6-DoF camera pose (Zhu *et al.*, 2018*a*) under the assumption of a static scene. As the authors assume static scene to compute ground truth optical flow, we exclude the *outdoor_driving* sequence from optical flow evaluation and only use the *indoor_flying* sequences. This is because the *outdoor_driving* sequence has many moving objects such as cars, pedestrians, etc. where the static scene assumption does not hold and hence affects the ground truth optical flow computation. This in turn affects the evaluation of the optical flow prediction accuracy. For a fair evaluation, we follow the procedure defined in (Zhu *et al.*, 2018*a*) to compute metrics for optical flow evaluation by computing the error only at pixels where an event has fired.

5.3.3 Implementation details

To train our network we used the dataset proposed in (Mueggler *et al.*, 2017). For the quantitative evaluation of the predicted optical flow, we use the *MVSEC* dataset (Zhu *et al.*, 2018*b*) which provides ground truth optical flow for event sensors. To further demonstrate the generalizability of our proposed algorithm, we also provide results on various event sensor datasets such as (Scheerlinck *et al.*, 2018*b*; Zhu *et al.*, 2018*b*; Mueggler *et al.*, 2017; Perot *et al.*, 2020).

In our proposed algorithm both the intensity frames and the optical flow are jointly predicted and the predicted optical flow is used to impose the temporal consistency in the predicted intensity frames. Another strategy would be to first learn a neural network to predict the intensity frames from event frames. These intensity frames will then be used to learn another neural network with self-supervised warping loss (Jason *et al.*, 2016) to predict optical flow from the input event frames. Although, in theory, this strategy sounds better than our proposed method, it has a practical problem. The predicted intensity frames will not be temporally consistent and can lead to errors in the optical flow prediction. This has been shown in (Rebecq *et al.*, 2019*a*), where the authors use pre-computed optical flow to explicitly impose temporal consistency loss between successive predicted intensity frames during training. Without explicitly imposing the temporal consistency during training, the predicted intensity frames are temporally inconsistent and affect the optical flow prediction accuracy. In the initial stages of learning, the output of the image prediction network in our algorithm is completely random and this may affect our joint training strategy. Hence, we give a head-start to the image prediction network by computing only the supervised intensity loss. In practice, we found that a head-start of 1000 iterations for the image prediction network was enough.

In the training phase, the time interval between the successive raw image frames is uniformly divided into 5 equal time intervals and the corresponding events are accumulated into 5 event frames. We form our training set with such pairs of 5 event frames and the corresponding raw image frames. During training, we use 40 event frames and correspondingly 8 raw image frames of one video (all in a sequence) and input to our algorithm as one instance of the batch. The neural network is trained using our overall cost-function described in Eq. (5.7). The brightness constancy loss specified in Eq. (5.4) is applicable at all 40 time-steps. But, the intensity supervision specified in Eq. (5.1), is applicable only at 8 time-steps of the sequence.

For training our network, we use Adam optimizer (Kingma and Ba, 2014) with a learning rate of 1×10^{-4} which was decayed by a factor of 0.95 every 10k iterations. The hyperparamter in Eq. (5.7) were set to be $\lambda_1 = 1$, $\lambda_2 = 0.1$, $\lambda_3 = 0.01$ and $\lambda_4 = 0.001$. The neural network is trained for 150k iterations with a batch size of 1. While testing, we accumulate a fixed number of events per event frame, which is akin to the SBN framework proposed in (Wang *et al.*, 2019*c*). Accumulating the event frames using the SBN principle has the advantage of frame rate being adaptive to the event rate which corresponds to the amount of motion in the scene.

5.3.4 Hyperparameter selection

The hyperparameters in our training are chosen empirically to be $\lambda_1 = 1$, $\lambda_2 = 0.1$, $\lambda_3 = 0.01$ and $\lambda_4 = 0.001$. The value of $\lambda_1 = 1.0$ was chosen as a reference value. The parameter $\lambda_2 = 0.1$, however, should be chosen carefully due to the nature of our proposed algorithm. Note that λ_1 weighs the importance of the intensity loss, $\mathcal{L}_{im}$ in Eq. (5.1) while λ_2 weighs the importance of the self-supervised optical flow loss in Eq. (5.4). A large value of λ_2 will try to make the predicted intensity frames as smooth as possible and sometimes predict an uniform intensity over the whole frame. However, a very small value of λ_2 will not learn to predict the optical flow accurately. Hence, to choose λ_2 we uniformly sampled values in $[0.05, 0.5]$ and found that the value of $\lambda_2 = 0.1$ worked best in terms of optical flow accuracy. The hyperparamters λ_3 and λ_4 impose spatial smoothness over the predicted intensity frame and optical flow respectively. The hyperparameters were chosen empirically to not impose too much or too little smoothness based on qualitative observations of the network predictions. Note that $\lambda_4 < \lambda_3$ as λ_3 imposes smoothness on the predicted intensity image and is imposed only once for every 5 timesteps.

5.4 Experiments

5.4.1 Qualitative comparison on intensity image prediction

Intensity image prediction from event sensor data has been investigated by many researchers in the past few years. In this work we predict intensity images in order to facilitate the learning of optical flow directly from event sensors. However, in order to predict a good estimate of the optical flow, the predicted intensity frames should be temporally consistent, have high dynamic range and be free of any noise or other artifacts.

In Fig. 5.4 we provide qualitative comparison of the intensity images from our proposed method with that of the other state-of-the-art intensity image-only estimation algorithms such as (Scheerlinck *et al.*, 2018*b*; Rebecq *et al.*, 2019*a*; Reinbacher *et al.*, 2016*a*). Images predicted from (Reinbacher *et al.*, 2016*a*) contains various artifacts such as trailing edges as it relies on the hand-crafted image prior based on event manifolds. The algorithm proposed in (Scheerlinck *et al.*, 2018*b*) is sensitive to noisy events and can be seen producing noisy intensity estimates when a large number of events are being triggered in the scene. This sensitivity arises from the lack of any spatial regularization in the complementary filter model proposed in (Scheerlinck *et al.*, 2018*b*). In (Rebecq *et al.*, 2019*a*), the authors propose a learning based method to predict intensity images from event sensor data. The method proposed in (Rebecq *et al.*, 2019*a*) demonstrate high quality intensity image prediction from event sensor data. In contrast to MR (Reinbacher *et al.*, 2016*a*) and CF (Scheerlinck *et al.*, 2018*b*) which produce images with trailing edge artifacts, our reconstructions are smooth and are free of most of the artifacts. As seen in Fig. 5.4, the predicted intensity frames from our model are comparable to the state of the art, learning-based, intensity only reconstruction method E2Vid (Rebecq *et al.*, 2019*a*).

E2Vid(Rebecq *et al.*, 2019*a*) eliminates most of the artifacts dominant in the event based intensity reconstruction such as bleeding edges. However, in some cases the reconstructed intensity images start to show a dark region as shown in Fig. 5.5. We also observe that the dark region on the reconstructed images grow and occupy larger area as more frames are reconstructed. The images from (Rebecq *et al.*, 2019*a*) were reconstructed by accumulating N_e events into an event voxel-grid consisting of 5 frames. We consider $N_e = 0.35 \times H \times W$, where H and W represent the sensor resolution and this has shown to produce impressive results in most cases. The scenes shown in Fig. 5.5 are from the *indoor_flying* and the *outdoor_driving* sequences from MVSEC dataset (Zhu *et al.*, 2018*b*). These sequences are acquired with a hybrid sensor with a sensor resolution of $H = 260$, $W = 346$. As our algorithm requires only a single event frame as input, we accumulate $N_e/5$ events into each event frame to predict the intensity frames. In Fig. 5.5, we provide the qualitative comparison for the predicted images from the two sequences *indoor_flying* and the *outdoor_driving*.

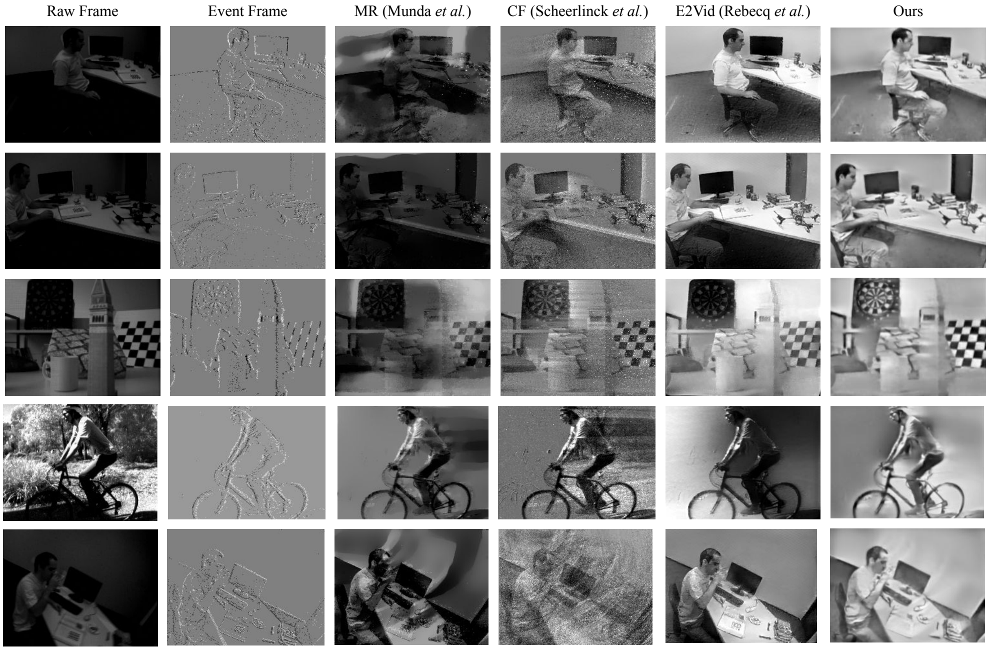


Fig. 5.4: Qualitative comparison of intensity frame reconstruction on various event sensor sequences. We see that the reconstructed intensity images from our method do not have trailing edges and noisy regions as compared to (Reinbacher *et al.*, 2016*a*; Scheerlinck *et al.*, 2018*b*). Ours and (Rebecq *et al.*, 2019*a*), both use learning based intensity reconstruction method and produce comparable results.

5.4.2 Optical Flow

In this section we evaluate the predicted optical flow, both qualtitatively and quantitatively, using the *indoor_flying* sequences from MVSEC dataset. Following (Zhu *et al.*, 2018*a*), we choose the metrics (a) Average End-point Error (AEE) which measures the mean absolute error and (b) percentage outliers for quantitative comparison. Percentage outlier (% outlier) measures the percentage of pixels with end-point error above 3 pixels and 5% of the magnitude of the flow vector. For fair comparison, we select two state of

Method	**indoor flying 1**		**indoor flying 2**		**indoor flying 3**	
	AEE	% outliers	AEE	% outliers	AEE	% outliers
Zhu *et al.* (2018*a*)	0.83	0.84	1.19	6.75	1.07	4.97
Zhu *et al.* (2019)	0.58	0	1.02	4	0.87	3
Ours	**0.49**	**0.02**	**0.55**	**0.05**	**0.53**	**0.03**

Table 5.2: Quantitative comparison of the predicted optical flow on event sequences from (Zhu *et al.*, 2018*b*).

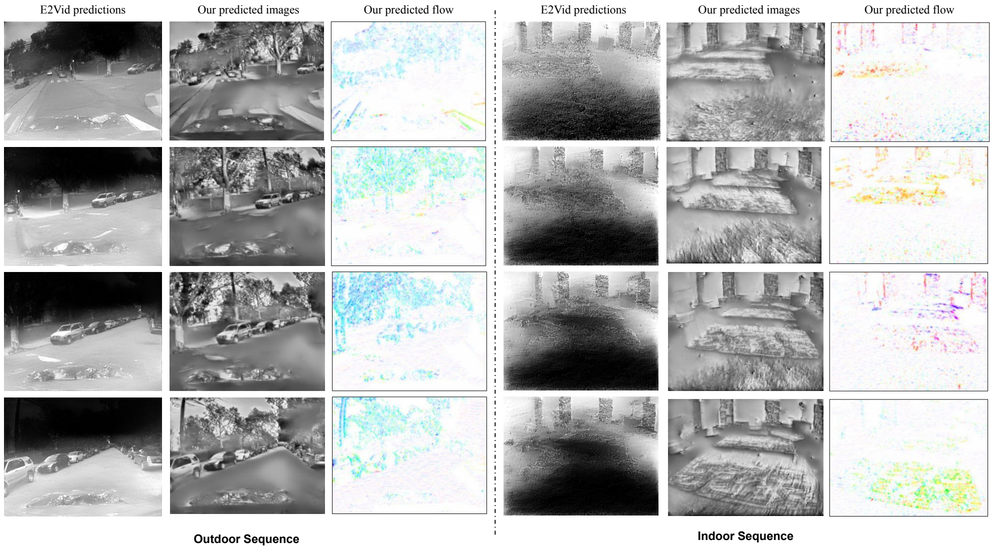


Fig. 5.5: We show two sequences, one outdoor and another indoor, never seen by our method or the one proposed in (Rebecq *et al.*, 2019*a*). We see that the images predicted by (Rebecq *et al.*, 2019*a*) degrade with a growing dark region in the predicted intensity images in this particular case. Our proposed method generalizes enough to provide a reliable estimate of the intensity image and the optical flow.

the art *unsupervised* learning-based optical flow algorithms (Zhu *et al.*, 2019, 2018*a*) to benchmark our proposed algorithm. In (Zhu *et al.*, 2018*a*), all the events between two successive intensity frames are accumulated into a frame-based representation and fed to the trained network. In (Zhu *et al.*, 2019), a volume consisting of 30000 events divided over 10 event frames is fed into the optical flow network. Effectively, each of event frames in (Zhu *et al.*, 2019) is formed by accumulating 3000 events from the event data. For a fair comparison, we too accumulate successive 3000 events into a single event frame which is then sequentially fed to our trained model.

In Table 5.2 we provide the quantitative metrics to compare our optical flow algorithms with the state of the art methods. We qualitatively compare the optical flow predicted from our model with (Zhu *et al.*, 2018*a*) in Fig. 5.6. We show optical flow predicted from various test sequences from datasets proposed by (Scheerlinck *et al.*, 2018*b*; Mueggler *et al.*, 2017) in Fig. 5.7. Note that these test sequences do not have ground truth optical flow to be compared against.

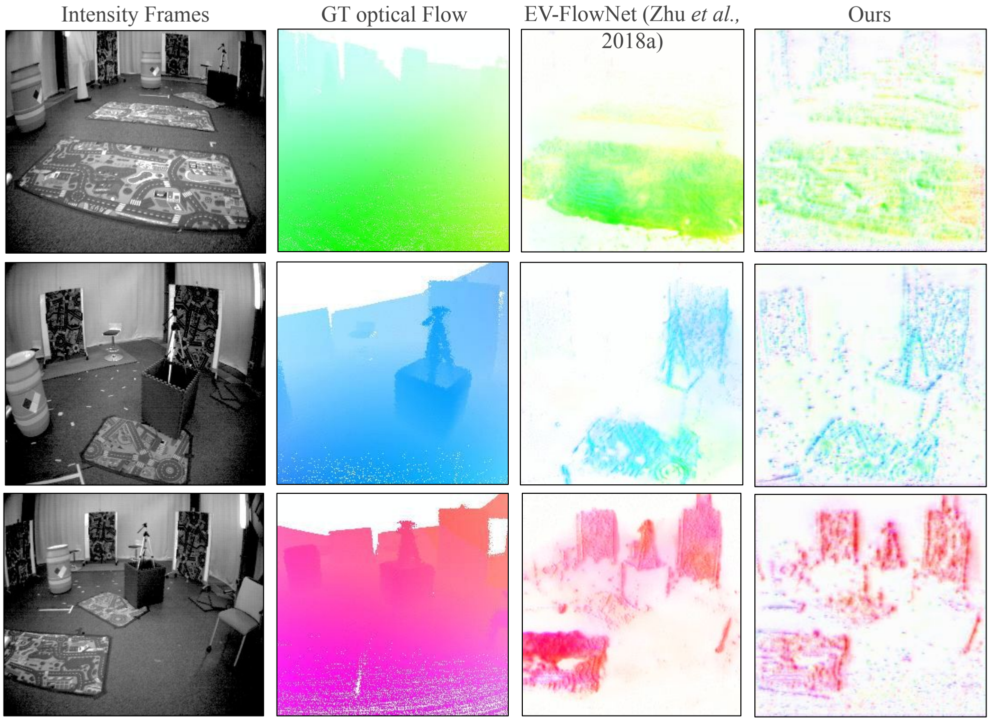


Fig. 5.6: We show some qualitative comparisons of the predicted optical flow on the *indoor_flying* sequence (Zhu *et al.*, 2018*b*).

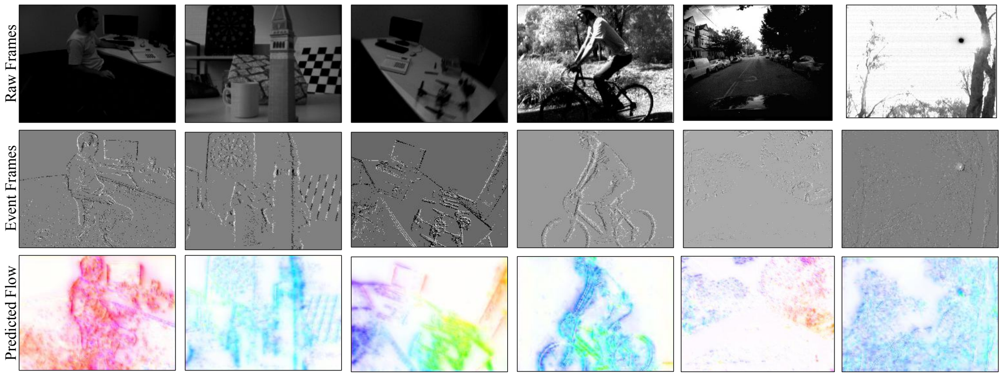


Fig. 5.7: We test our proposed optical flow model for its generalizability on various test sequences obtained from (Mueggler *et al.*, 2017; Scheerlinck *et al.*, 2018*b*; Zhu *et al.*, 2018*b*).

5.4.3 Advantages of event-based optical flow prediction

In this section, we demonstrate the advantages event sensors can provide over conventional image sensors for challenging scenes with fast motion and high dynamic range. In Fig. 5.1, we show an indoor scene with significant motion blur in the acquired image frames. A significant temporal information has also been lost between the two intensity frames. However, due to the high temporal resolution of the event sensors we are able to

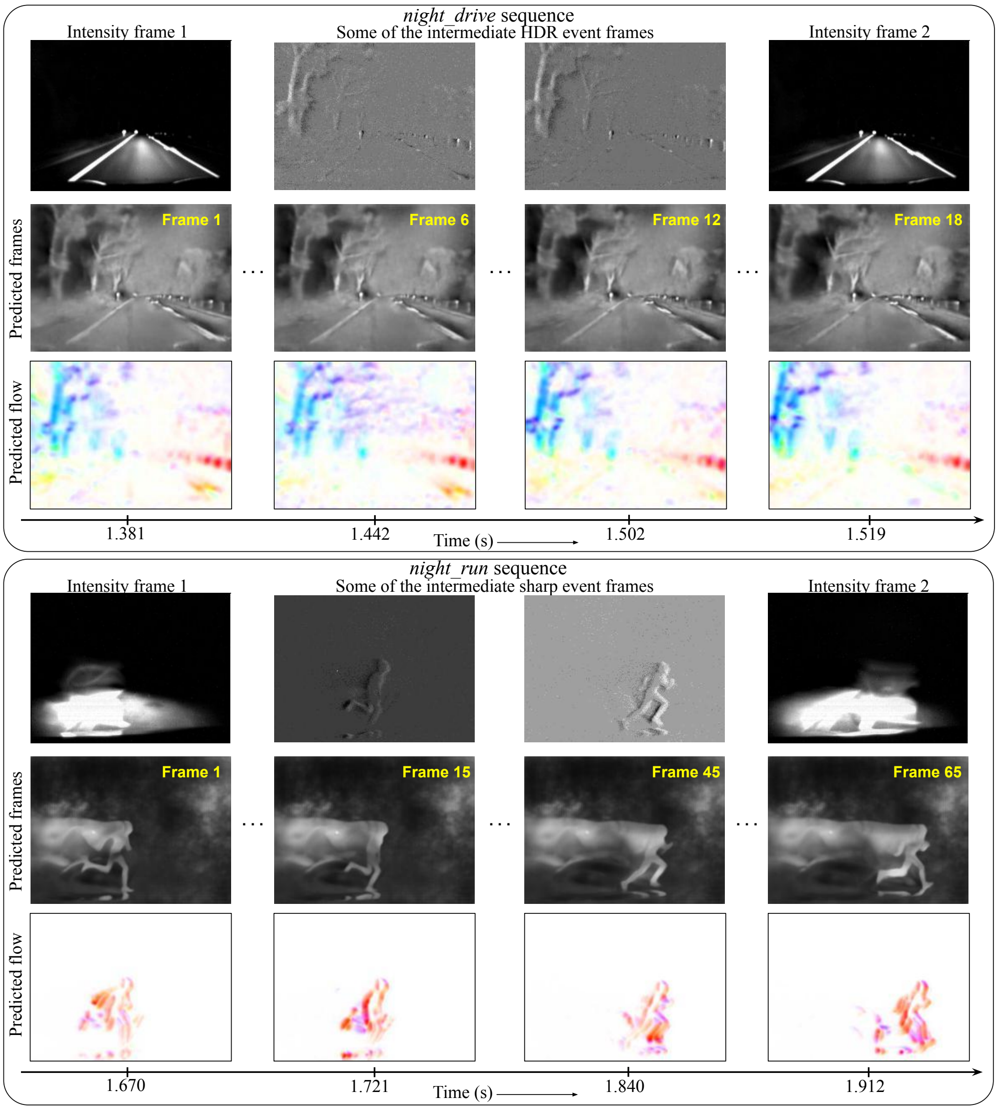


Fig. 5.8: The top figure shows the *night_drive* sequence shot in low-light conditions, demonstrating the ability of event sensors to sense objects at a high dynamic range, allowing the prediction of optical flow in extreme challenging cases. The *night_run* sequence combines two challenging scenarios, low-light and motion blur. With the help of event sensors we are able to predict the optical flow and intensity images at an effective rate of 1300 fps.

reconstruct multiple intensity frames, 60 in this case, between the successive intensity frames. Some of the 60 optical flow predictions have been shown in Fig. 5.1. Effectively, for this case, the intensity image and optical flow are being predicted at 1200 fps. This is a very high temporal resolution compared to many commercially available image sensors. We also show the optical flow predicted by two frame-based techniques (Hui *et al.*, 2018; Liu *et al.*, 2009) in Fig. 5.1.

In Fig. 5.8, we consider two more cases. A *night_drive* sequence which is captured

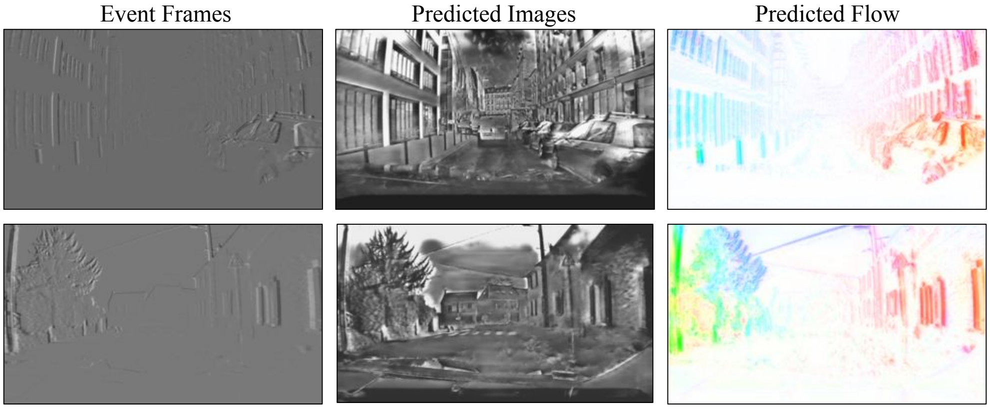


Fig. 5.9: Reconstruction results obtained from dataset proposed in (Perot *et al.*, 2020). The dataset is collected using a 1MP resolution ATIS sensor which acquires only the event sensor data and no intensity frames. We observe that our proposed algorithm is able to generalize well to this new dataset.

in extreme low-light conditions and a *night_run* sequence which combines both the extreme low-light and the fast scene-motion cases. These two sequences are obtained from the dataset proposed in (Scheerlinck *et al.*, 2018*b*). In the *night_drive* sequence, the acquired intensity frames are under-saturated with most of the frame being dark. However, the intensity frames reconstructed from the event sensor reveals most of the details such as trees on the roadside. The *night_run* sequence reveals the high dynamic range and high temporal resolution nature of the event sensor. In this sequence, a person runs across the road in an extremely low-light scenario lit by only car headlamps. The acquired intensity frames are severely blurred along with parts of the image being saturated. Again, the intensity frames reconstructed from the event sensor data reveal the full details of the scene being captured. In this particular case, the intensity frames and optical flow are being reconstructed at an effective frame rate of 1300 fps. These examples clearly demonstrate the advantages of obtaining the optical flow directly from the event sensor data.

5.5 Generalization of the algorithm

5.5.1 Generalization to novel sensors

The proposed algorithm is built assuming a specific category of event sensor where a positive or negative event is triggered when there is a change in the intensity. As long as this assumption is satisfied, we believe that the proposed algorithm should be able to predict the intensity image as well as the optical flow. To verify this, we considered a new dataset proposed by Perot *et al.* (2020), collected using a 1 megapixel ATIS (Posch *et al.*, 2014). This dataset is sufficiently different from the one that we have used for training. The resolution of ATIS is far larger than the DAVIS sensor and the sensor technology is developed independently of the Dynamic Vision Sensor (DVS)/DAVIS sensor family. We provide the predicted optical flow and intensity images in Fig. 5.9 without training the proposed algorithm on this novel dataset. We observe that our proposed algorithm is able to generalize well showing that the algorithm works well with different types of sensors.

5.5.2 Generalization to new event rates

We chose the SBN strategy for event frame generation due to its property of being able to adapt to slow and fast motions. However, our proposed network was trained by generating frames with the SBT strategy. In this strategy events from a fixed time interval are grouped into frames. We note that, the number of events in each fixed time interval can vary depending on the texture and relative camera motion. We provide the distribution of the number of events in a fixed time interval averaged across all sequences from the training set in Fig. 5.10. We observe that by using the SBT approach, we are training our algorithm for event frames containing different number of events per frame. However, a majority of the event frames contain number of events in the range $[2000, 4000]$. Hence, by using 3000 events per frame there's no major domain shift while testing. In Table 5.3, we show the quantitative results on optical flow accuracy for $1000, 3000, 5000$ and 7000 events per event frame.

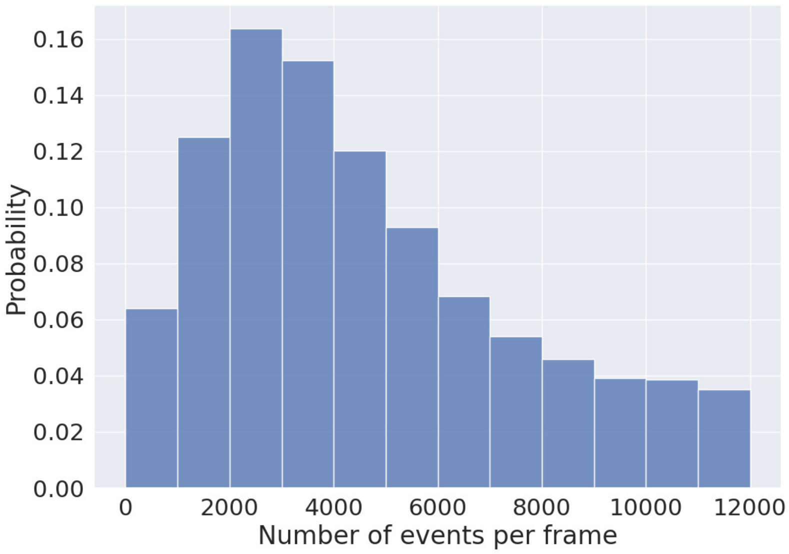


Fig. 5.10: Histogram of number of events per frame in the SBT strategy used to form event frames for training.

Events per frame	indoor flying 1		indoor flying 2		indoor flying 3	
	AEE	% outliers	AEE	% outliers	AEE	% outliers
1000	0.83	2.04	0.97	2.89	1.05	3.04
3000	0.49	0.02	0.55	0.05	0.53	0.03
5000	0.613	0.2	0.736	0.21	0.711	2.4
7000	0.842	1.05	1.04	2.4	1.02	2.27

Table 5.3: Quantitative optical flow comparison for different number of events per frame. Optical flow accuracy is highest for event frames with 3000 events per frame and degrading gracefully for other values of the number of events.

5.6 Ablation studies on the architecture

5.6.1 Choice of distance metric for intensity image supervision

In Eq. (5.2), we introduced a gradient-based L1 distance metric suitable for supervising intensity frame prediction from event sensors. Here, we evaluate the effectiveness of our proposed metric against other common metrics used for supervising image regression problems. We particularly consider two different cost functions, one based on pixel-wise error and the other based on perceptual similarity metric. For pixel-wise error we consider the MAE defined as,

$$d(\hat{I}, I) = \frac{1}{M} \sum \|(\hat{I} - I) \odot m\|_1 \tag{5.8}$$

where I and $\hat{I}$ are respectively the ground truth and the predicted intensity images. The mask m defined in Eq. 5.3 is again used to mask the pixels which are saturated in the low dynamic range intensity images. Rebecq *et al.* (2019*a*) use a learned perceptual similarity metric, LPIPS (Zhang *et al.*, 2018*a*), for supervising intensity image prediction from event data. We also use the same perceptual metric, LPIPS (Zhang *et al.*, 2018*a*), as the distance metric between our predicted and the ground truth intensity images. For a fair comparison, we retrain our proposed network on these two metrics with the same hyperparameters as used for the main experiment. In Fig. 5.11, we qualitatively compare the intensity images obtained by using the MAE and the LPIPS metrics. The MAE distance metric wrongly penalizes the neural network to predict the absolute intensity values at each pixel that cannot be recovered from the event sensor data alone. As we use real data to train our proposed network, the mismatch in the dynamic range of the input event data and the ground truth intensity images make the LPIPS metric unsuitable. When using pixel-wise loss, the image regions which do not match the dynamic range can be masked. Such a flexibility is not provided by perceptual metrics such as LPIPS. Thus, we observe that the predicted images contain artifacts when using the MAE and the LPIPS metrics. From Table 5.4, we also see that the MAE and LPIPS metrics affect the accuracy of the predicted optical flow. Hence, our proposed gradient-based L1 metric performs better for the case of training with real data than other metrics for intensity image regression.

5.6.2 Single decoder network to predict intensity image and optical flow

Our network is trained in a multi-task learning fashion with a single encoder and two decoders for the two different tasks of intensity image and optical flow prediction. However, it is also possible to use only a single decoder to predict both the intensity image and optical flow. This leads to reduction in the number of parameters that need to be

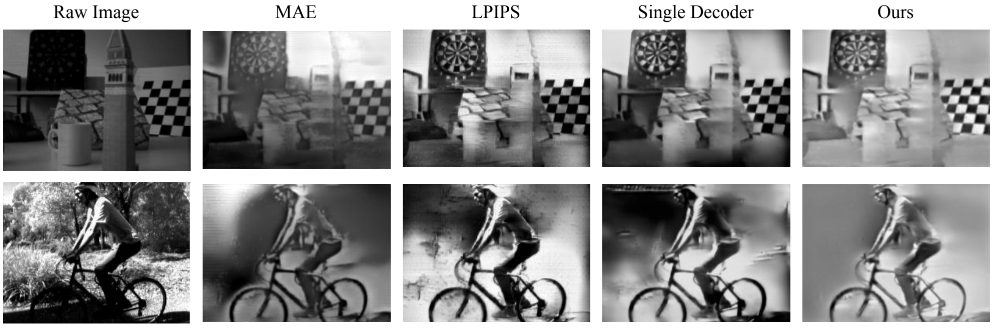


Fig. 5.11: We compare the effect of various architectural and supervision choices on intensity image estimation with respect to our proposed method. We show intensity image estimates for two different sequences obtained when using different two different cost functions, MAE and a perceptual metric LPIPS (Zhang *et al.*, 2018*a*). We also show intensity image estimates when using a single decoder to predict both the intensity frame and the optical flow.

	indoor flying 1		**indoor flying 2**		**indoor flying 3**	
	AEE	% outliers	AEE	% outliers	AEE	% outliers
MAE	0.57	0.04	0.63	0.75	0.61	0.07
LPIPS	0.53	0.1	0.58	0.5	0.58	0.1
Single Decoder	0.54	0.6	0.61	0.1	0.59	0.23
Ours	**0.49**	**0.02**	**0.55**	**0.05**	**0.53**	**0.03**

Table 5.4: We quantitatively compare the accuracy in optical flow estimation when the intensity image is supervised with MAE and LPIPS (Zhang *et al.*, 2018*a*). We also compare the optical flow accuracy for the case when a single decoder is used to predict both the intensity images and the optical flow.

trained, hence reducing the amount of data required to train the network. We explored this option of training a single decoder network to predict both the intensity image and the optical flow. For this experiment, we use our proposed decoder network *decoderImg* as our base network to predict the intensity images. To this network we augment two additional convolutional layers for optical flow prediction with 2 channels as output. These convolutional layers take as input the feature maps from the final 2 layers of the *decoderImg* network. Again, for a fair comparison we use the same hyperparamters to train this network as the ones used for our main experiment as described in Sec. 5.4. We provide qualitative results of the intensity images predicted from the single decoder network in Fig. 5.11. We also compare the optical flow estimation accuracy quantitatively for the different ablation experiments in Table 5.4. It can be observed that using

Network	Number of parameters	Run time at resolution	
		180×240	256×256
Two decoders	2.4 M parameters	4.91 ms	5.89 ms
Single decoder	1.9 M parameters	3.9 ms	4.8 ms

Table 5.5: Runtime of different networks. Our proposed framework can process more than 150 fps at a resolution of 256×256.

a single decoder reduces the performance of the algorithm on both the intensity and optical flow prediction. However, use of two different decoder networks does not increase the runtime significantly as shown in Table 5.5. The inference time of the different networks is computed on a machine with Nvidia TitanX GPU with Intel Xeon processor. We can see that our proposed framework can process more than 150 fps at a resolution of 256×256.

5.7 Conclusion

In this work, we propose an algorithm to simultaneously predict the intensity and optical flow from event sensor data. The optical flow prediction is self-supervised and hence does not require difficult to acquire ground truth optical flow for event data. As our algorithm requires as few as 3000 events per time-step, the optical flow is predicted at a very high temporal resolution of more than 1000 fps for scenes with large motion. This high temporal resolution prediction also enables our algorithm to handle any non-linear relative motion of the scene. Due to the sparse nature of event sensor data, the predicted optical flow is sparse as well, and predicting a dense optical flow from event data alone can be an interesting future direction.

CHAPTER 6

Self-supervised Light-Field Video Reconstruction from Stereo Video

6.1 Introduction

In Chapters 3, 4, and 5, we discussed reconstruction of high-speed videos from their corresponding low data-bandwidth measurements. LF video acquisition is another such high data-bandwidth signal that is challenging to acquire. LF imaging has emerged as a promising imaging technique to overcome the limitations of conventional photography such as post-capture focus control, novel view synthesis, and post-capture depth-of-field control. With video acquisition surging in popularity, LF video capture could enable simple post-capture focus control for videos acquired on consumer devices. However, acquiring LF video data at useful frame-rates remains challenging. For example, commercial LF cameras such as Lytro acquire LF videos at only 3 fps (Wang *et al.*, 2017). This is mainly because of the trade-off between angular, spatial, and temporal resolution of the acquired LF video. Modern cameras easily capture monocular videos at 720p resolutions at frame-rates of 30 fps. Ignoring the challenges of complex LF sensor, capturing a LF video at reasonable angular resolution of 7×7 requires a staggering $\sim 50\times$ more bandwidth. This is equivalent to capturing a 50MP video at 30 fps, something that is currently unimaginable for consumer devices.

While computational photography is poised to solve some of these problems in the upcoming decade via jointly optimized hardware-software solutions (Inagaki *et al.*, 2018; Veeraraghavan *et al.*, 2007; Wang *et al.*, 2015; Vadathya *et al.*, 2019), a practical solution is yet to be found. Numerous approaches have been proposed to overcome this challenge of high resolution LF imaging using hardware commonly available today. Table 6.1 provides a concise review of such existing methods. We particularly note the recent work attempting to reconstruct LF *images* from sparsely sampled angular views

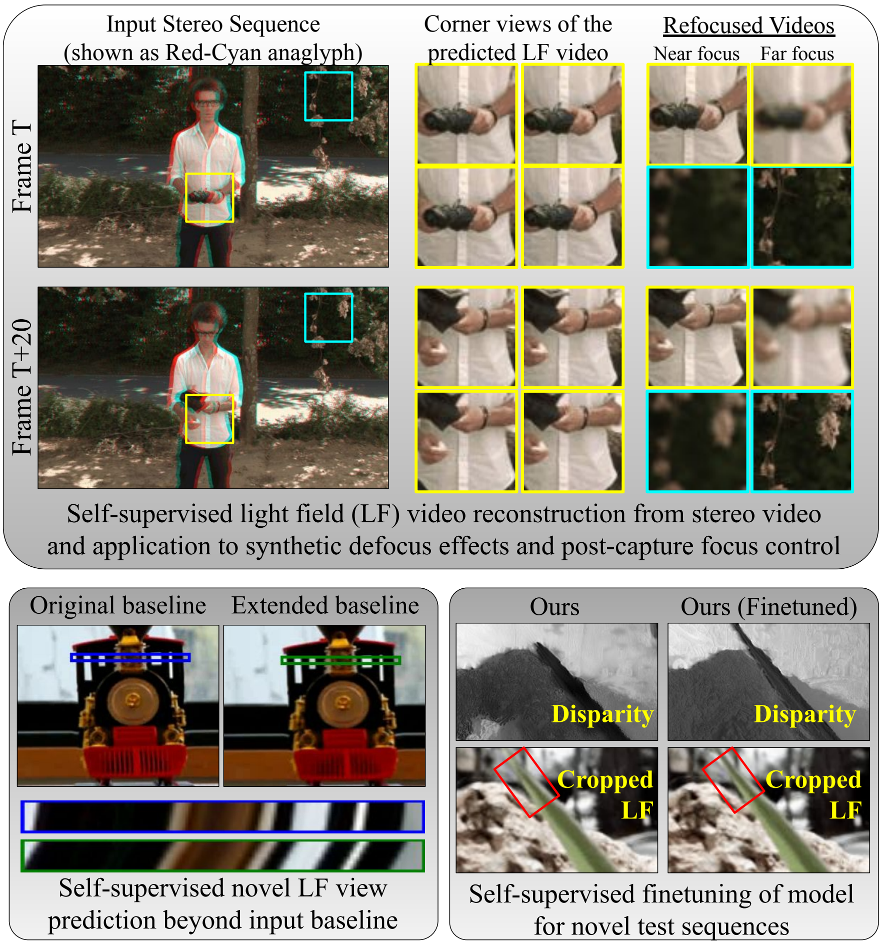

Input Stereo Sequence
(shown as Red-Cyan anaglyph)
Corner views of the
predicted LF video
Refocused Videos
Near focus
Far focus
Frame T
Frame T+20
Self-supervised light field (LF) video reconstruction from stereo video
and application to synthetic defocus effects and post-capture focus control
Original baseline
Extended baseline
Self-supervised novel LF view
prediction beyond input baseline
Ours
Ours (Finetuned)
Disparity
Disparity
Cropped
LF
Cropped
LF
Self-supervised finetuning of model
for novel test sequences

Fig. 6.1: We propose a self-supervised algorithm for LF video reconstruction from a stereo video, enabling applications such as post-capture focus control for videos. Our proposed algorithm allows for post-training fine-tuning on test sequences and variable angular view interpolation as well as extrapolation.

(Kalantari *et al.*, 2016; Bemana *et al.*, 2020). Considering the current limitations on available LF-hardware, we consider a simple case of sparse samples: the stereo image pair. In this paper, we tackle the task of reconstructing LF video from a sequence of stereo frames and propose a self-supervised learning-based algorithm as our solution.

The LF reconstruction in our self-supervised algorithm is guided via the geometric and temporal information embedded in a stereo video sequence. A recurrent neural network first takes the stereo frames at the current time-step as input and outputs a low-rank representation for LF frames based on layered LF displays (Wetzstein *et al.*, 2012). The full 4D LF frame is then obtained from this representation via a deterministic linear operation. To enforce the LF epipolar consistency, we impose a disparity-based geometric consistency constraint on the generated LF frames. To ensure temporal consistency of the generated LF frames, we enforce an optical flow-based constraint (Lai *et al.*, 2018). Two different recurrent neural networks are learned to estimate the disparity maps and optical flow from the input stereo video. All three networks, are trained via self-supervised cost functions during training.

One significant advantage of our approach is that it is self-supervised, and hence does not require hard-to-acquire ground-truth data for neural network training. Our algorithm is able to estimate the full 4D LF with any number of angular views from the input stereo views. We also show that our algorithm allows us to extend the baseline of the input views and generate novel views outside the original stereo baseline. Finally, our algorithm can be fine-tuned (see Sec. 6.5.4 and Fig. 6.1 and 6.11) on specific video sequences as it does not require ground truth data for supervision. Such self-supervised fine-tuning is especially useful when the test sequences do not follow the same distribution as the training sequences. We show that our proposed algorithm outperforms the state-of-the-art disparity-based LF reconstruction algorithms. Our algorithm also performs on par with unsupervised LF reconstruction approaches, e.g. X-fields (Bemana *et al.*, 2020) that requires 4 corner-views of the LF as its input. Overall, our contributions are:

- A self-supervised learning-based algorithm for LF video reconstruction from stereo video.
- Effective use of layered LF display based regularization for self-supervised LF video prediction.
- Facilitate post-training fine-tuning on test sequences and variable angular view prediction for both view interpolation and extrapolation.
- We show LF video reconstruction results on publicly available stereo videos captured in the wild.

6.2 Related Work

LF super-resolution The past decade saw the rise of commercial LF cameras but they quickly faded out of popularity due to the inherent angular and spatial resolution trade-off. Exploiting the correlations in the angular and spatial dimensions, several algorithms have been proposed to overcome this trade-off in LF imaging. Some of these

	Self-	Stereo-	
Method	Supervision	View	Video
LF synthesis (Kalantari *et al.*, 2016; Wu *et al.*, 2017; Wang *et al.*, 2018*a*; Farrugia and Guillemot, 2019; Guo *et al.*, 2018)	✗	✗	✗
View synthesis (Kalantari *et al.*, 2016; Mildenhall *et al.*, 2019; Flynn *et al.*, 2019)	✗	✗	✗
View Synthesis (Mildenhall *et al.*, 2020; Liu *et al.*, 2021; Zhang *et al.*, 2020)	✔	✗	✗
LF Video (Hajisharif *et al.*, 2020; Wang *et al.*, 2017)	✗	✗	✔
Bino-LF (Zhang *et al.*, 2015)	✔	✔	✗
X-fields (Bemana *et al.*, 2020)	✔	✗	✔
Ours	✔	✔	✔

Table 6.1: A concise, categorized overview of the related work.

approaches involve modified hardware setups such as coded masks on the aperture (Inagaki *et al.*, 2018; Veeraraghavan *et al.*, 2007; Wang *et al.*, 2015; Vadathya *et al.*, 2019) and near the sensor (Gupta *et al.*, 2017; Marwah *et al.*, 2013; Hajisharif *et al.*, 2020; Vadathya *et al.*, 2019, 2017). However, the complex optical hardware setups hinder small form factors necessary for consumer devices. Hence, other approaches that use conventional cameras have been proposed such as focal-stack (Vadathya *et al.*, 2019) and high-resolution LF reconstruction from sparse measurements (Kalantari *et al.*, 2016; Wu *et al.*, 2017; Wang *et al.*, 2018*a*; Farrugia and Guillemot, 2019; Guo *et al.*, 2018; Bemana *et al.*, 2020). Alternative approaches for a 3D scene such as Multi-Plane Image (MPI) (Zhou *et al.*, 2018*a*; Mildenhall *et al.*, 2019; Flynn *et al.*, 2019) and Neural Radiance Fields (NeRF) (Mildenhall *et al.*, 2020; Liu *et al.*, 2021; Zhang *et al.*, 2020) have also shown how to generate high-quality LFs. With the evolution of machine learning-based methods to estimate disparity from image semantics in a single image, synthesizing LF images from single images has also been popular (Li and Kalantari, 2020; Srinivasan *et al.*, 2017*b*; Tucker and Snavely, 2020).

LF video reconstruction While the spatial and angular dimensions of LF have received much attention, commercial LF cameras also suffer from low temporal resolution. A hybrid hardware setup with a commercial LF camera and a DSLR to en-

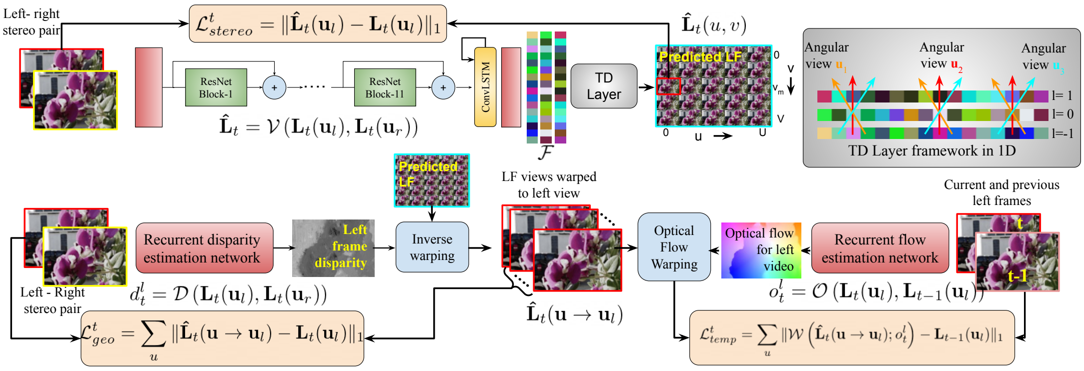

Left- right stereo pair
$\mathcal{L}^t_{stereo} = \|\hat{\mathbf{L}}_t(\mathbf{u}_l) - \mathbf{L}_t(\mathbf{u}_l)\|_1$
ResNet Block-1
+
ResNet Block-11
+
ConvLSTM
$\hat{\mathbf{L}}_t = \mathcal{V}(\mathbf{L}_t(\mathbf{u}_l), \mathbf{L}_t(\mathbf{u}_r))$
$\mathcal{F}$
TD Layer
$\hat{\mathbf{L}}_t(u, v)$
Predicted LF
0
v
v_m
V
0
u
U
Angular view $\mathbf{u}_1$
Angular view $\mathbf{u}_2$
Angular view $\mathbf{u}_3$
l= 1
l= 0
l=-1
TD Layer framework in 1D
Predicted LF
LF views warped to left view
Current and previous left frames
Recurrent disparity estimation network
Left frame disparity
Inverse warping
Optical Flow Warping
Optical flow for left video
Recurrent flow estimation network
t
t-1
Left - Right stereo pair
$d^l_t = \mathcal{D}(\mathbf{L}_t(\mathbf{u}_l), \mathbf{L}_t(\mathbf{u}_r))$
$\hat{\mathbf{L}}_t(\mathbf{u} \to \mathbf{u}_l)$
$o^l_t = \mathcal{O}(\mathbf{L}_t(\mathbf{u}_l), \mathbf{L}_{t-1}(\mathbf{u}_l))$
$\mathcal{L}^t_{geo} = \sum_u \|\hat{\mathbf{L}}_t(\mathbf{u} \to \mathbf{u}_l) - \mathbf{L}_t(\mathbf{u}_l)\|_1$
$\mathcal{L}^t_{temp} = \sum_u \|\mathcal{W}\left(\hat{\mathbf{L}}_t(\mathbf{u} \to \mathbf{u}_l); o^l_t\right) - \mathbf{L}_{t-1}(\mathbf{u}_l)\|_1$

Fig. 6.2: Overall flow of the proposed self-supervised algorithm for LF video reconstruction from stereo video. The LF frames are generated from the input stereo pair via an intermediate low-rank tensor-display (TD) based representation. The self-supervised learning of LF reconstruction is guided via self-supervised cost functions involving stereo pair, disparity maps and optical flow maps.

able capturing of LF videos at 30 fps was proposed in (Wang *et al.*, 2017). A single sensor-based compressive imaging approach that requires a mask near the sensor was proposed in (Hajisharif *et al.*, 2020). While these require complex hardware setups, Bae *et al.* (2021) propose to utilize a single monocular camera for 5D LF video reconstruction. Algorithms such as (Wang *et al.*, 2017; Hajisharif *et al.*, 2020; Bae *et al.*, 2021) are learning-based approaches that require supervised training data. As collecting large-scale ground-truth LF videos for training is challenging, Bemana *et al.* (2020) propose X-Fields, a self-supervised approach eliminating the need for ground-truth training datasets. X-Fields interpolates novel views in both angular and temporal directions. However, the X-Field results in the paper (Bemana *et al.*, 2020) use 4-views and our experiments in this paper demonstrate that reconstruction quality deteriorates significantly for this method when only two stereo views are available (see Fig. 6.5, Table 6.3). We propose a self-supervised algorithm capable of LF reconstruction from only a pair of stereo frames. The distinguishing factor of our work is that the layered LF display (Wetzstein *et al.*, 2012) based regularization that enforces correlations between horizontal and vertical disparity of the reconstructed LF. This constraint enables high-quality LF reconstruction even when only 1D disparity information is available (*i.e.*, stereo views).

Layered LF displays and neural networks Previously, layered LF display representations have been used in conjunction with neural networks. Maruyama *et al.* (2019) built an end-to-end pipeline from a coded aperture scene acquisition to displaying the scene on a layered LF display. Similar work in (Takahashi *et al.*, 2018; Kobayashi *et al.*, 2017) aims at capturing a focal stack and then learning to display the scene onto the LF display. Although layered display representations have been used in conjunction with neural networks, to the best of our knowledge, we are the first ones to use it as a regularizer for self-supervised LF reconstruction.

6.3 Self-supervised LF Video Reconstruction

In this section, we introduce our self-supervised algorithm for LF video reconstruction from an input video of stereo frames. We assume that the input stereo video is captured using a pair of rectified, synchronized and identical stereo cameras. LF video reconstruction from the stereo video is guided via geometric information obtained from individual stereo pairs and the temporal information obtained from the video sequences. A deep recurrent neural network first takes as input an individual stereo pair at the current time-step. It outputs an intermediate low-rank LF representation based on layered LF displays (Wetzstein *et al.*, 2012) (or Tensor Displays (TD)). A differentiable TD layer then takes this representation as input and generates the corresponding LF frame at the current time-step. Three different self-supervised cost functions based on photometric, geometric, and temporal constraints guide the self-supervised learning for LF reconstruction. The geometric and temporal constraints are imposed by disparity and optical flow maps, respectively. These are obtained via two separate self-supervised recurrent neural networks similar to (Godard *et al.*, 2017*a*; Meister *et al.*, 2018). Self-supervision of the full 4D LF prediction is explained in Sec. 6.3.1. Obtaining the LF frame from the intermediate representation is a deterministic linear operation as elaborated in Sec. 6.3.3.

6.3.1 Stereo LF estimation

In our proposed algorithm, to obtain the LF video sequence, we estimate the full 4D LF frame for each input pair of stereo frames. Let the required full 4D LF video sequence be denoted by $\mathbf{L}_t(\mathbf{u})$, where $\mathbf{u} = (u, v)$ denotes the 2D angular coordinates of the LF sub-aperture image (SAI). Here, we assume the input left-right frames, $\mathbf{L}_t(\mathbf{u}_l)$ and $\mathbf{L}_t(\mathbf{u}_r)$ are sparse samples of $\mathbf{L}_t(\mathbf{u})$ at SAI co-ordinates $\mathbf{u}_l = (0, v_m)$ and $\mathbf{u}_r = (U, v_m)$, as shown in Fig. 6.2. To predict the LF, we use a deep residual neural network (Maruyama *et al.*, 2019), $\mathcal{V}$, coupled with a recurrent architecture as shown in Fig. 6.2. The network $\mathcal{V}$ takes as input the stereo frames $(\mathbf{L}_t(\mathbf{u}_l), \mathbf{L}_t(\mathbf{u}_r))$ and outputs a low-rank approximation, $\mathcal{F}$, of the desired LF $\hat{\mathbf{L}}_t$. A parameter-free TD layer (Maruyama *et al.*, 2019), added after $\mathcal{V}$, takes the representation $\mathcal{F}$ as input and outputs the estimated 4D LF frame $\hat{\mathbf{L}}_t$. We further elaborate on this TD layer in Sec. 6.3.3 and for now, we assume that $\mathcal{V}$ finally outputs the LF frame $\hat{\mathbf{L}}_t$ from the input frames $(\mathbf{L}_t(\mathbf{u}_l), \mathbf{L}_t(\mathbf{u}_r))$. As we do not have ground truth LF $\mathbf{L}_t$, we supervise the training of $\mathcal{V}$ by three different self-supervised cost functions based on photometric, geometric and temporal consistency constraints.

Photometric consistency We define the photometric consistency cost as

$$\mathcal{L}^t_{stereo} = \|\hat{\mathbf{L}}_t(\mathbf{u}_l) - \mathbf{L}_t(\mathbf{u}_l)\|_1 + \|\hat{\mathbf{L}}_t(\mathbf{u}_r) - \mathbf{L}_t(\mathbf{u}_r)\|_1 \,, \tag{6.1}$$

which ensures the consistency of $\hat{\mathbf{L}}_t$ with respect to the two known measurements, $\mathbf{L}_t(\mathbf{u}_l)$, $\mathbf{L}_t(\mathbf{u}_r)$, of $\mathbf{L}_t$.

Geometric consistency The geometric consistency cost enforces $\hat{\mathbf{L}}_t$ to follow the same underlying scene geometry as that of the captured stereo pair. To enforce such a constraint, we first estimate dense disparity maps from the individual input stereo frames via a recurrent neural network $\mathcal{D}$. The network architecture $\mathcal{D}$ is inspired from FlowNet (Dosovitskiy *et al.*, 2015) and is augmented with a ConvLSTM network after the encoder network. The disparity maps d_t^l and d_t^r are estimated as,

$$d_t^l = \mathcal{D}\left(\mathbf{L}_t\left(\mathbf{u}_l\right), \mathbf{L}_t\left(\mathbf{u}_r\right)\right) \quad d_t^r = \mathcal{D}(\mathbf{L}_t\left(\mathbf{u}_r\right), \mathbf{L}_t\left(\mathbf{u}_l\right)) \ . \tag{6.2}$$

As no ground-truth disparity maps are available for supervision, disparity prediction is self-supervised via a photo-consistency based loss (Godard *et al.*, 2017*a*,*b*; Zhou *et al.*, 2017; Yin and Shi, 2018; Godard *et al.*, 2019*b*),

$$\mathcal{L}_{disp}^t = \|\mathcal{W}(\mathbf{L}_t\left(\mathbf{u}_l\right); d_t^l) - \mathbf{L}_t\left(\mathbf{u}_r\right)\|_1 + \|\mathcal{W}(\mathbf{L}_t\left(\mathbf{u}_r\right); d_t^r) - \mathbf{L}_t\left(\mathbf{u}_l\right)\|_1 \ . \tag{6.3}$$

Here, $\mathcal{W}$ denotes the bilinear inverse warping operator (Jaderberg *et al.*, 2015) that takes as input a displacement map and remaps the images. In inverse warping, a displacement of $(\delta x, \delta y)$ is specified at a pixel $P = (x, y)$. Using this, we fill the intensity value at pixel P of the target frame T from the source frame S. The intensity to be filled at $T(x, y)$ is obtained from $S(x + \delta x, y + \delta y)$. As S is a discrete signal, the intensity at $S(x + \delta x, y + \delta y)$ has to be generally interpolated from the neighboring pixels ($\{p_1, p_2, p_3, p_4\}$): $p_1 = (\lfloor x + \delta x \rfloor, \lfloor y + \delta y \rfloor)$, $p_2 = (\lfloor x + \delta x \rfloor, \lceil y + \delta y \rceil)$, $p_3 = (\lceil x + \delta x \rceil, \lfloor y + \delta y \rfloor)$, $p_4 = (\lceil x + \delta x \rceil, \lceil y + \delta y \rceil)$. For bilinear interpolation we define the weights $a = x + \delta x - \lfloor x + \delta x \rfloor$ and $b = y + \delta y - \lfloor y + \delta y \rfloor$. Then $T(x, y) = abS(p_1) + a(1 - b)S(p_2) + (1 - a)bS(p_3) + (1 - a)(1 - b)S(p_4)$.

To impose the geometric consistency on $\hat{\mathbf{L}}_t$, we take a SAI $\hat{\mathbf{L}}_t(\mathbf{u})$ at $\mathbf{u}$ and approximate the LF views at $\mathbf{u}_l$ and $\mathbf{u}_r$ via disparity based warping as seen in Fig. 6.2. But, we already know the ground-truth intensity frame at SAI co-ordinates $\mathbf{u}_l$ and $\mathbf{u}_r$ which are the input stereo frames $\mathbf{L}_t\left(\mathbf{u}_l\right), \mathbf{L}_t\left(\mathbf{u}_r\right)$ respectively. The error between the approximated and the known input stereo views acts as the supervisory signal for LF estimation. In essence, we warp $\hat{\mathbf{L}}_t(\mathbf{u})$ to the SAIs at $\mathbf{u}_l$ and $\mathbf{u}_r$ to obtain $\hat{\mathbf{L}}_t(\mathbf{u} \to \mathbf{u}_l)$ and $\hat{\mathbf{L}}_t(\mathbf{u} \to \mathbf{u}_r)$ respectively. This can be expressed as,

$$\hat{\mathbf{L}}_t(\mathbf{u} \to \mathbf{u}_r) = \mathcal{W}\left(\hat{\mathbf{L}}_t(\mathbf{u}); ((\mathbf{u} - \mathbf{u}_r)d_t^r)\right) \tag{6.4}$$

$$\hat{\mathbf{L}}_t(\mathbf{u} \to \mathbf{u}_l) = \mathcal{W}\left(\hat{\mathbf{L}}_t(\mathbf{u}); ((\mathbf{u} - \mathbf{u}_l)d_t^l)\right) \tag{6.5}$$

In Eqs. (6.4) and (6.5), $\mathbf{u}-\mathbf{u}_l$ and $\mathbf{u}-\mathbf{u}_r$ are each vectors of length 2. When multiplied with $d_t^l \in \mathbb{R}^{h\times w}$ and $d_t^r \in \mathbb{R}^{h\times w}$ respectively, we obtain the displacements in both x and y directions to warp the angular views to the input stereo views. The implementation of $(\mathbf{u}-\mathbf{u}_l)d_t^l$ is performed as a matrix multiplication between $(\mathbf{u}-\mathbf{u}_l) \in \mathbb{Z}^{2\times 1}$ and $d_t^l \in \mathbb{R}^{1\times h\times w}$. Similarly, $(\mathbf{u}-\mathbf{u}_r)d_t^r$ is implemented with $(\mathbf{u}-\mathbf{u}_r) \in \mathbb{Z}^{2\times 1}$ and $d_t^r \in \mathbb{R}^{1\times h\times w}$. The geometric consistency error between the approximated stereo pairs (from the estimated LF) and the known input stereo pairs is then defined as,

$$\mathcal{L}_{geo}^t = \sum_{\mathbf{u}} \sum_{k\in\{l,r\}} \|\hat{\mathbf{L}}_t(\mathbf{u}\to\mathbf{u}_k) - \mathbf{L}_t(\mathbf{u}_k)\|_1. \tag{6.6}$$

Temporal consistency The sequence of estimated LF frames $\hat{\mathbf{L}}_t$ form a video sequence when they are temporally consistent. Here, we use the optical flow estimated from the input sequence of stereo frames to enforce temporal consistency between successive predicted LF frames. With solely the stereo frames as input, it is only possible to estimate optical flow at SAIs $\mathbf{u}_l$ and $\mathbf{u}_r$. We employ a recurrent neural network $\mathcal{O}$ to estimate the optical flows $o_t^l, o_t^r \in \mathbb{R}^{h\times w\times 2}$ for the left and right temporal sequences, respectively. The input left-right pairs are input to $\mathcal{O}$ and the optical flow is obtained as

$$o_t^l = \mathcal{O}\left(\mathbf{L}_t\left(\mathbf{u}_l\right), \mathbf{L}_{t-1}\left(\mathbf{u}_l\right)\right)\quad, o_t^r = \mathcal{O}\left(\mathbf{L}_t\left(\mathbf{u}_r\right), \mathbf{L}_{t-1}\left(\mathbf{u}_r\right)\right). \tag{6.7}$$

Since the ground truth optical flow is unavailable, we choose to learn the optical flow with a self-supervised learning algorithm (Meister *et al.*, 2018; Ren *et al.*, 2017; Wang *et al.*, 2018*b*, 2019*b*; Jason *et al.*, 2016). We define the photoconsistency based self-supervised cost function (Meister *et al.*, 2018; Ren *et al.*, 2017; Wang *et al.*, 2018*b*, 2019*b*; Jason *et al.*, 2016) for training optical flow network $\mathcal{O}$ as,

$$\mathcal{L}_{flow}^t = \sum\nolimits_{k\in\{l,r\}} \|\mathcal{W}\left(\mathbf{L}_t\left(u_k\right); o_t^k\right) - \mathbf{L}_{t-1}\left(u_k\right)\|_1 \tag{6.8}$$

where we use $k \in \{l, r\}$ to sum over both left and right images. To enforce temporal consistency, we utilize the images $\hat{\mathbf{L}}_t(\mathbf{u}\to\mathbf{u}_l)$ and $\hat{\mathbf{L}}_t(\mathbf{u}\to\mathbf{u}_r)$ which represent the LF SAIs warped to the stereo SAI co-ordinates $\mathbf{u}_l$ and $\mathbf{u}_r$. With the estimated optical flows o_t^l and o_t^r, $\hat{\mathbf{L}}_t(\mathbf{u}\to\mathbf{u}_l)$ and $\hat{\mathbf{L}}_t(\mathbf{u}\to\mathbf{u}_r)$ are warped to approximate the images at

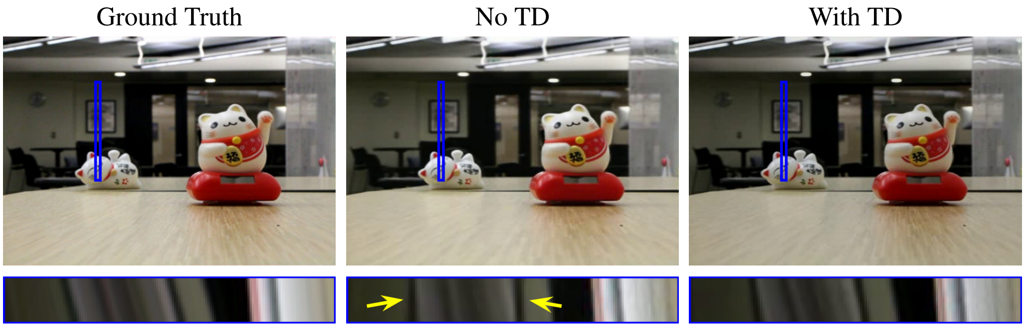


Fig. 6.3: The figure shows EPI for vertical views for a small region of the image. It can be seen that the intermediate representation $\mathcal{F}$ assists in better recovery of the LF frame than direct regression.

the SAIs $\mathbf{u}_l$ and $\mathbf{u}_r$ at the timeframe $t-1$. The SAIs $\mathbf{u}_l$ and $\mathbf{u}_r$ at the timeframe $t-1$ are given by $\mathbf{L}_{t-1}\left(\mathbf{u}_l\right)$ and $\mathbf{L}_{t-1}\left(\mathbf{u}_r\right)$ respectively. The corresponding temporal error is defined as,

$$\mathcal{L}_{temp}^{t} = \sum_{\mathbf{u}} \sum_{k\in\{l,r\}} \|\mathcal{W}\left(\hat{\mathbf{L}}_t\left(\mathbf{u}\rightarrow\mathbf{u}_k; o_t^k\right)\right) - \mathbf{L}_{t-1}\left(u_k\right)\|_1 \tag{6.9}$$

where minimizing the error enforces temporal consistency between successive frames.

6.3.2 Overall loss

We finally add total-variation (TV)-based smoothness constraint on the predicted disparity maps, optical flow and the LF frames. We define the TV smoothness loss as,

$$TV(I) = \|\nabla_x I\|_1 + \|\nabla_y I\|_1 \ , \tag{6.10}$$

where ∇_x and ∇_y are the x and y-gradient operators respectively. We define the overall smoothness loss as,

$$\mathcal{L}_{TV}^{t} = TV\left(\hat{\mathbf{L}}_t\right) + \sum\nolimits_{k\in\{l,r\}} TV\left(d_t^k\right) + TV\left(o_t^k\right). \tag{6.11}$$

Including all the cost functions, the overall cost function used to optimize the neural networks is defined as,

$$\mathcal{L} \;=\; \sum_{t=1}^{T} \lambda_1 \mathcal{L}^t_{disp} + \lambda_2 \mathcal{L}^t_{flow} + \lambda_3 \mathcal{L}^t_{stereo} + \lambda_4 \mathcal{L}^t_{geo} + \lambda_5 \mathcal{L}^t_{temp} + \lambda_6 \mathcal{L}^t_{TV} \ , \tag{6.12}$$

where T is the total number of frames in the video sequence.

6.3.3 Low-rank regularization

As elaborated in Sec. 6.3.1, the LF reconstruction network $\mathcal{V}$ learns to estimate a low-rank representation $\mathcal{F}$ of the desired LF frame. Let's consider the direct estimation of the full 4D LF frame $\hat{\mathbf{L}}_t$ of angular resolution $U \times V$. In this case, $\mathcal{V}$ outputs $U \times V \times 3$ independent channels representing $U \times V$ RGB frames. Such a network design ignores the grid-like structure inherent to a 4D LF frame. Effective utilization of such a structure can lead to a better overall performance of the algorithm. We choose to impose the grid-like structure of the 4D LF frames via the tensor-display (Wetzstein *et al.*, 2012) based low-rank representation. In Fig. 6.3 we show that imposing such a low-rank regularizer indeed helps in better recovery of the LF frame. The network $\mathcal{V}$ outputs an intermediate low-rank representation $\mathcal{F} = [\mathbf{f}_{-L/2}, \ldots, \mathbf{f}_0, \ldots, \mathbf{f}_{L/2}]$, where $\mathbf{f}_k = [f^1_k, f^2_k, \ldots, f^M_k]^T$, $f^m_k \in [0,1]^{h \times w \times 3}$ consists of LM RGB channels, where L and M represent the number of layers and the rank, respectively. A linear, parameter-free layer $TD(\cdot)$ takes as input the representation $\mathcal{F}$ and outputs the corresponding LF frame. An intuitive picture of the TD layer is shown in Fig. 6.2. The operation of $TD(\cdot)$ can be mathematically described as (Wetzstein *et al.*, 2012),

$$L(x,y,u,v) = TD(\mathcal{F}) = \sum_{m=1}^{M} \prod_{l=-L/2}^{L/2} f^l_m(x+lu, y+lv) \tag{6.13}$$

where $L(x,y,u,v)$ represents the 4D LF rays, where (x,y) and (u,v) represent the spatial and angular dimensions respectively.

In Eq. (6.13) we observe that a matrix f^l_m is shifted by various values of u and v and multiplied with f^{l+1}_m. So, in essence, we are scaling the matrix f^{l+1}_m with the values in f^l_m creating a tensor of matrices in the process. All the matrices in this tensor are linearly dependent and can be expressed as scaled versions of f^{l+1}_m, and hence the tensor is rank-1. The outer-sum then adds multiple rank-1 tensors together thereby approximating the full LF frame as a low-rank tensor. This is analogous to the principal component analysis, where we represent an image as a sum of multiple rank-1 matrices.

6.4 Architecture and implementation details

6.4.1 Network Architecture

Here, we provide the details of the 3 different network architectures $\mathcal{V}$, $\mathcal{D}$ and $\mathcal{O}$.

Light field prediction network, $\mathcal{V}$ The LF prediction network consists of an input convolutional layers followed by 11 ResNet blocks (He *et al.*, 2016). The first convolutional layer takes as input the two RGB stereo pairs (6 channels) and outputs a 64 channel feature map without any spatial downsampling. This feature map is then input to the ResNet block where the number of channels at the output is kept the same as that of the input (here, 64 channels). Each ResNet block consists of 2 convolutional layers followed by the Rectified Linear Unit (ReLU) (Nair and Hinton, 2010) activation. The first convolutional layer of the ResNet block takes the 64 channel feature as input and outputs a 32 channeled feature map. The second convolutional layer in the ResNet block takes this intermediate 32 channel feature as input and outputs again a 64 channel feature map. There is no spatial downsampling or upsampling within the ResNet blocks. The feature map at the output of the 11^{th} ResNet block is then input to a ConvLSTM network (Shi *et al.*, 2015). The cell state of this ConvLSTM network is then input to a final convolutional layer which outputs 36 RGB (108) channels corresponding to the $L = 3$ layers and $M = 12$ rank of the low-rank LF representation $\mathcal{F}$. ReLU non-linearity is used at the output of the final convolutional layer to ensure non-negative values in $\mathcal{F}$.

Disparity and optical flow estimation network, $\mathcal{D}$ and $\mathcal{O}$ As shown in Fig. 6.4, the neural networks $\mathcal{D}$ and $\mathcal{O}$ are derived from FlowNet (Dosovitskiy *et al.*, 2015) network architecture. Although both networks share similar network architecture, the weights are completely independent and are not shared between the two networks. To facilitate

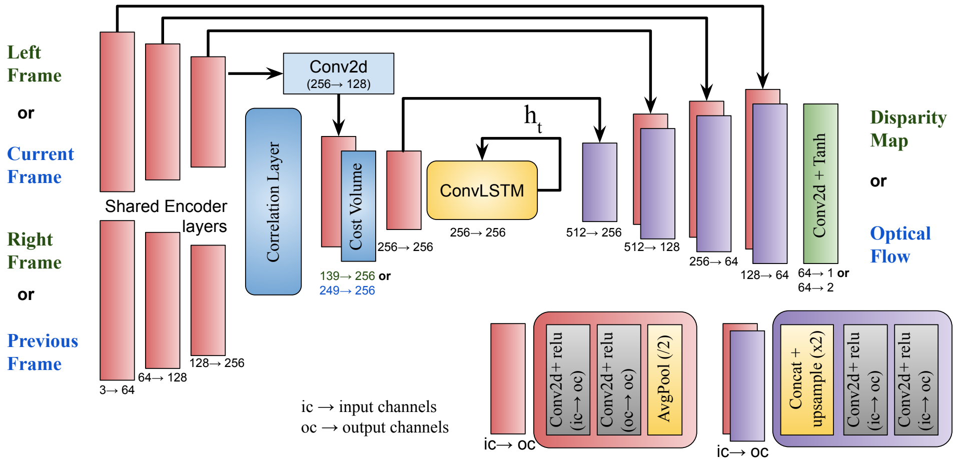


Fig. 6.4: For estimation of the disparity map and optical flow, we modify the FlowNet (Dosovitskiy *et al.*, 2015) architecture to include the ConvLSTM network (Shi *et al.*, 2015) at the encoder. All the layers in the neural network use 2D convolutional layers with kernel size of 3×3.

Parameter	kernel-size	patch-size	stride	padding	dilation	dilation-patch
Value ($\mathcal{O}$)	1×1	11×11	1	0	1	2
Value ($\mathcal{D}$)	1×1	1×11	1	0	1	2

Table 6.2: We show the values which we use for the different parameters of the correlation layer (Pinard, 2021) in the networks $\mathcal{O}$ and $\mathcal{D}$.

temporal consistency in the predicted outputs, a ConvLSTM network is used after the encoder block, following (Lai *et al.*, 2018). The major differences between the two networks are in the correlation layer (Dosovitskiy *et al.*, 2015) and the final output convolutional layer. The *correlation layer* which computes the cost volume between the two feature maps has 6 parameters (Pinard, 2021): kernel-size, patch-size, stride, padding, dilation, and dilation-patch. The details of these parameters for both the networks, $\mathcal{D}$, and $\mathcal{O}$, are provided in Table 6.2. Other network details such as the number of channels per layer are provided in Fig. 6.4. The final convolutional layer of both the networks uses the tanh non-linearity as shown in Fig. 6.4 While the disparity estimation network $\mathcal{D}$ outputs a single channel, the flow estimation network $\mathcal{O}$ outputs two channels.

The neural networks $\mathcal{O}$ and $\mathcal{D}$ are both recurrent architectures with a base network similar to that of FlowNet. We augment the FlowNet (Dosovitskiy *et al.*, 2015) architecture with a ConvLSTM (Shi *et al.*, 2015) after the encoder to form our disparity and flow estimation networks, $\mathcal{D}$ and $\mathcal{O}$. The output of $\mathcal{D}$ and $\mathcal{O}$ consist of 1 and 2 channels respectively.

6.4.2 Implementation details

For training our proposed algorithm, we first obtain a LF *image* dataset proposed by Kalantari *et al.* (2016). Assuming a static scene, we generate stereo videos by simulating random 6-DoF camera motions through resampling the 4D LF data (Lumentut *et al.*, 2019; Srinivasan *et al.*, 2017a). The dataset contains a total of 125 LF images, and we generate ten videos of five frames each from each LF image. The camera motion for each of the ten videos is randomly sampled from a pool of 40 simulated camera motions. Hence, in total, we have 1250 stereo video sequences, each with five frames and a spatial resolution of 375×540. While training, we obtain a stereo video of 4 frames and randomly crop a patch of size 128×128 from both left and right image pairs. We further augment the data by shifting the focal plane between $[-5, 5]$ pixels. The neural network is trained using AdamW optimizer (Loshchilov and Hutter, 2017) for 200 epochs with an initial learning rate of 0.0001. We use We use Pytorch (Paszke *et al.*, 2019a) for all our neural network experiments. The initial learning rate is decreased by $1.1\times$ when the validation loss plateaus for more than 10 epochs. We empirically choose the hyperparameters as $\lambda_1 = 1$, $\lambda_2 = 1$, $\lambda_3 = 0.1$, $\lambda_4 = 1$, $\lambda_5 = 0.1$ and $\lambda_6 = 0.01$ in Eq. (6.12).

6.4.3 Generating stereo *video* from a 4D LF *image*

Consider a 4D LF image of the form $L(x, y, u, v)$ where (x, y) are the spatial co-ordinates and (u, v) are the angular co-ordinates. While simulating the video sequence, we assume a model of multiple pinhole cameras located at the co-ordinates (u, v) from which individual views of the LF are captured. Simulating a camera motion through the given LF is equivalent to resampling the given 4D light-field function and projecting it to the desired camera location (Lumentut *et al.*, 2019). We consider the 6-DoF camera motion with translation and rotation defined as $P(t) = [p_x(t), p_y(t), p_z(t)]$ and $R(t) = [\theta_x(t), \theta_y(t), \theta_z(t)]$, respectively. We consider the stereo camera located at the two views $(0, v_m)$ and (U, v_m). For the given 6-DoF translation and rotation $P(t)$ and

$R(t)$ respectively, the left view at time t is given by,

$$\mathbf{L}_t\left(\mathbf{u}_l\right) = L(x^j, y^j, p_x^i(t) - x^j p_z(t), v_m + p_y^i(t) - y^j p_z(t)) \tag{6.14}$$

$$x^j = (x - U/2)\cos\theta_z(t) - y\sin(\theta_z(t)) + U/2 \tag{6.15}$$

$$y^j = (x - U/2)\sin\theta_z(t) + y\cos(\theta_z(t)) \tag{6.16}$$

$$p_x^i(t) = p_x(t) + f\theta_x(t) \tag{6.17}$$

$$p_y^i(t) = p_y(t) + f\theta_y(t) \tag{6.18}$$

where f is the focal length of the camera. Similarly, the right view of the camera is given by,

$$\mathbf{L}_t\left(\mathbf{u}_r\right) = L(x^j, y^j, U + p_x^i(t) - x^j p_z(t), v_m + p_y^i(t) - y^j p_z(t)) \tag{6.19}$$

We refer the readers to (Lumentut *et al.*, 2019) for a detailed derivation of the above equations.

While we only require stereo videos for training, we require ground-truth LF video in order to quantitatively evaluate the estimated LF videos during testing. For this, we generate full 5D LF videos from a single 4D LF image. The LF video generation process follows that of the stereo video generation. The video generation process described above is repeated across all the views of the LF instead of just 2 extreme views for the stereo video.

6.5 Experiments

To validate our proposed algorithm, we perform various experiments on a variety of datasets. For quantitative comparison against the ground truth, we use the *Raytrix* dataset comprising of ground truth LF videos acquired using an industrial LF camera (Guillo *et al.*, 2018). However, this dataset has only three video sequences with limited scene diversity and a limited angular resolution of 5×5 views. Moreover, it has a maximum disparity of < 2 pixels between adjacent views at a spatial resolution of 1080×1920. Hence, to further validate our proposed algorithm, we test on challenging video sequences from the *Hybrid* video data from (Wang *et al.*, 2017). Furthermore, to include more diversity in the scenes, we simulate videos from 15 LF images in the test set of (Kalantari *et al.*, 2016) and call this diverse dataset *ViewSynth*. While testing, we obtain the stereo sequences from these datasets and provide them as an input to the network $\mathcal{V}$ and generate the LF sequences. Note that, during inference, we do not require estimation of disparity and optical flow maps from $\mathcal{D}$ and $\mathcal{O}$.

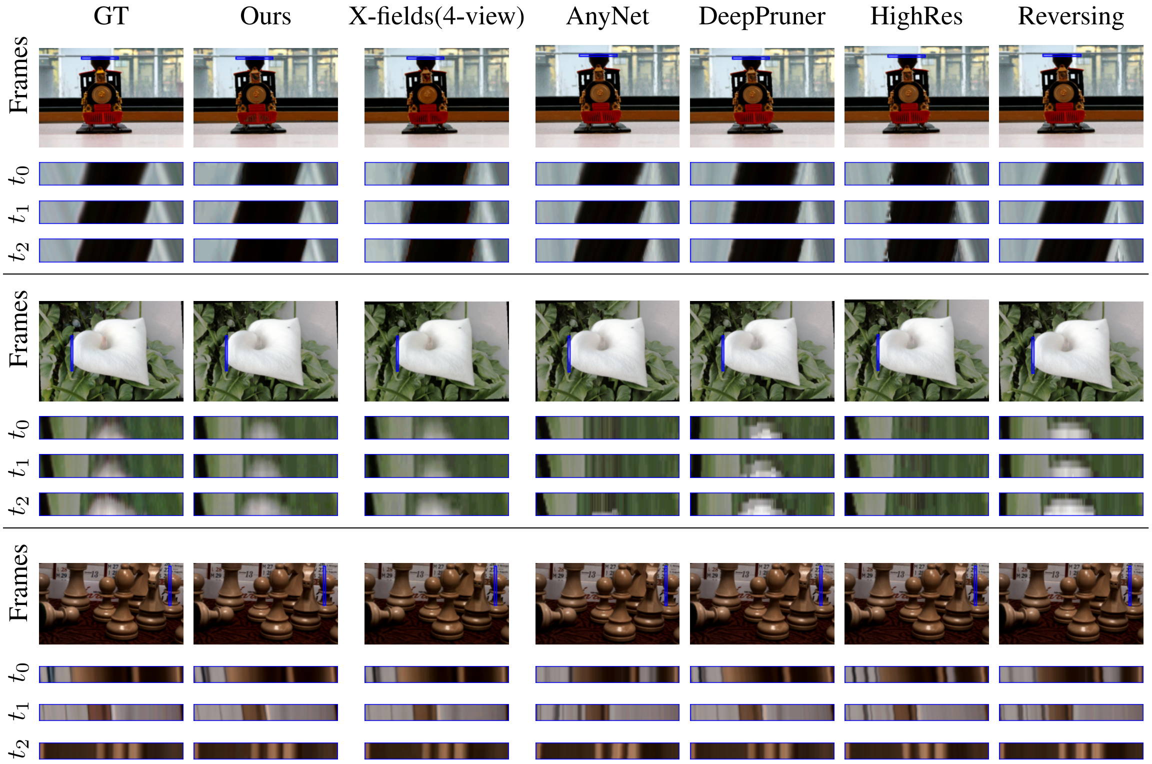


Fig. 6.5: Qualitatively, our algorithm out-performs disparity-based LF prediction techniques. Our proposed algorithm also performs on par with unsupervised LF prediction technique that requires 4 corner views as input.

6.5.1 LF video reconstruction

We compare the accuracy of our proposed algorithm with self-supervised (Bemana *et al.*, 2020) and disparity based methods (Zhang *et al.*, 2015; Wang *et al.*, 2019*a*; Yang *et al.*, 2019; Tankovich *et al.*, 2020; Aleotti *et al.*, 2020; Duggal *et al.*, 2019). For each LF test video, we extract a stereo pair from each frame of the sequence. We consider the two extreme SAIs of the central row of the 4D LF frame as the stereo input to our algorithm to estimate the corresponding 4D LF video. We compare our proposed

Datasets	Hybrid		ViewSynth		Raytrix		Average	
Model	PSNR	LPIPS	PSNR	LPIPS	PSNR	LPIPS	PSNR	LPIPS
AnyNet (Wang *et al.*, 2019*a*)	27.59	0.070	14.88	0.181	16.35	0.251	19.61	0.167
DeepPruner (Duggal *et al.*, 2019)	30.49	0.068	21.35	0.094	30.98	0.064	27.61	0.075
HighRes (Yang *et al.*, 2019)	30.79	0.069	25.57	**0.063**	32.59	0.057	29.65	**0.063**
HITNet (Tankovich *et al.*, 2020)	30.78	0.070	25.51	**0.078**	32.71	**0.061**	29.67	0.069
Reversing (Aleotti *et al.*, 2020)	29.58	**0.061**	13.51	0.262	14.97	0.247	19.35	0.188
X-fields (Bemana *et al.*, 2020) (2-view)	25.53	0.089	24.58	0.099	31.24	0.089	27.12	0.092
X-fields (Bemana *et al.*, 2020) 4-view	**31.66**	0.076	**28.21**	0.091	**32.66**	0.095	**30.84**	0.087
Ours	**34.21**	**0.054**	**30.10**	0.122	**35.57**	**0.045**	**33.29**	**0.071**

Table 6.3: A quantitative comparison of our algorithm against existing algorithms on various datasets. We show that our method outperforms existing methods for self-supervised LF video synthesis. Note that the first five methods require warping. **Red** and **Blue** represent the first and second best algorithm in each column.

Model	Wang *et al.* (2019*a*)	Duggal *et al.* (2019)	Tankovich *et al.* (2020)	Aleotti *et al.* (2020)	Yang *et al.* (2019)	Bemana *et al.* (2020) (2-view)	Bemana *et al.* (2020) (4-view)	Ours
Error $(\times 10^{-2})$	2.576	2.504	2.493	2.538	2.429	3.103	**1.725**	**1.580**

Table 6.4: Mean absolute error (lower is better) obtained after warping successive predicted LF frames via optical flow computed from ground truth LF frames. Our proposed algorithm shows better temporal consistency than other algorithms.

self-supervised algorithm with X-fields (Bemana *et al.*, 2020), also an unsupervised algorithm. In X-fields, each novel view of the LF is estimated using the 3 neighboring views with the network capacity multiplier set to 8 (a higher value resulted in GPU memory error). Since X-fields aims at interpolating views, it fails to generate the full 4D LF from only the stereo views as input (X-fields (2-view). Hence, for X-fields (2-view), we employ a trick of duplicating the stereo pair as the corner views of the LF with the baseline in the v axis of the LF set to zero. For completeness in our comparisons, we include results for LF generation with the four corner views as input (X-fields

(4-view)). We mainly compare our algorithm against the *X-fields (4-view)* variant and use the *(2-view)* variant as only a baseline. We observe that even 4-views variant fails to generate good LF reconstruction in some challenging cases.

We also compare with disparity-based unsupervised LF estimation approach (Zhang *et al.*, 2015), which reconstructs LF via disparity-based warping. Without access to the implementation of (Zhang *et al.*, 2015), we first estimate the disparity from learning-based methods and warp the input views to the LF. We use several state-of-the-art supervised (AnyNet (Wang *et al.*, 2019*a*), HighRes (Yang *et al.*, 2019), DeepPruner (Duggal *et al.*, 2019), HITNet (Tankovich *et al.*, 2020)) and unsupervised (Reversing (Aleotti *et al.*, 2020)) stereo disparity estimation algorithms for comparison. Using this disparity, the input views are then warped to the LF views with the assumption that disparity remains the same in both horizontal and vertical directions. Due to the small baseline of the input views, there are no large holes in the output frames and the small holes due to warping are filled via interpolation.

For quantitative comparison, we used two metrics: PSNR (higher is better) and learned perceptual similarity (LPIPS) (Zhang *et al.*, 2018*b*) (lower is better). Table 6.3 details the quantitative comparisons of various algorithms against all 3 datasets: Raytrix, Hybrid, and ViewSynth. When compared to algorithms that use only 2-views as input, our algorithm outperforms in terms of PSNR. Other algorithms have a slightly better LPIPS (Zhang *et al.*, 2018*b*) metric as their output is just a warped input image and hence tend to be much sharper than the ones generated from our algorithm. However, we can see the real distinction when we compare the images qualitatively in Fig. 6.5 and especially take into account the EPI for the LF views. Algorithms dependent on disparity-based warping suffer from artifacts arising from incorrect disparity estimation, as seen in Fig. 6.5. Our proposed algorithm performs consistently better in predicting LF frames as can be seen from both Table 6.3 and Fig. 6.5.

Temporal consistency Our proposed algorithm aims to reconstruct LF *video* sequences, where temporal consistency between successive LF frames is a crucial factor. To compare temporal consistency of the predicted videos, we require ground-truth optical flow for the ground-truth LF videos. However, it's almost impossible to obtain ground-truth optical flow for real video sequences. Hence, we utilize state-of-the-art algorithm RAFT (Teed and Deng, 2020) to predict optical flow for individual ground-truth LF frames. The optical flow is computed between corresponding angular views in the successive LF frames. We call this estimated optical flow ‘pseudo-ground-truth optical flow’ as it serves as a proxy to the hard-to-acquire true optical flow between successive ground-truth LF frames. Then the mean absolute error is computed after warping successive predicted frames via the estimated pseudo-ground-truth optical flow. As can be seen in Table 6.4 our algorithm shows much better temporal consistency.

6.5.2 Ablation Study

Effect of various loss terms In Table 6.5, we quantitatively compare our proposed model with its variants based on the loss terms in Eq. (6.12). The loss terms, $\mathcal{L}_{stereo}$ and $\mathcal{L}_{temp}$ do not have a significant effect on the model performance, but are still important to ensure the photometric and temporal consistencies. Enforcing the epipolar geometric consistency via $\mathcal{L}_{geo}$ is crucial for our task as we observe a significant performance drop in $V3$. However, between $V3$ and $V4$ we observe that the structure imposed by TD layer helps in obtaining reasonable accuracy even in the absence of $\mathcal{L}_{geo}$ term. When using the $\mathcal{L}_{geo}$ constraint, the performance of both without and with TD model, $V5$ and $Ours$ respectively, is enhanced. For $V5$, we modify $\mathcal{V}$ to output 49 RGB frames corresponding to each view of the 7×7 LF frame. Between $V5$ and our proposed model, we observe a PSNR gain of ~1.9dB due to the layered LF-display-based intermediate representation.

In Fig. 6.6, we make qualitative comparisons for some of the important model variants in Table 6.5. As we observe from the EPI in Fig. 6.6 most of the reconstructed frames in $V4$ are zero due to the absence of both the low-rank representation $\mathcal{F}$ and the geometric consistency term $\mathcal{L}_{geo}$. This shows the importance of the intermediate representation $\mathcal{F}$ in the absence of geometric consistency cost, $\mathcal{L}_{geo}$. Comparing $V3$ and Ours, the importance of the epipolar consistency term $\mathcal{L}_{geo}$ is demonstrated. In $V3$, the layered LF-display-based representation $\mathcal{F}$ imposes the inherent grid-like structure of LF on the output. This ensures that the output frames are reasonably close to the actual LF frames. However, the additional geometric consistency term $\mathcal{L}_{geo}$ in our proposed model provides accurate reconstructions as can be seen from the EPI.

Efficacy of layered-display regularizer We study the effect of varying rank configurations ($M = [1, 3, 6, 9, 12]$) for the low-rank representation $\mathcal{F}$, with number of layers fixed to $L = 3$ (Wetzstein *et al.*, 2012; Maruyama *et al.*, 2019; Takahashi *et al.*, 2018). The quantitative comparison is shown in Table 6.6 for 7×7 angular resolution LF output. While the PSNR improves with increasing rank, we also observe a corresponding increase in time complexity. Hence, we use a rank of $M = 12$ for the representation $\mathcal{F}$ in all our experiments, unless stated otherwise. As seen from Table 6.6, direct regression of LF frame provides the computational advantage but under-performs in terms of PSNR of the output LF. We also see from Fig. 6.7 that the intermediate representation helps obtain sharper LF reconstructions.

In Fig. 6.8 we qualitatively compare the reconstruction performance in the presence (Ours) and absence ($V5$ in Table 6.5) of the intermediate low-rank representation $\mathcal{F}$. The training is done with the full loss function as described in Eq. (6.12) of the manuscript. We observe that the reconstructed LF frames are significantly blurred when we directly predict the LF frame as the output of the network $\mathcal{V}$.

Model	TD	$\mathcal{L}_{geo}$	$\mathcal{L}_{temp}$	$\mathcal{L}_{stereo}$	PSNR
$V1$	✔	✔	✔	✗	32.20
$V2$	✔	✔	✗	✔	31.98
$V3$	✔	✗	✔	✔	19.20
$V4$	✗	✗	✔	✔	6.04
$V5$	✗	✔	✔	✔	30.50
Ours	✔	✔	✔	✔	32.39

Table 6.5: Ablation study of the proposed model with various loss terms from Eq. (6.12)

Metric	Rank of $\mathcal{F}$ (Layers=3)					$V5$
	1	3	6	9	12	–
PSNR	31.43	32.21	31.87	31.92	32.39	30.50
Time	0.103	0.167	0.248	0.319	0.381	0.108

Table 6.6: Quantitative comparison of the efficacy of the proposed layered-display regularizer. $V5$, as shown in Table 6.5, refers to the model where the LF frame is directly output from $\mathcal{V}$ instead of through the intermediate representation $\mathcal{F}$.

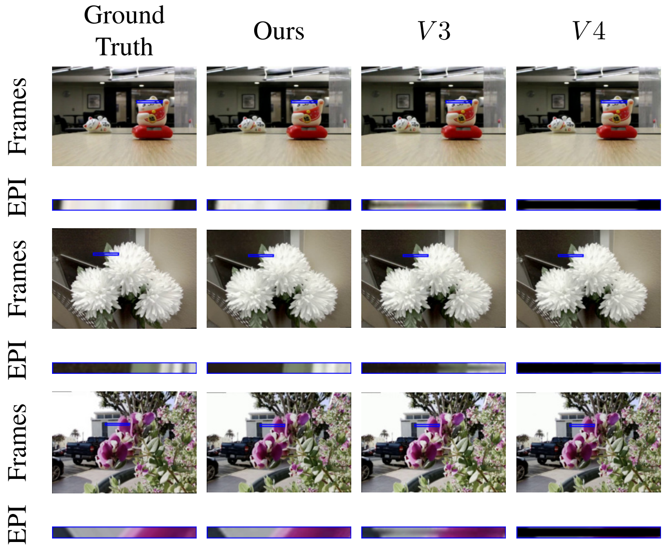


Fig. 6.6: We show the qualitative comparisons for two important configurations of our proposed network architecture, $V3$ and $V4$. The low-rank representation $\mathcal{F}$, inherently imposes the structure of LF in $V3$ producing reasonable reconstructions. However, in the absence of the representation $\mathcal{F}$, most frames predicted by $V4$ are zero. Further, enforcing explicit geometric consistency via $\mathcal{L}_{geo}$ produces significantly better reconstructions as can be seen in the second column.

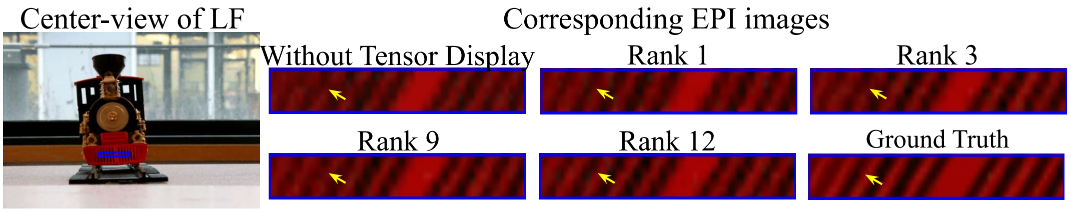


Fig. 6.7: Predicted LF frames are sharper when using higher rank than at lower ranks or not using the low-rank representation at all.

6.5.3 Variable Angular View Prediction

A supervised algorithm for LF prediction is limited to predict the angular views at the angular co-ordinates present in the ground-truth data. However, such a restriction does not exist for our proposed self-supervised algorithm. We demonstrate this with two experiments. First, we train our proposed algorithm to generate various angular resolution LF frames such as 3×3, 5×5, 7×7, and 9×9. We show the results in Fig. 6.9. Commercially available LF cameras such as Lytro capture 14×14 angular resolution images. While only the central 8×8 of those views are actually usable, our algorithm allows us to generate LF sequences with higher number of angular views

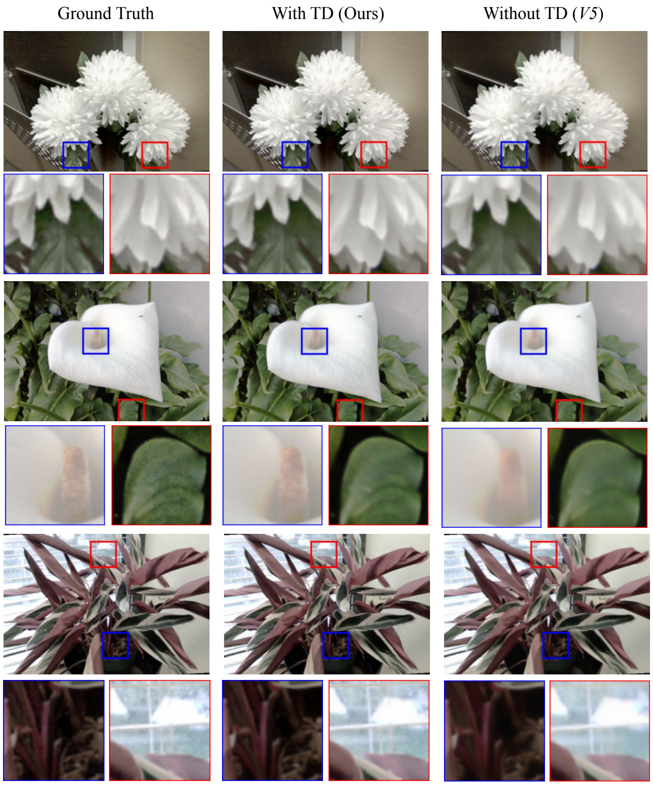


Fig. 6.8: We qualitatively compare the reconstruction performance in the presence (With TD) and absence (Without TD) of the intermediate low-rank representation $\mathcal{F}$. We observe significant blurring in the reconstructed images when not using the low-rank representation.

such as 9×9.

Next, we demonstrate our algorithm's capability for extrapolating the angular views to new views outside of the input baseline. Throughout our experiments, we assume that the input stereo views correspond to the extreme views of the predicted LF frame. For extrapolating the views beyond the input baseline, we employ a simple trick: the input stereo views are now assumed to correspond to adjacent horizontal views of the predicted LF frame. We show qualitative results in Fig. 6.10 where the EPI of the *extended* images show increased slopes. This indicates an increased disparity between adjacent views of the *extended* images than that of the *original* images.

6.5.4 Fine-tuning on test sequences

The training procedure for our algorithm is to minimize the overall cost function, Eq. (6.12), while jointly estimating the LF video, disparity, and optical flow maps from the

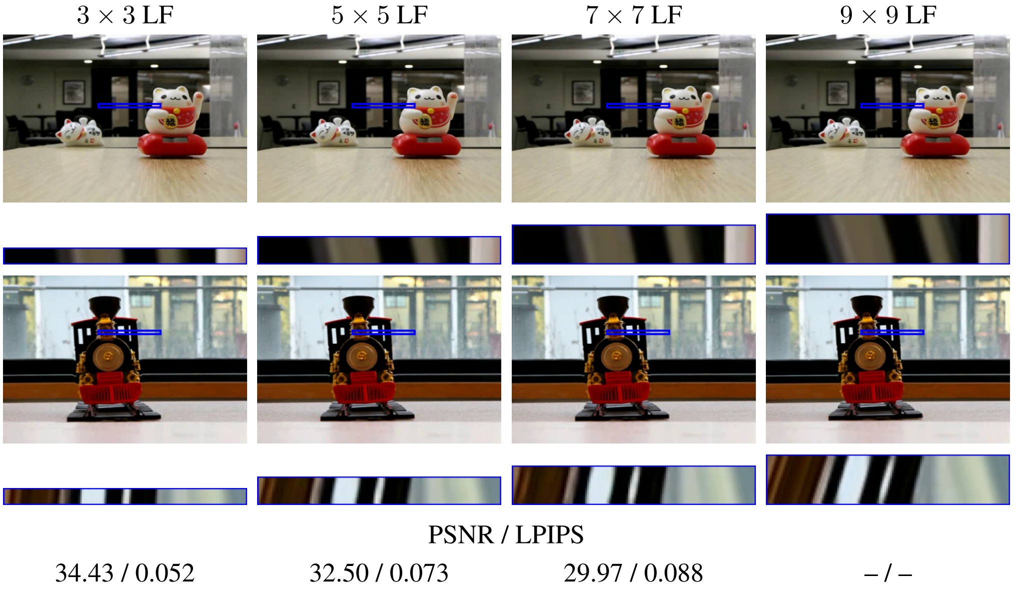


Fig. 6.9: The proposed algorithm can be used for prediction of variable number of angular views between the input stereo sequence. Above, we show the LF sequence predicted at angular resolutions of $3, 5, 7, 9$ and also provide the PSNR / LPIPS metrics where ground truth is available.

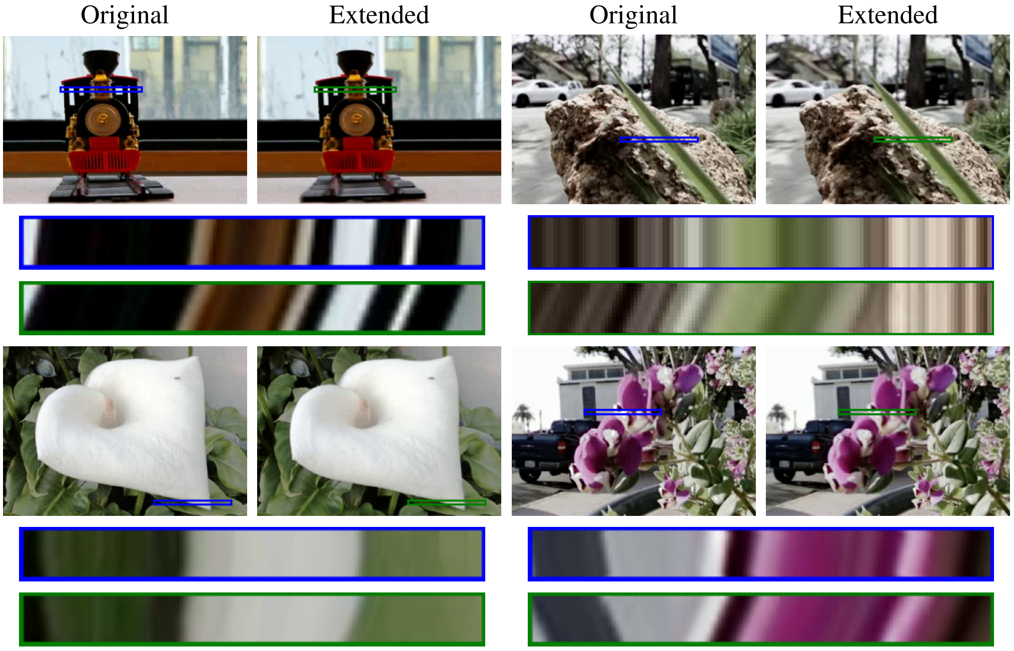


Fig. 6.10: Our proposed self-supervised algorithm can be used to predict novel views beyond the baseline of the input image pair. Larger slopes of EPI images depict larger disparities between adjacent views, indicating an increased baseline between the extreme views.

input stereo video. However, due to domain mismatch, the network can fail to reconstruct reasonable sequences during test time. For such cases, our proposed algorithm

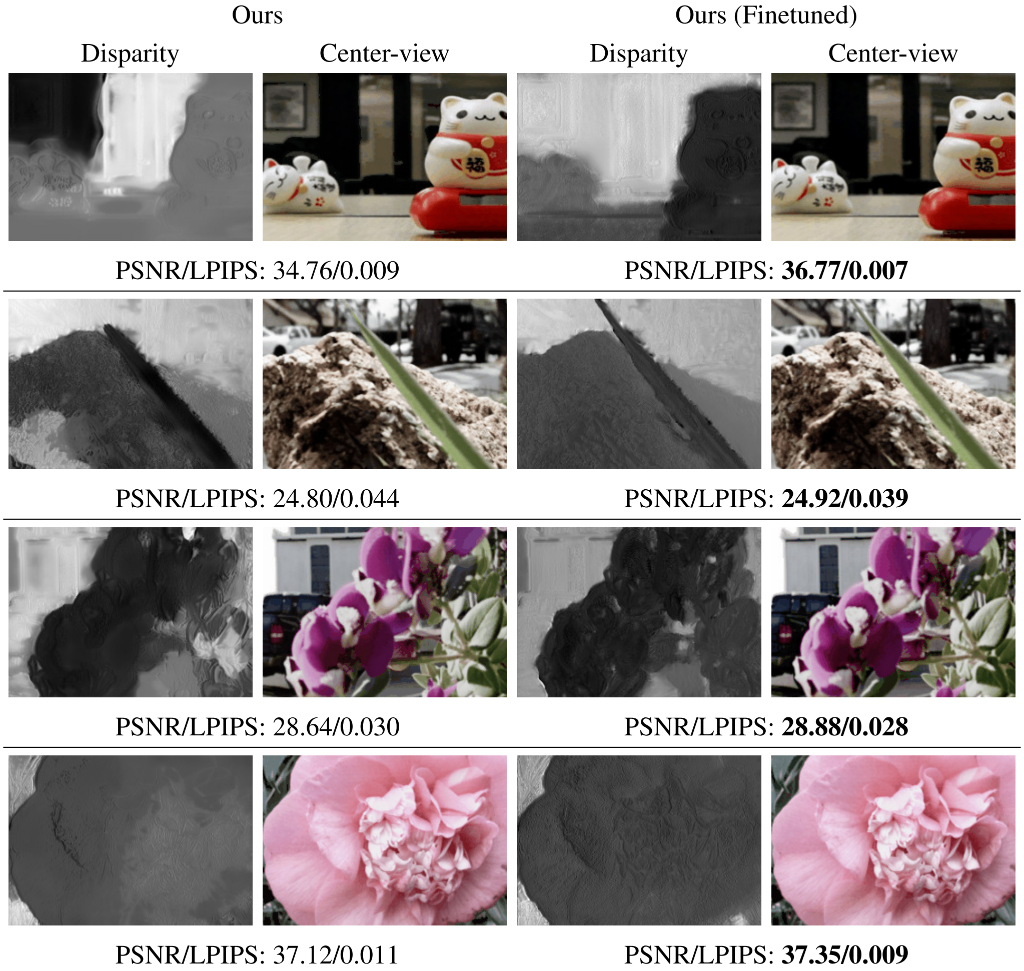


Fig. 6.11: We show the results of finetuning the trained networks on novel test sequences. The first two rows show cases where the network does not perform well initially but we observe significant improvement with finetuning. Overall, we see a consistent improvement in the predictions as the result of finetuning.

allows for *fine-tuning* the neural network on single test sequences. During fine-tuning, the overall cost function in Eq. (6.12) is minimized with AdamW optimizer for 500 iterations. As can be seen from Fig. 6.11, fine-tuning consistently improves the accuracy of the predicted LF sequence while also producing significant qualitative improvements in the reconstructed disparity maps.

6.5.5 Application to video refocusing

We demonstrate the application of the predicted LF videos on post-capture focus control. In Fig. 6.1 we show a video sequence camera from (Urvoy *et al.*, 2012) acquired

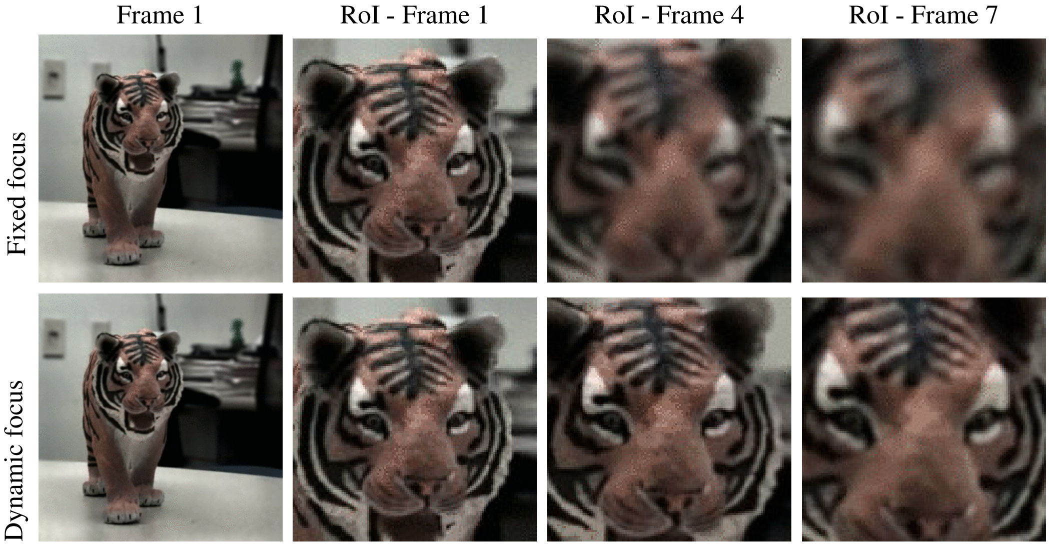


Fig. 6.12: Using the LF generated from our proposed algorithm we show the application to RoI based focus tracking where the focus is dynamically adjusted on the toy.

using a commercial stereoscopic camera. As the video is acquired with a large baseline (6cm) stereoscopic camera, we synthetically reduce the baseline by downsampling the spatial resolution to 270×480 from 1080×1920. The left and right image pairs are rectified using (Xiao *et al.*, 2018) as the calibration files are unavailable. We can see from Fig. 6.1 that the generated LF video from the input stereo sequence can be seamlessly used for post-capture focus control. As our algorithm only requires the stereo pair as input, it can be employed to generate synthetically defocused sequences from smartphones, many of which now come with dual-lens cameras. In Fig. 6.12, we show another instance of post-capture focus control. We extract a stereo video sequence consisting of 8 frames from the LF video dataset in (Wang *et al.*, 2017). The proposed algorithm is used to generate the LF video from the stereo video. The focal plane is fixed in the original video, due to which the toy gets increasingly blurred. However, with our predicted LF video sequence, we can dynamically change the focal plane to be fixed on the toy. This ensures that the object of interest, which is the toy-tiger, remains in focus throughout the video.

6.6 Discussion

Our proposed algorithm can recover perceptually appealing light-field videos from only a stereo video sequence. With only a stereo video input, there is limited knowledge about the objects being disoccluded in the vertical direction. However, occlusions do not pose a huge challenge because we use a relatively small baseline. The proposed algorithm implicitly learns to inpaint the disoccluded regions. One of the ways to handle occlusions would be to exploit long-range temporal correlations in the input video. Another option would be to use a small corpus of training data for supervised training to handle occlusions.

6.6.1 Comparison with X-fields (4-view)

On some sequences, X-fields (4-view) (Bemana *et al.*, 2020) achieves better results for two important reasons. One, X-fields uses both horizontal and vertical disparity information to produce the light-fields. Our technique however has only the horizontal disparity information from the stereo image. Second, it is trained over one particular sequence and is hence expected to perform better. X-fields is certainly a more versatile technique allowing for *interpolation* in time, view and light directions. Our work is a complementary technique to X-fields: we allow for finetuning the reconstruction on a particular sequence, while also utilizing data-driven approaches to improve performance in a way that generalizes well to arbitrary scenes. Our work also provides a technique for *extrapolation*, which X-fields is not designed to handle currently.

6.6.2 Loss in spatial frequency

We observe that there's a loss in spatial details for some of the sequences shown in Fig. 6.13. While we observe blurring in some of our reconstructed sequences, it is not a fundamental limitation of our overall technique. Incorporating detail-preserving losses on top of the low-rank regularizer can preserve high-frequency details. For instance, a low-rank+sparse decomposition model for LF, combined with a perceptual loss, could help recover the high-frequency details. As we see in Fig. 6.13, the spatial frequency details can be restored to a reasonable accuracy.

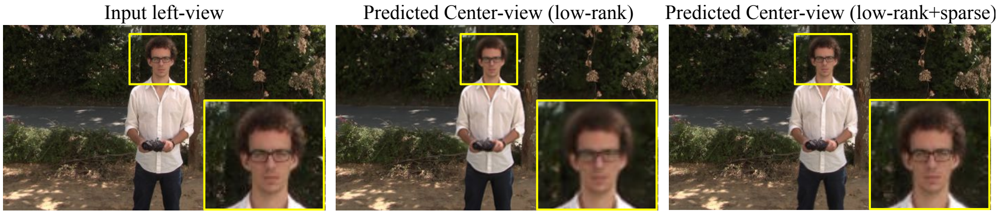

Input left-view
Predicted Center-view (low-rank)
Predicted Center-view (low-rank+sparse)

Fig. 6.13: As seen in middle image, we observe loss of spatial details in the reconstructed frames for some video sequences. However, this is not a fundamental limitation of the proposed model. We observe in the right image that the details can be recovered by the use of detail preserving perceptual cost metrics such as LPIPS (Zhang *et al.*, 2018*b*) and low-rank+sparse decomposition model.

6.7 Conclusion

We propose a self-supervised algorithm for light-field video reconstruction from a stereo video. A layered light field display-based low-rank representation is used as a regularizer for guiding the self-supervised reconstruction of light-field frames. The algorithm is applicable for widespread consumer use because we require only a stereo video as input. The proposed self-supervised algorithm confers advantages over supervised learning, such as post-training fine-tuning on test sequences. Other advantages include variable angular view synthesis both between and beyond the input baseline. The reconstructed light-field videos also enable post-capture focus control applications for video sequences.

CHAPTER 7

Conclusions

In the last two decades, several technological advancements have reduced the cost of image sensors leading to widespread use of cameras. However, some specialized cameras that can acquire high-speed videos and LF videos still remain expensive and out of reach for most consumers. This is because high-speed video and LF video capture requires specialized hardware capable of handling large data bandwidth. Hence, modern cameras do not use such specialized hardware in a bid to keep the sensor costs low. However, high-speed videos and LF videos have several applications in both scientific imaging and consumer photography as we saw in Chapter 1.

This thesis explored the reconstruction of these high data-bandwidth videos from low data-bandwidth measurements acquired from various sensor systems. We proposed three different frameworks for reconstruction of high-speed videos from low data-bandwidth measurements acquired using coded-exposure sensors (Chapter 3) and neuromorphic event sensors (Chapters 4 and 5). We also proposed a learning-based framework for reconstruction of LF video from stereo videos (Chapter 6). The hardware systems proposed in each of the Chapters are also suitable for commercial/consumer use and not limited to controlled lab settings. Along with these hardware systems, we proposed appropriate learning-based frameworks which can boost the performance of the system. Below, we provide major concluding statements for each of the frameworks proposed in this thesis.

7.1 High-speed imaging with coded-exposure sensors

Coded exposure sensors have remained a popular computational imaging technique for high-speed video reconstruction under limited data-bandwidth constraints. There has been a renewed interest in coded exposure imaging as there are novel sensors that make it easy to implement coded-exposure imaging. The effectiveness of deep-learning-based methods for solving inverse imaging problems is also another factor for this renewed interest. While several recent deep-learning works use fully-connected networks for video reconstruction, in our work we show that locally-connected fully-convolutional networks are a better choice for the network architecture. Our proposed unified framework for coded-exposure video reconstruction can reconstruct videos from three different frameworks, namely, flutter-shutter, single pixel-wise coded exposure and coded-2-bucket sensor. Inspired from the simple linear algebraic solution, we demonstrate that using a SVC layer over standard convolutional layer provides high-fidelity reconstructions.

7.1.1 Insights for future work

In coded-exposure imaging, learning the optimal coded exposure sequence for high-fidelity reconstruction has been crucial. Traditionally, exposure sequences of small spatial size such as 8×8 were tiled across the whole image. However, recently a new train of thought has emerged where the spatial size of the exposure sequence spans the whole sensor size such as 128×128 or 256×256. A thorough investigation can be carried out into these choices as whether to use repeated tiles or let the exposure sequence span the whole image. Our proposed unified framework does provide a ready platform for such a comparison and our open-source code can be utilized for the same.

In almost all the previous works, the exposure sequence remains fixed regardless of the scene content and its attributes. However, an exposure sequence that adapts to the scene attributes might provide better reconstruction and noise performance. For instance, in regions where the scene is static, one need not do any coding and just keep the exposure open to collect more light and reduce noise while also not trading off on blur. Some very recent work in this direction using reinforcement learning has shown promise in this idea (Lu *et al.*, 2021).

A theoretical analysis on the optimality of a coded exposure sequence was provided for image deblurring in (Raskar *et al.*, 2006). Using a simple linear system inversion, Raskar *et al.* (2006) concluded that exposure sequences with a broadband frequency response are the most optimal for well-posed image deblurring. However, when using data-driven techniques for video recovery from coded-exposure images, the reconstruction accuracy is dependent on the code as well as the neural network design. This was demonstrated in Chapter 3 where reconstruction accuracy improved with better neural network design while keeping the code fixed. Determining the association between the two factors: code-design and network design, for better reconstruction could be a promising future direction. In decoupling the influence of code-design and neural networks on reconstruction, it might be useful to also study the properties of an optimal coded exposure mask. If these properties are also differentiable they could be incorporated as constraints into the neural network training process to obtain the best reconstruction performance.

7.2 High-speed imaging with event sensors

In comparison to coded-exposure imaging, event sensors are a novel kind of sensors. Their advantages such as low-power, low-latency and low-bandwidth operation, along with high dynamic range capabilities has attracted attention for various applications. In this thesis, we explored the application of event sensors in high-speed imaging under limited data-bandwidth constraints. First, we proposed a hybrid sensor framework consisting of co-located event and intensity sensors. With this hybrid sensor framework, we proposed a pipeline for reconstruction of high-speed photorealistic intensity images. In this pipeline, temporally dense event-sensor information was used to warp the low frame-rate intensity frames to temporally dense locations. The pipeline utilized a hybrid solution framework consisting of optimization with initialization from pretrained deep-learning frameworks. With our proposed framework, we achieved up to $60\times$ frame upsampling from the base frame-rate.

Our proposed hybrid-sensor-based framework ignored the high dynamic range nature of the event sensors. Hence, we next propose a learning-based pipeline for high frame-rate, high dynamic range intensity frame reconstruction from event sensors. Our proposed semi-supervised learning based technique can learn from real event sensor data, with real event sensor noise. This helps it to generalize to different scenes acquired with different sensors and different environment conditions such as indoor-outdoor, day-night, *etc.* We also achieve a temporal super-resolution of up to $100\times$ with our proposed technique.

7.2.1 Insights for future work

Event sensors are now finding their way into smartphones along with other novel sensors. On these devices, event sensors can first and foremost be used for high-speed imaging without having to worry about excessive power and memory consumption. However, in consumer market, image quality is also of utmost importance. Our technique of high frame-rate and high-dynamic range video from events, falls short in providing superior image quality as it only has access to events and not the texture-rich intensity images. Our photorealistic image reconstruction framework is very suitable in this scenario. However, the algorithm does not generalize to dynamic scenes as it involved depth-based image rendering. Hence, a novel framework that utilizes the complementary advantages of the event sensor and the intensity sensor for high-speed, high-dynamic range video reconstruction is a very promising future direction. Preliminary results from high-speed video reconstruction from such hybrid sensor was shown very recently by Tulyakov *et al.* (2021). However, incorporating high-dynamic range nature of event sensors as well into the reconstruction framework remains to be investigated.

The event sensor technology has advanced quite a lot over the last 5 years. Throughout this progress, the underlying basic mechanism of event generation has remained the same (change in observed intensity). However, there has not yet been a good deterministic (or even probabilistic) model relating the intensity change and event generation. Previous works that use only an approximate mathematical model do not generate very accurate results. The absence of any noise model for the generated events also pose significant challenge to these algorithms. This has lead to the use of deep-learning methods that use data-driven techniques to implicitly bypass modeling the event firing and its associated noise. Hence, addressing these two challenges: a) accurate mathematical model for event firing and b) an accurate noise model for events, can inform us on designing efficient algorithms to process event sensor data.

7.3 Light-field video from stereo videos

LF imaging has been very popular for its applications in novel view synthesis, post-capture focus control and post-capture aperture control. However, LF video acquisition in commercial/consumer devices has been challenging due to the requirement of large bandwidth and specialized hardware. Nowadays, several commercial and mobile devices come with stereo cameras that acquire stereo videos. Stereo videos can be thought of as a sparse, low-bandwidth sample of the LF video. Hence, we explored the utility of these stereo cameras for LF video reconstruction. We proposed a novel self-supervised learning-based framework for the reconstruction of LF videos from stereo video sequences. Our proposed technique utilizes the low-rank tensor-display based framework for regularizing the LF prediction and achieve better performance. We demonstrate reconstruction of LF video from stereo videos obtained from commercial stereo cameras. The reconstructed LF videos from our technique show applications in novel view synthesis and post-capture focus control for videos.

7.3.1 Insights for future work

With our technique, we demonstrated that by using the scene geometry information and the low-rank LF regularization, we can reconstruct LF videos with high fidelity. However, it has been demonstrated that we can obtain scene geometry from not just stereo, but from other sources as well. These sources could be monocular videos where we can estimate depth via structure-from-motion technique. More recently dual-pixel sensors in smartphones have also been shown to be useful in estimating depth maps. Exploiting this, we can also reconstruct LF videos from monocular videos and dual-pixel videos. These configurations are attractive because one can easily acquire monocular and dual-pixel videos from most modern smartphones. This will open up new possibilities for the application of LF imaging on mobile devices and smartphones.

To quantitatively evaluate the accuracy of the reconstructed LF we use metrics such as PSNR and LPIPS. However, these metrics are not suitable for evaluating 4D LF images/videos as they do not take into account the dependence between the angular views. Currently, there are also no widely used metrics to quantitatively evaluate the LF reconstruction accuracy. Hence, formulating a quantitative metric that can evaluate the accuracy of a reconstructed LF with the ground-truth is a promising research direction.

REFERENCES

1. **Agrawal, A.**, **M. Gupta**, **A. Veeraraghavan**, and **S. G. Narasimhan** (2010). Optimal coded sampling for temporal super-resolution. *In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE. 21

2. **Aleotti, F.**, **F. Tosi**, **L. Zhang**, **M. Poggi**, and **S. Mattoccia** (2020). Reversing the cycle: self-supervised deep stereo through enhanced monocular distillation. *In European Conference on Computer Vision*. Springer. 96, 97, 98

3. **Andreopoulos, A.**, **H. J. Kashyap**, **T. K. Nayak**, **A. Amir**, and **M. D. Flickner** (2018). A low power, high throughput, fully event-based stereo system. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 42

4. **Anupama, S.**, **P. Shedligeri**, **A. Pal**, and **K. Mitra** (2021). Video reconstruction by spatio-temporal fusion of blurred-coded image pair. *In 2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 4, 18

5. **Asif, M. S.**, **A. Ayremlou**, **A. Sankaranarayanan**, **A. Veeraraghavan**, and **R. G. Baraniuk** (2016). Flatcam: Thin, lensless cameras using coded aperture and computation. *IEEE Transactions on Computational Imaging*, **3**(3), 384–397. 24

6. **Bae, K.**, **A. Ivan**, **H. Nagahara**, and **I. K. Park** (2021). 5d light field synthesis from a monocular video. *In 2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 7, 85

7. **Baraniuk, R. G.**, **T. Goldstein**, **A. C. Sankaranarayanan**, **C. Studer**, **A. Veeraraghavan**, and **M. B. Wakin** (2017). Compressive video sensing: algorithms, architectures, and applications. *IEEE Signal Processing Magazine*, **34**(1), 52–66. 17, 21

8. **Bardow, P.**, **A. J. Davison**, and **S. Leutenegger** (2016). Simultaneous optical flow and intensity estimation from an event camera. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5, 40, 43, 56, 59, 60

9. **Barron, J. T.** and **B. Poole** (2016). The fast bilateral solver. *In European Conference on Computer Vision*. Springer. 49

10. **Barua, S.**, **Y. Miyatani**, and **A. Veeraraghavan** (2016). Direct face detection and video reconstruction from event cameras. *In 2016 IEEE winter conference on applications of computer vision (WACV)*. IEEE. 5, 40, 43

11. **Bemana, M.**, **K. Myszkowski**, **H.-P. Seidel**, and **T. Ritschel** (2020). X-fields: implicit neural view-, light-and time-image interpolation. *ACM Transactions on Graphics (TOG)*, **39**(6), 1–15. 82, 83, 84, 85, 96, 97, 106

12. **Bergen, J. R.** and **E. H. Adelson** (1991). The plenoptic function and the elements of early vision. *Computational models of visual processing*, **1**, 8. 1

13. **Berner, R.**, **C. Brandli**, **M. Yang**, **S.-C. Liu**, and **T. Delbruck** (2013). A 240× 180 10mw 12μs latency sparse-output vision sensor for mobile applications. *In Symposium on VLSI Circuits (VLSIC)*. 40, 49

14. **Brandli, C.**, **R. Berner**, **M. Yang**, **S.-C. Liu**, and **T. Delbruck** (2014). A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, **49**(10), 2333–2341. 6, 7, 58, 61, 65

15. **Bryner, S.**, **G. Gallego**, **H. Rebecq**, and **D. Scaramuzza** (2019). Event-based, direct camera tracking from a photometric 3d map using nonlinear optimization. *In IEEE Int. Conf. Robot. Autom.(ICRA)*, volume 2. 42

16. **Delbrück, T.**, **B. Linares-Barranco**, **E. Culurciello**, and **C. Posch** (2010). Activity-driven, event-based vision sensors. *In Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. IEEE. 56

17. **Dosovitskiy, A.**, **P. Fischer**, **E. Ilg**, **P. Hausser**, **C. Hazirbas**, **V. Golkov**, **P. Van Der Smagt**, **D. Cremers**, and **T. Brox** (2015). Flownet: Learning optical flow with convolutional networks. *In Proceedings of the IEEE international conference on computer vision*. xvii, 87, 92, 93

18. **Duggal, S.**, **S. Wang**, **W.-C. Ma**, **R. Hu**, and **R. Urtasun** (2019). Deeppruner: Learning efficient stereo matching via differentiable patchmatch. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*. 96, 97, 98

19. **Farrugia, R. A.** and **C. Guillemot** (2019). Light field super-resolution using a low-rank prior and deep convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, **42**(5), 1162–1175. 84

20. **Flynn, J.**, **M. Broxton**, **P. Debevec**, **M. DuVall**, **G. Fyffe**, **R. Overbeck**, **N. Snavely**, and **R. Tucker** (2019). Deepview: View synthesis with learned gradient descent. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 84

21. **Gallego, G.**, **T. Delbruck**, **G. Orchard**, **C. Bartolozzi**, **B. Taba**, **A. Censi**, **S. Leutenegger**, **A. Davison**, **J. Conradt**, **K. Daniilidis**, *et al.* (2019*a*). Event-based vision: A survey. *arXiv preprint arXiv:1904.08405*. 43

22. **Gallego, G.**, **M. Gehrig**, and **D. Scaramuzza** (2019*b*). Focus is all you need: Loss functions for event-based vision. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 42

23. **Gallego, G.**, **H. Rebecq**, and **D. Scaramuzza** (2018). A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 42, 59

24. **Gers, F. A.**, **J. Schmidhuber**, and **F. Cummins** (1999). Learning to forget: continual prediction with lstm. *In ICANN*, volume 2. ISSN 0537-9989, doi:10.1049/cp:19991218. 61

25. **Godard, C.**, **O. Mac Aodha**, and **G. J. Brostow** (2017*a*). Unsupervised monocular depth estimation with left-right consistency. *In CVPR*. 86, 88

26. **Godard, C.**, **O. Mac Aodha**, and **G. J. Brostow** (2017*b*). Unsupervised monocular depth estimation with left-right consistency. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 88

27. **Godard, C.**, **O. Mac Aodha**, **M. Firman**, and **G. J. Brostow** (2019*a*). Digging into self-supervised monocular depth estimation. *In Proceedings of the IEEE International Conference on Computer Vision*. 64

28. **Godard, C.**, **O. Mac Aodha**, **M. Firman**, and **G. J. Brostow** (2019*b*). Digging into self-supervised monocular depth estimation. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*. 88

29. **Gu, J.**, **Y. Hitomi**, **T. Mitsunaga**, and **S. Nayar** (2010). Coded rolling shutter photography: Flexible space-time sampling. *In 2010 IEEE International Conference on Computational Photography (ICCP)*. IEEE. 11, 17

30. **Guillo, L.**, **X. Jiang**, **G. Lafruit**, and **C. Guillemot** (2018). Light field video dataset captured by a r8 raytrix camera. *Tech. rep., ISO/IEC JTC1/SC29/WG1 & WG11*. 95

31. **Guo, M.**, **H. Zhu**, **G. Zhou**, and **Q. Wang** (2018). Dense light field reconstruction from sparse sampling using residual network. *In Asian Conference on Computer Vision*. Springer. 84

32. **Gupta, M.**, **A. Jauhari**, **K. Kulkarni**, **S. Jayasuriya**, **A. Molnar**, and **P. Turaga** (2017). Compressive light field reconstructions using deep learning. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 84

33. **Haessig, G.**, **A. Cassidy**, **R. Alvarez**, **R. Benosman**, and **G. Orchard** (2018). Spiking optical flow for event-based sensors using ibm's truenorth neurosynaptic system. *IEEE transactions on biomedical circuits and systems*, **12**(4), 860–870. 59

34. **Hajisharif, S.**, **E. Miandji**, **C. Guillemot**, and **J. Unger** (2020). Single sensor compressive light field video camera. *In Computer Graphics Forum*, volume 39. Wiley Online Library. 7, 84, 85

35. **He, K.**, **X. Zhang**, **S. Ren**, and **J. Sun** (2016). Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. 65, 92

36. **Heeger, D. J.** (1996). Notes on motion estimation. 45

37. **Herbst, E.**, **S. Seitz**, and **S. Baker** (2009). Occlusion reasoning for temporal interpolation using optical flow. *Department of Computer Science and Engineering, University of Washington, Tech. Rep. UW-CSE-09-08-01*. 17, 20

38. **Holloway, J.**, **A. C. Sankaranarayanan**, **A. Veeraraghavan**, and **S. Tambe** (2012). Flutter shutter video camera for compressive sensing of videos. *In 2012 IEEE International Conference on Computational Photography (ICCP)*. IEEE. 4, 17, 19, 21

39. **Hubara, I.**, **M. Courbariaux**, **D. Soudry**, **R. El-Yaniv**, and **Y. Bengio** (2016). Binarized neural networks. *In Advances in neural information processing systems*. 37

40. **Hui, T.-W.**, **X. Tang**, and **C. Change Loy** (2018). Liteflownet: A lightweight convolutional neural network for optical flow estimation. *In Proceedings of the IEEE conference on CVPR*. 74

41. **Iliadis, M.**, **L. Spinoulas**, and **A. K. Katsaggelos** (2018). Deep fully-connected networks for video compressive sensing. *Digital Signal Processing*, **72**, 9–18. 17, 19, 21

42. **Iliadis, M.**, **L. Spinoulas**, and **A. K. Katsaggelos** (2020). Deepbinarymask: Learning a binary mask for video compressive sensing. *Digital Signal Processing*, **96**, 102591. 4, 17, 18, 19, 21, 24, 36

43. **Inagaki, Y.**, **Y. Kobayashi**, **K. Takahashi**, **T. Fujii**, and **H. Nagahara** (2018). Learning to capture light fields through a coded aperture camera. *In Proceedings of the European Conference on Computer Vision (ECCV)*. 81, 84

44. **Jaderberg, M.**, **K. Simonyan**, **A. Zisserman**, and **K. Kavukcuoglu** (2015). Spatial transformer networks. *arXiv preprint arXiv:1506.02025*. 88

45. **Jason, J. Y.**, **A. W. Harley**, and **K. G. Derpanis** (2016). Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. *In European Conference on Computer Vision*. Springer. 7, 58, 63, 68, 89

46. **Jiang, H.**, **D. Sun**, **V. Jampani**, **M.-H. Yang**, **E. Learned-Miller**, and **J. Kautz** (2018). Super slomo: High quality estimation of multiple intermediate frames for video interpolation. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 17, 20

47. **Jin, M.**, **G. Meishvili**, and **P. Favaro** (2018). Learning to extract a video sequence from a single motion-blurred image. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 20

48. **Kalantari, N. K.**, **T.-C. Wang**, and **R. Ramamoorthi** (2016). Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (TOG)*, **35**(6), 1–10. 82, 84, 94, 96

49. **Khoei, M. A.**, **S.-h. Ieng**, and **R. Benosman** (2019). Asynchronous event-based motion processing: From visual events to probabilistic sensory representation. *Neural computation*, **31**(6), 1114–1138. 59

50. **Kim, H.**, **S. Leutenegger**, and **A. J. Davison** (2016). Real-time 3d reconstruction and 6-dof tracking with an event camera. *In European Conference on Computer Vision*. Springer. 42, 43

51. **Kingma, D. P.** and **J. Ba** (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. 29, 50, 68

52. **Kobayashi, Y.**, **K. Takahashi**, and **T. Fujii** (2017). From focal stacks to tensor display: A method for light field visualization without multi-view images. *In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 86

53. **Lai, W.-S.**, **J.-B. Huang**, **O. Wang**, **E. Shechtman**, **E. Yumer**, and **M.-H. Yang** (2018). Learning blind video temporal consistency. *In Proceedings of the European conference on computer vision (ECCV)*. 83, 93

54. **Li, Q.** and **N. K. Kalantari** (2020). Synthesizing light field from a single image with variable mpi and two network fusion. *ACM Transactions on Graphics (TOG)*, **39**(6), 1–10. 84

55. **Li, Y.**, **M. Qi**, **R. Gulve**, **M. Wei**, **R. Genov**, **K. N. Kutulakos**, and **W. Heidrich** (2020). End-to-end video compressive sensing using anderson-accelerated unrolled networks. *In 2020 IEEE International Conference on Computational Photography (ICCP)*. 4, 17, 18, 19, 21, 28, 30, 31, 32, 36

56. **Lichtsteiner, P.**, **C. Posch**, and **T. Delbruck** (2008). A 128×128 120 db 15μ s latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, **43**(2), 566–576. 39

57. **Liu, C.** *et al.* (2009). *Beyond pixels: exploring new representations and applications for motion analysis*. Doctoral thesis, Massachusetts Institute of Technology. 74

58. **Liu, D.**, **J. Gu**, **Y. Hitomi**, **M. Gupta**, **T. Mitsunaga**, and **S. K. Nayar** (2013). Efficient space-time sampling with pixel-wise coded exposure for high-speed imaging. *IEEE transactions on pattern analysis and machine intelligence*, **36**(2), 248–260. 4, 17, 18, 19, 21, 25

59. **Liu, L.**, **J. Gu**, **K. Z. Lin**, **T.-S. Chua**, and **C. Theobalt** (2021). Neural sparse voxel fields. 84

60. **Liu, M.** and **T. Delbruck** (2018). Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. *British Machine Vision Conference*. 58

61. **Llull, P.**, **X. Liao**, **X. Yuan**, **J. Yang**, **D. Kittle**, **L. Carin**, **G. Sapiro**, and **D. J. Brady** (2013). Coded aperture compressive temporal imaging. *Optics express*, **21**(9), 10526–10545. 4, 17, 18, 19

62. **Loshchilov, I.** and **F. Hutter** (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*. 94

63. **Lu, S.**, **X. Yuan**, **A. K. Katsaggelos**, and **W. Shi** (2021). Reinforcement learning for adaptive video compressive sensing. *arXiv preprint arXiv:2105.08205*. 109

64. **Lumentut, J. S.**, **T. H. Kim**, **R. Ramamoorthi**, and **I. K. Park** (2019). Deep recurrent network for fast and full-resolution light field deblurring. *IEEE Signal Processing Letters*, **26**(12), 1788–1792, doi:10.1109/LSP.2019.2947379. 94, 95

65. **Lumentut, J. S.**, **T. H. Kim**, **R. Ramamoorthi**, and **I. K. Park** (2019). Fast and full-resolution light field deblurring using a deep neural network. *arXiv preprint arXiv:1904.00352*. 94

66. **Mahjourian, R.**, **M. Wicke**, and **A. Angelova** (2018). Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 45

67. **Martel, J. N. P.**, **L. K. Müller**, **S. J. Carey**, **P. Dudek**, and **G. Wetzstein** (2020). Neural sensors: Learning pixel exposures for hdr imaging and video compressive sensing with programmable sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **42**(7), 1642–1653. 4, 17, 18, 19, 24

68. **Maruyama, K.**, **Y. Inagaki**, **K. Takahashi**, **T. Fujii**, and **H. Nagahara** (2019). A 3-d display pipeline from coded-aperture camera to tensor light-field display through cnn. *In 2019 IEEE International Conference on Image Processing (ICIP)*. IEEE. 86, 87, 100

69. **Marwah, K.**, **G. Wetzstein**, **Y. Bando**, and **R. Raskar** (2013). Compressive light field photography using overcomplete dictionaries and optimized projections. *ACM Transactions on Graphics (TOG)*, **32**(4), 1–12. 84

70. **Meister, S.**, **J. Hur**, and **S. Roth** (2018). Unflow: Unsupervised learning of optical flow with a bidirectional census loss. *In Thirty-Second AAAI Conference on Artificial Intelligence*. 7, 58, 63, 86, 89

71. **Mildenhall, B.**, **P. P. Srinivasan**, **R. Ortiz-Cayon**, **N. K. Kalantari**, **R. Ramamoorthi**, **R. Ng**, and **A. Kar** (2019). Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, **38**(4), 1–14. 84

72. **Mildenhall, B.**, **P. P. Srinivasan**, **M. Tancik**, **J. T. Barron**, **R. Ramamoorthi**, and **R. Ng** (2020). Nerf: Representing scenes as neural radiance fields for view synthesis. *In ECCV*. 84

73. **Mueggler, E.**, **H. Rebecq**, **G. Gallego**, **T. Delbruck**, and **D. Scaramuzza** (2017). The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. *The International Journal of Robotics Research*, **36**(2), 142–149. xvi, 42, 49, 53, 65, 67, 72, 73

74. **Munda, G.**, **C. Reinbacher**, and **T. Pock** (2018). Real-time intensity-image reconstruction for event cameras using manifold regularisation. *International Journal of Computer Vision*, **126**(12), 1381–1393. xiv, xv, 5, 40, 41, 42, 43, 46, 49, 52, 53, 54

75. **Nagata, J.**, **Y. Sekikawa**, **K. Hara**, and **Y. Aoki** (2019). Foe-based regularization for optical flow estimation from an in-vehicle event camera. *In International Workshop on Advanced Image Technology (IWAIT) 2019*, volume 11049. International Society for Optics and Photonics. 59

76. **Nah, S.**, **T. Hyun Kim**, and **K. Mu Lee** (2017). Deep multi-scale convolutional neural network for dynamic scene deblurring. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. xi, 29, 30, 31, 36, 54

77. **Nair, V.** and **G. E. Hinton** (2010). Rectified linear units improve restricted boltzmann machines. *In ICML*. 92

78. **Nayar, S. K.** and **M. Ben-Ezra** (2004). Motion-based motion deblurring. *IEEE transactions on pattern analysis and machine intelligence*, **26**(6), 689–698. 21

79. **Nguyen, A.**, **T.-T. Do**, **D. G. Caldwell**, and **N. G. Tsagarakis** (2019). Real-time 6dof pose relocalization for event cameras with stacked spatial lstm networks. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 42

80. **Niklaus, S.**, **L. Mai**, and **F. Liu** (2017*a*). Video frame interpolation via adaptive convolution. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 17, 20

81. **Niklaus, S.**, **L. Mai**, and **F. Liu** (2017*b*). Video frame interpolation via adaptive separable convolution. *In Proceedings of the IEEE International Conference on Computer Vision*. 17, 20

82. **Okawara, T.**, **M. Yoshida**, **H. Nagahara**, and **Y. Yagi** (2020). Action recognition from a single coded image. *In 2020 IEEE International Conference on Computational Photography (ICCP)*. 19, 27, 35, 36

83. **Paliwal, A.** and **N. K. Kalantari** (2020). Deep slow motion video reconstruction with hybrid imaging system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 21

84. **Paredes-Vallés, F.**, **K. Y. W. Scheper**, and **G. C. H. E. De Croon** (2019). Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception. *IEEE TPAMI*. 59

85. **Park, J. Y.** and **M. B. Wakin** (2009). A multiscale framework for compressive sensing of video. *In 2009 Picture Coding Symposium*. IEEE. 21

86. **Paszke, A.**, **S. Gross**, **F. Massa**, **A. Lerer**, **J. Bradbury**, **G. Chanan**, **T. Killeen**, **Z. Lin**, **N. Gimelshein**, **L. Antiga**, **A. Desmaison**, **A. Kopf**, **E. Yang**, **Z. DeVito**, **M. Raison**, **A. Tejani**, **S. Chilamkurthy**, **B. Steiner**, **L. Fang**, **J. Bai**, and **S. Chintala** (2019*a*). Pytorch: An imperative style, high-performance deep learning library. *In Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc. 94

87. **Paszke, A.**, **S. Gross**, **F. Massa**, **A. Lerer**, **J. Bradbury**, **G. Chanan**, **T. Killeen**, **Z. Lin**, **N. Gimelshein**, **L. Antiga**, *et al.* (2019*b*). Pytorch: An imperative style, high-performance deep learning library. *In Advances in Neural Information Processing Systems*. 29

88. **Perot, E.**, **P. de Tournemire**, **D. Nitti**, **J. Masci**, and **A. Sironi** (2020). Learning to detect objects with a 1 megapixel event camera. *arXiv preprint arXiv:2009.13436*. xvii, 67, 75, 76

89. **Pinard, C.** (2021). Pytorch correlation module. URL `https://github.com/ClementPinard/Pytorch-Correlation-extension`. xii, 93

90. **Posch, C.**, **T. Serrano-Gotarredona**, **B. Linares-Barranco**, and **T. Delbruck** (2014). Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output. *Proceedings of the IEEE*, **102**(10), 1470–1484, `doi:10.1109/JPROC.2014.2346153`. 76

91. **Purohit, K.**, **A. Shah**, and **A. Rajagopalan** (2019). Bringing alive blurred moments. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 20

92. **Raskar, R.**, **A. Agrawal**, and **J. Tumblin** (2006). Coded exposure photography: motion deblurring using fluttered shutter. *In ACM transactions on graphics (TOG)*, volume 25. ACM. 4, 11, 17, 19, 21, 29, 109

93. **Rebecq, H.**, **T. Horstschäfer**, **G. Gallego**, and **D. Scaramuzza** (2016*a*). Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE Robotics and Automation Letters*, **2**(2), 593–600. 42

94. **Rebecq, H.**, **T. Horstschäfer**, **G. Gallego**, and **D. Scaramuzza** (2016*b*). Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE Robotics and Automation Letters*, **2**(2), 593–600. 43

95. **Rebecq, H.**, **R. Ranftl**, **V. Koltun**, and **D. Scaramuzza** (2019*a*). Events-to-video: Bringing modern computer vision to event cameras. *In Proceedings of the IEEE Conference on CVPR*. xiv, xvi, 6, 41, 43, 56, 57, 60, 68, 70, 71, 72, 78

96. **Rebecq, H.**, **R. Ranftl**, **V. Koltun**, and **D. Scaramuzza** (2019*b*). High speed and high dynamic range video with an event camera. *IEEET-PAMI*. 6, 7, 56, 57, 58

97. **Reddy, D.**, **A. Veeraraghavan**, and **R. Chellappa** (2011). P2c2: Programmable pixel compressive camera for high speed imaging. *In CVPR 2011*. IEEE. 4, 11, 17, 19, 21

98. **Reinbacher, C.**, **G. Graber**, and **T. Pock** (2016*a*). Real-time intensity-image reconstruction for event cameras using manifold regularisation. *arXiv preprint arXiv:1607.06283*. xvi, 56, 60, 70, 71

99. **Reinbacher, C.**, **G. Graber**, and **T. Pock** (2016*b*). Real-Time Intensity-Image Reconstruction for Event Cameras Using Manifold Regularisation. *In 2016 British Machine Vision Conference (BMVC)*. 5

100. **Ren, Z.**, **J. Yan**, **B. Ni**, **B. Liu**, **X. Yang**, and **H. Zha** (2017). Unsupervised deep learning for optical flow estimation. *In Thirty-First AAAI Conference on Artificial Intelligence*. 7, 58, 63, 89

101. **Ronneberger, O.**, **P. Fischer**, and **T. Brox** (2015). U-net: Convolutional networks for biomedical image segmentation. *In International Conference on Medical image computing and computer-assisted intervention*. Springer. 27, 35, 65

102. **Sarhangnejad, N.**, **N. Katic**, **Z. Xia**, **M. Wei**, **N. Gusev**, **G. Dutta**, **R. Gulve**, **H. Haim**, **M. M. Garcia**, **D. Stoppa**, *et al.* (2019). 5.5 dual-tap pipelined-code-memory coded-exposure-pixel cmos image sensor for multi-exposure single-frame computational imaging. *In 2019 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE. 5, 11, 19, 21

103. **Scheerlinck, C.**, **N. Barnes**, and **R. Mahony** (2018*a*). Continuous-time intensity estimation using event cameras. *In Asian Conference on Computer Vision*. Springer. xiv, xv, 40, 41, 43, 49, 52, 53, 54, 55

104. **Scheerlinck, C.**, **N. Barnes**, and **R. Mahony** (2018*b*). Continuous-time intensity estimation using event cameras. *In Asian Conference on Computer Vision*. Springer. xvi, 60, 67, 70, 71, 72, 73, 75

105. **Shechtman, E.**, **Y. Caspi**, and **M. Irani** (2005). Space-time super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(4), 531–545. 21

106. **Shedligeri, P.** and **K. Mitra** (2019). Photorealistic image reconstruction from hybrid intensity and event-based sensor. *Journal of Electronic Imaging*, **28**(6), 063012. 21

107. **Shedligeri, P. A.** and **K. Mitra** (2018). Photorealistic image reconstruction from hybrid intensity and event based sensor. *arXiv preprint arXiv:1805.06140*. 60

108. **Shi, X.**, **Z. Chen**, **H. Wang**, **D. Yeung**, **W. Wong**, and **W. Woo** (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *In Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. xvii, 92, 93

109. **Srinivasan, P. P.**, **R. Ng**, and **R. Ramamoorthi** (2017*a*). Light field blind motion deblurring. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 94

110. **Srinivasan, P. P.**, **T. Wang**, **A. Sreelal**, **R. Ramamoorthi**, and **R. Ng** (2017*b*). Learning to synthesize a 4d rgbd light field from a single image. *In Proceedings of the IEEE International Conference on Computer Vision*. 84

111. **Sun, D.**, **X. Yang**, **M.-Y. Liu**, and **J. Kautz** (2018). Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. xv, 45, 46, 50, 51, 52

112. **Szeliski, R.** (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media. 49

113. **Takahashi, K.**, **Y. Kobayashi**, and **T. Fujii** (2018). From focal stack to tensor light-field display. *IEEE Transactions on Image Processing*, **27**(9), 4571–4584. 86, 100

114. **Tankovich, V.**, **C. Häne**, **S. Fanello**, **Y. Zhang**, **S. Izadi**, and **S. Bouaziz** (2020). Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching. *arXiv preprint arXiv:2007.12140*. 96, 97, 98

115. **Teed, Z.** and **J. Deng** (2020). Raft: Recurrent all-pairs field transforms for optical flow. *In European Conference on Computer Vision*. Springer. 99

116. **Tucker, R.** and **N. Snavely** (2020). Single-view view synthesis with multiplane images. *In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 84

117. **Tulyakov, S.**, **D. Gehrig**, **S. Georgoulis**, **J. Erbach**, **M. Gehrig**, **Y. Li**, and **D. Scaramuzza** (2021). Time lens: Event-based video frame interpolation. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 111

118. **Ummenhofer, B.**, **H. Zhou**, **J. Uhrig**, **N. Mayer**, **E. Ilg**, **A. Dosovitskiy**, and **T. Brox** (2017). Demon: Depth and motion network for learning monocular stereo. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 50, 51, 52

119. **Urvoy, M.**, **M. Barkowsky**, **R. Cousseau**, **Y. Koudota**, **V. Ricorde**, **P. Le Callet**, **J. Gutierrez**, and **N. Garcia** (2012). Nama3ds1-cospad1: Subjective video quality assessment database on coding conditions introducing freely available high quality 3d stereoscopic sequences. *In 2012 Fourth International Workshop on Quality of Multimedia Experience*. IEEE. 104

120. **Vadathya, A. K.**, **S. Cholleti**, **G. Ramajayam**, **V. Kanchana**, and **K. Mitra** (2017). Learning light field reconstruction from a single coded image. *In 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*. IEEE. 84

121. **Vadathya, A. K.**, **S. Girish**, and **K. Mitra** (2019). A unified learning-based framework for light field reconstruction from coded projections. *IEEE Transactions on Computational Imaging*, **6**, 304–316. 81, 84

122. **Veeraraghavan, A.**, **R. Raskar**, **A. Agrawal**, **A. Mohan**, and **J. Tumblin** (2007). Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Trans. Graph.*, **26**(3), 69. 81, 84

123. **Wang, T.-C.**, **J.-Y. Zhu**, **N. K. Kalantari**, **A. A. Efros**, and **R. Ramamoorthi** (2017). Light field video capture using a learning-based hybrid imaging system. *ACM Transactions on Graphics (TOG)*, **36**(4), 1–13. 7, 81, 84, 85, 96, 105

124. **Wang, Y.**, **Z. Lai**, **G. Huang**, **B. H. Wang**, **L. Van Der Maaten**, **M. Campbell**, and **K. Q. Weinberger** (2019*a*). Anytime stereo image depth estimation on mobile devices. *In 2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 96, 97, 98

125. **Wang, Y.**, **F. Liu**, **Z. Wang**, **G. Hou**, **Z. Sun**, and **T. Tan** (2018*a*). End-to-end view synthesis for light field imaging with pseudo 4dcnn. *In Proceedings of the European Conference on Computer Vision (ECCV)*. 84

126. **Wang, Y.**, **P. Wang**, **Z. Yang**, **C. Luo**, **Y. Yang**, and **W. Xu** (2019*b*). Unos: Unified unsupervised optical-flow and stereo-depth estimation by watching videos. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 89

127. **Wang, Y.**, **Y. Yang**, **Z. Yang**, **L. Zhao**, **P. Wang**, and **W. Xu** (2018*b*). Occlusion aware unsupervised learning of optical flow. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 89

128. **Wang, Y.-P.**, **L.-C. Wang**, **D.-H. Kong**, and **B.-C. Yin** (2015). High-resolution light field capture with coded aperture. *IEEE Transactions on Image Processing*, **24**(12), 5609–5618. 81, 84

129. **Wang, Z. W.**, **P. Duan**, **O. Cossairt**, **A. Katsaggelos**, **T. Huang**, and **B. Shi** (2020). 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*. 21

130. **Wang, Z. W.**, **W. Jiang**, **K. He**, **B. Shi**, **A. Katsaggelos**, and **O. Cossairt** (2019*c*). Event-driven video frame synthesis. *In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 21, 43, 50, 58, 60, 62, 68

131. **Wei, M.**, **N. Sarhangnejad**, **Z. Xia**, **N. Gusev**, **N. Katic**, **R. Genov**, and **K. N. Kutulakos** (2018). Coded two-bucket cameras for computer vision. *In Proceedings of the European Conference on Computer Vision (ECCV)*. 21

132. **Wetzstein, G.**, **D. Lanman**, **M. Hirsch**, and **R. Raskar** (2012). Tensor Displays: Compressive Light Field Synthesis using Multilayer Displays with Directional Backlighting. *ACM Trans. Graph. (Proc. SIGGRAPH)*, **31**(4), 1–11. 8, 82, 85, 86, 91, 100

133. **Wilburn, B.**, **N. Joshi**, **V. Vaish**, **E.-V. Talvala**, **E. Antunez**, **A. Barth**, **A. Adams**, **M. Horowitz**, and **M. Levoy** (2005). High performance imaging using large camera arrays. *In ACM SIGGRAPH 2005 Papers*, 765–776. ACM. 21

134. **Wu, G.**, **M. Zhao**, **L. Wang**, **Q. Dai**, **T. Chai**, and **Y. Liu** (2017). Light field reconstruction using deep convolutional network on epi. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 84

135. **Xiao, J.**, **A. Owens**, and **A. Torralba** (2013). Sun3d: A database of big spaces reconstructed using sfm and object labels. *In Proceedings of the IEEE International Conference on Computer Vision*. 51

136. **Xiao, R.**, **W. Sun**, **J. Pang**, **Q. Yan**, and **J. Ren** (2018). Dsr: Direct self-rectification for uncalibrated dual-lens cameras. *3DV*. 105

137. **Yang, G.**, **J. Manela**, **M. Happold**, and **D. Ramanan** (2019). Hierarchical deep stereo matching on high-resolution images. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 96, 97, 98

138. **Yang, J.**, **X. Yuan**, **X. Liao**, **P. Llull**, **D. J. Brady**, **G. Sapiro**, and **L. Carin** (2014). Video compressive sensing using gaussian mixture models. *IEEE Transactions on Image Processing*, **23**(11), 4863–4878. xiv, 21, 28, 30, 31, 32

139. **Yin, Z.** and **J. Shi** (2018). Geonet: Unsupervised learning of dense depth, optical flow and camera pose. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. 88

140. **Yoshida, M.**, **A. Torii**, **M. Okutomi**, **K. Endo**, **Y. Sugiyama**, **R.-i. Taniguchi**, and **H. Nagahara** (2018). Joint optimization for compressive video sensing and reconstruction under hardware constraints. *In Proceedings of the European Conference on Computer Vision (ECCV)*. xi, xiv, 4, 17, 18, 19, 21, 24, 28, 29, 30, 31, 32, 36

141. **Yuan, X.** (2016). Generalized alternating projection based total variation minimization for compressive sensing. *In 2016 IEEE International Conference on Image Processing (ICIP)*. IEEE. 21

142. **Zhang, K.**, **G. Riegler**, **N. Snavely**, and **V. Koltun** (2020). Nerf++: Analyzing and improving neural radiance fields. 84

143. **Zhang, R.**, **P. Isola**, **A. A. Efros**, **E. Shechtman**, and **O. Wang** (2018*a*). The unreasonable effectiveness of deep features as a perceptual metric. *In CVPR*. xii, xvii, 78, 79

144. **Zhang, R.**, **P. Isola**, **A. A. Efros**, **E. Shechtman**, and **O. Wang** (2018*b*). The unreasonable effectiveness of deep features as a perceptual metric. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. xviii, 98, 107

145. **Zhang, Z.**, **Y. Liu**, and **Q. Dai** (2015). Light field from micro-baseline image pair. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 84, 96, 98

146. **Zhou, T.**, **M. Brown**, **N. Snavely**, and **D. G. Lowe** (2017). Unsupervised learning of depth and ego-motion from video. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. 64, 88

147. **Zhou, T.**, **R. Tucker**, **J. Flynn**, **G. Fyffe**, and **N. Snavely** (2018*a*). Stereo magnification: Learning view synthesis using multiplane images. *In SIGGRAPH*. 84

148. **Zhou, Y.**, **G. Gallego**, **H. Rebecq**, **L. Kneip**, **H. Li**, and **D. Scaramuzza** (2018*b*). Semi-dense 3d reconstruction with a stereo event camera. *In Proceedings of the European Conference on Computer Vision (ECCV)*. 42

149. **Zhu, A.**, **L. Yuan**, **K. Chaney**, and **K. Daniilidis** (2018*a*). Ev-flownet: Self-supervised optical flow estimation for event-based cameras. *In Proceedings of Robotics: Science and Systems*. Pittsburgh, Pennsylvania, `doi:10.15607/RSS.2018.XIV.062`. 59, 60, 67, 71, 72

150. **Zhu, A. Z.**, **D. Thakur**, **T. Özaslan**, **B. Pfrommer**, **V. Kumar**, and **K. Daniilidis** (2018*b*). The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. *IEEE Robotics and Automation Letters*, **3**(3), 2032–2039. xi, xvi, 67, 70, 71, 73

151. **Zhu, A. Z.**, **L. Yuan**, **K. Chaney**, and **K. Daniilidis** (2018*c*). Unsupervised event-based optical flow using motion compensation. *In European Conference on Computer Vision Workshop*. Springer. 59

152. **Zhu, A. Z.**, **L. Yuan**, **K. Chaney**, and **K. Daniilidis** (2019). Unsupervised event-based learning of optical flow, depth, and egomotion. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 42, 59, 60, 71, 72

153. **Zihao Zhu, A.**, **Y. Chen**, and **K. Daniilidis** (2018). Realtime time synchronized event-based stereo. *In Proceedings of the European Conference on Computer Vision (ECCV)*. 42