HIGH-SPEED VIDEO RECONSTRUCTION FROM CODED-EXPOSURE IMAGES

A Project Report

submitted by

ANUPAMA S

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY
&
MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS

JULY 2020

THESIS CERTIFICATE

This is to certify that the thesis titled **HIGH-SPEED VIDEO RECONSTRUCTION FROM CODED-EXPOSURE IMAGES**, submitted by **ANUPAMA S**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology & Master of Technology**, is a bonafide record of the research work done by him under our supervision. 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.

Prof. Kaushik Mitra
Research Guide
Professor
Dept. of Electrical Engineering
IIT Madras, 600 036

Place: Chennai

Date: July 2020

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my project advisor, Prof. Kaushik Mitra, for his immense guidance and support throughout the course of my project, and also for providing the lab resources to run the experiments needed for this project. I would also like to thank Prasan Shedligeri, PhD research scholar at Computational Imaging Lab, working under Prof. Mitra, for his extensive guidance and help in ideating and analyzing concepts for this project. Finally, I want to thank my family and friends for their constant encouragement and support throughout my education and research.

ABSTRACT

Recently, learning based methods have proven to be effective in recovering a full video sequence from a single blurred image. Although these techniques do not require any hardware modifications, the inversion problem involved is inherently ill-posed and suffers from motion ambiguity leading to poor video reconstruction. Coded exposure techniques on the other hand require significant hardware modification but provide a better posed recovery system. Recently, a novel prototype image sensor based on multiple buckets per pixel was proposed. For the first time, these sensors have enabled the ability to acquire multiple coded images in a single exposure of the sensor. As with any compressed sensing system, multiple measurements from the same underlying signal make it better-posed to recover the original signal. This project proposes a system to recover a video sequence from a single exposure of this multi-bucket sensor. The objective is to show that a better video can be recovered when we have two coded images as input rather than one. A two-stage learning based model is proposed to recover the original video from the compressed measurements. The first stage consists of an inversion layer that extracts features of the video to be recovered. This inversion is modeled using a single layer of convolutional neural network where the weights are allowed to be spatially adaptive. In the second stage, the extracted features of the video are refined and used to recover the complete video using a deep neural network. Through this project, it was observed that with two coded exposure measurements, the recovered video quality is much better than having a single coded exposure image. The proposed model is fully-convolutional, therefore the video is reconstructed at once from the entire image which avoids the artifacts from the patch-based reconstruction methods of dictionary learning and some recent neural network based methods.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Deblurring a blurred image has been a long-standing problem in the field of image processing and computer vision. This problem is inherently ill-posed and requires strong prior knowledge to be imposed on the restoration problem. Hence, learning based methods which learn to impose a data-driven prior, have made a significant progress in obtaining better deblurred images. As the blurred images encode the scene motion information, there has been a recent interest in recovering a video sequence from a single blurred image (Purohit *et al.*, 2019; Jin *et al.*, 2018). Although this is a very challenging and highly ill-posed problem, there has been some advancement with the help of strong priors in the form of trained neural networks. For these video recovery techniques, motion ambiguity has been a major challenge which can be addressed by using coded exposure compressive video sensing. Coded exposure techniques can be broadly divided into two categories: a) global, sensor-level coding (Holloway *et al.*, 2012; Raskar *et al.*, 2006; Llull *et al.*, 2013) and b) pixel-wise coding (Reddy *et al.*, 2011; Liu *et al.*, 2013; Iliadis *et al.*, 2018, 2020; Yoshida *et al.*, 2018; Martel *et al.*, 2020; Li *et al.*, 2020).

In these methods, a single exposure to acquire the image is divided into multiple sub-exposures. A pre-determined code on the local pixel level or the frame level is then used to encode these sub-exposure frames into a single coded frame. It has been shown that exposure codes which have a broad frequency spectrum are generally a good choice for compressive sensing. Several prototype sensors have been proposed over the years to implement this compressive measurement technique as there are no commercially available sensors (Liu *et al.*, 2013; Reddy *et al.*, 2011; Yoshida *et al.*, 2018). These coded exposure techniques typically throw away about 50% of the incoming light (Baraniuk *et al.*, 2017), leading to significant light loss. A second co-located image sensor can be used to capture the full exposure without coding to overcome such a light loss. A recently proposed prototype image sensor based on multi-bucket pixels can be used for this task. The prototype sensor called *Coded 2 Bucket* (C2B) sensor (Wei *et al.*, 2018) has 2 light-collecting buckets per pixel and allows pixel-wise control of the code.

C2B outputs two images for each exposure where the first image is encoded based on the predetermined code while the second image is encoded with the complement of the predetermined code. We can obtain a fully-exposed image by simply averaging the two output image frames. Hence, C2B sensors are 100% light efficient while providing complete freedom to control the exposure pattern on an individual pixel level.

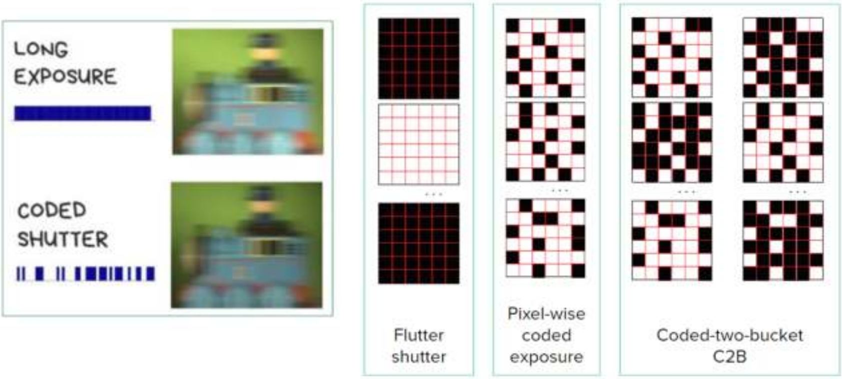
In the past, several algorithms have been proposed to recover the underlying video signal from compressed measurements of different imaging systems. Each of these algorithms use an unique code pattern and also differ in the compression rate (number of sub-exposures per full exposure) achieved. Here, the aim is to bring all the different imaging systems under a single umbrella so that a fair comparison is possible between them. This comparison can give a definitive answer on the video recovery performance of different imaging systems and let us have a fair discussion on the different hardware trade-offs involved in the imaging architecture. To achieve this, a two-stage learning based algorithm is proposed, consisting of an inversion stage and a refinement stage. The inversion stage consists of a *shift-variant* convolutional layer which is inspired from the linear algebra principles of solving an under-determined system of equations. The refinement stage consists of a deep neural network that outputs the recovered video signal. The proposed network provides enough flexibility for it to be easily adapted to various compressed imaging architectures.

CHAPTER 2

RELATED WORK

Coded exposure imaging: In Raskar *et al.* (2006), flutter shutter camera was proposed for making motion deblurring a well-posed problem. A similar system was proposed in Veeraraghavan *et al.* (2010), which used a coded strobing photography system for compressive sensing of high-speed videos. The flutter shutter camera further extended to recover a video sequence in Holloway *et al.* (2012). In Gu *et al.* (2010), the authors created a high speed camera by cleverly sampling the rows in a rolling shutter camera. In Reddy *et al.* (2011), the authors proposed a pixel-wise coded exposure system for compressive sensing of high-speed videos. In Liu *et al.* (2013), the authors used a similar coded exposure architecture but constrained the system to use the existing CMOS image sensor architecture. Recently, Antipa *et al.* (2019) proposed a compressive video acquisition system where the lens was replaced by a diffractive optical element. In Yoshida *et al.* (2018), the authors use the system proposed in Liu *et al.* (2013) and jointly learn the coded exposure mask as well as the recovery of video using a neural network. In Gupta *et al.* (2010) a coded exposure system is proposed which gives the post-capture control of changing the spatial and temporal resolutions.

Figure 2.1: Exposure patterns for different compressive sensing systems.



High-speed imaging systems: In Wilburn *et al.* (2004); Shechtman *et al.* (2002), authors use a multi-camera system for capturing a high-speed video. A hybrid intensity and event sensor based system was proposed in Pan *et al.* (2019) for extracting a video sequence using a blurry image and information from an event sensor. Event based sensors have also been used to design a low power high-speed camera (Scheerlinck *et al.*, 2018; Reinbacher *et al.*, 2016; Rebecq *et al.*, 2019; Shedligeri and Mitra, 2018). Other methods of high-speed imaging involve temporal super-resolution of video sequences captured from a low frame-rate camera (Karim *et al.*, 2003). Some methods propose interpolation of multiple frames between successive frames of a low-frame rate video using optical flow (Kaviani and Shirani, 2015), auto-regressive model (Zhang *et al.*, 2009), kernel regression (Takeda *et al.*, 2009), learning-based methods (Jiang *et al.*, 2018) among others. Recently, few works have also explored the possibility of decomposing a single blurred frame into a sequence of video frames (Purohit *et al.*, 2019; Jin *et al.*, 2018).

CHAPTER 3

VIDEO RECONSTRUCTION FROM COMPRESSED MEASUREMENTS

This section explains the proposed method to obtain video from the compressed measurements of the underlying video signal. As flutter shutter video camera is the most basic of the compressed video sensing architectures, the algorithm is first explained for this camera. Later, a brief explanation is provided on how the proposed method can be adapted to other compressive sensing architectures as well. First, a mathematical representation of the video compressive sensing architecture is provided, followed by the explanation of the proposed algorithm to recover the underlying video signal from the compressed measurements.

3.1 Compressive video sensing

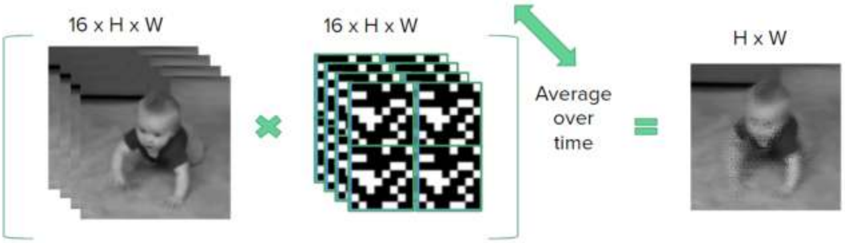


Figure 3.1: Simulation of coded exposure images.

In compressed video sensing a single exposure of the image sensor is sub-divided into T sub-exposures. These sub-exposures are used to multiplex the T video frames into a single frame using a coded exposure pattern. To simulate this process, consider a video sequence of T frames which can be denoted by $X = [x_1, x_2, \ldots, x_T]$, where $x_t \in [0, 1]^{M\times N}$ and a coded exposure pattern $\phi = [\phi_1, \phi_2, \ldots, \phi_T]$ where $\phi_t \in \{0, 1\}^{M\times N}$. Then the coded-exposure image I can then be written as,

$$I = \frac{1}{T}\Sigma_{t=1}^{T} c_t \odot x_t \tag{3.1}$$

where $\odot$ denotes element-wise multiplication. For the case of C2B sensor, we have two images output from the buckets $B0$ and $B1$ of the sensor, denoted by I^0 and I^1 respectively. I^0 and I^1 can be written as,

$$I^0 = \frac{1}{T}\Sigma_{t=1}^{T} c_t \odot x_t \tag{3.2}$$

$$I^1 = \frac{1}{T}\Sigma_{t=1}^{T} (1 - c_t) \odot x_t \tag{3.3}$$

where $\odot$ denotes element-wise multiplication. The corresponding fully-exposed image can be obtained by $I^b = I^0 + I^1$.

This coded exposure pattern can be different for each pixel in the image. However, images are correlated only in local neighborhood regions, hence the exposure pattern is made periodic with a period of P pixels. So, for ease of explanation from now on, images I^0, I^1 and I^b denote only the $P \times P$ pixel patch of the whole image. Correspondingly the exposure pattern ϕ and the original video X are also considered for a small spatial patch $P \times P$ and the full temporal extent of T frames. The individual pixels of the image frames are addressed as I^0_p, I^1_p and I^b_p and the corresponding exposure pattern as ϕ_p, where $p \in \{1, 2, \ldots, P^2\}$. The following sections elaborate on the details of the inversion stage and the refinement stage of the video reconstruction algorithm.

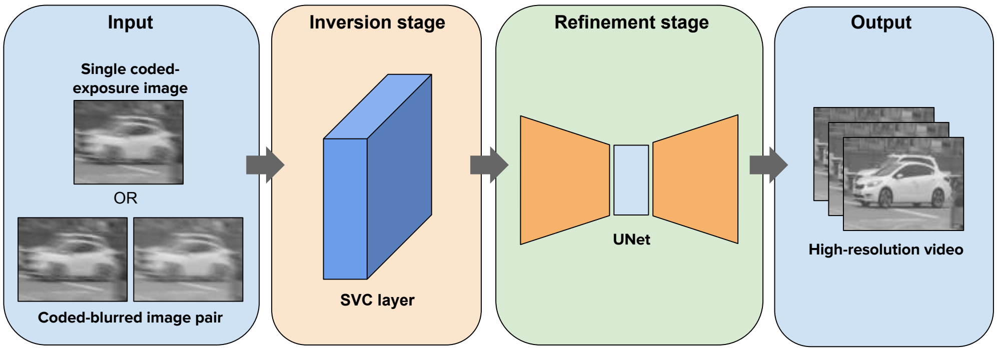


Figure 3.2: Diagrammatic representation of the proposed architecture.

3.2 Inversion stage

For an easier analysis in this section, let us restrict ourselves to the flutter-shutter imaging system where every pixel shares the same global exposure pattern and we obtain only a single coded exposure image from each exposure. Lets denote this restricted exposure pattern as ϕ_r which is a row-vector of length T as it encodes T temporal frames into a single coded frame I_r. This can be written as,

$$I_r = \phi_r X, \tag{3.4}$$

where $X \in [0,1]^{T \times P^2}$, whose columns represent the video sequence at each spatial pixel. This is an under-determined and ill-posed system and the compressed signal can be approximately recovered by,

$$\tilde{X} = \phi_r^{\dagger} I_r \tag{3.5}$$

where $\phi_r^{\dagger}$ represents the Moore-Penrose pseudo-inverse of the matrix ϕ_r. The matrix $\phi_r^{\dagger}$ satisfies $\phi_r \phi_r^{\dagger} X \approx X$ and is given by

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

As we have restricted to a special case of flutter shutter coded exposure, the pseudo-inverse matrix $\phi_r^{\dagger}$ is a column vector. From Eq. (3.5), we can observe that each column of the approximately recovered video $\tilde{X}$ is just a scaled version of the pseudo-inverse matrix $\phi_r^{\dagger}$. The weights for scaling are determined by the column entries of the compressed measurements I_r. The important thing to be observed here is that, in order to recover the video sequence at a particular pixel, we only need the compressed measurements at that pixel alone.

Learning based algorithms have been proposed to solve the inverse problem of recovering video signal from a coded exposure measurement. Most of these algorithms first divide the input image into overlapping patches, then input these patches into a fully-connected network and the output video patches are averaged and stitched to recover the video. The recovered videos thus have patch artifacts and the number of computations required to obtain the video increases quadratically with the overlap ratio. It's also well known that a fully-connected network increases the number of parameters in the network. A fully-convolutional network on the other hand can reconstruct the entire video sequence in a single forward pass and also use much less number of parameters. Unlike fully-connected networks, convolutional networks have local connectivity at each layer. As mentioned earlier, to recover a video at a particular pixel, only that pixel's compressed measurements are necessary. This proves that global connectivity is not necessary for effectively recovering the video from the input coded images. Hence, the local connectivity offered by convolutional networks can be efficiently used to model the task of recovering the video signal.

Eq. (3.5) provides an approximate solution to recover the original video signal. This solution does not utilize any knowledge about the data itself, while the proposed method is a supervised learning method that incorporates the training data distribution to learn the inverse mapping. The simplest approach to learning the inverse is to set up a linear regression problem, where we want to learn a linear mapping from the input compressed measurements to the ground truth video signal. This can be mathematically written as,

$$\min_{w} \|X - I_0 w\|_2, \tag{3.7}$$

where w represents the linear regression coefficients. This is a standard least squares problem and the solution for this is given by,

$$w = I_0^{\dagger} X, \tag{3.8}$$

$$w = (\phi_0 X)^{\dagger} X. \tag{3.9}$$

We can notice that the learned weights w are a function of the underlying coded exposure sequence ϕ_0.

3.2.1 Shift-variant convolution

As described in Sec. 3.1, the coded exposure sequence is periodic with period P pixels and varies for each pixel in the local image region of $P \times P$ pixels. The model should give the network freedom to learn different weights to invert the linear system when the underlying exposure sequence is different. Although a standard convolutional layer effectively models the necessary local connectivity, it does not allow the weights to vary between consecutive pixels when the underlying coded exposure sequence varies. Hence, the standard convolutional layer is modified such that the weights vary for each pixel in the local image region of $P \times P$ pixels. Such a *shift variant convolutional layer* has been proposed in Okawara *et al.* (2020) and this layer is used as the first stage in the proposed algorithm.

This layer takes either a single coded image or a pair of coded-blurred images as input (depending on the compressive sensing system used) and extracts a feature map of 64 channels $\tilde{X}$, which is then passed on to the refinement stage.

3.3 Refinement stage

To exploit the full strength of the neural network training, a refinement stage is proposed, consisting of a UNet (Ronneberger *et al.*, 2015) like deep neural network. The refinement stage takes in the feature map $\tilde{X}$ from the inversion stage and outputs a refined video sequence $\hat{X}$. The output of this network is supervised using the ground truth video frames, with a loss function as follows,

$$\mathcal{L}_{ref} = \|\hat{X} - X\|_1. \tag{3.10}$$

In addition to this loss, a TV-smoothness loss is added on the final predicted video frame defined as,

$$\mathcal{L}_{tv} = \|\nabla X\|_1, \tag{3.11}$$

where ∇ is the gradient operator in the x-y directions. Therfore, the overall loss function then becomes,

$$\mathcal{L} = \mathcal{L}_{ref} + \lambda_{tv}\mathcal{L}_{tv}, \tag{3.12}$$

where λ_{tv} is a hyperparameter that weighs the loss function.

CHAPTER 4

EXPERIMENTAL RESULTS

Supplementary presentation containing experimental results and videos can be found in this link.

The proposed network was trained 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 under each sequence were taken and spatially downsampled by a factor of 2. Further, overlapping patch volumes of size $16 \times 64 \times 64$ (temporal$\times$spatial$\times$spatial) were extracted with an overlap of 8 pixels in the temporal dimension and 32 pixels in the spatial dimensions, resulting in a total of $263, 340$ patch volumes of size $16 \times 64 \times 64$ (temporal$\times$spatial$\times$spatial) to form the training dataset. The input to the network is obtained from each patch volume using a $16 \times 8 \times 8$ exposure mask repeated to fill the spatial dimensions, as described in (3.1) and (3.2). The network was trained using Adam optimizer with a learning rate of 0.0001 and batch size of 50 for 500 epochs, with λ_{tv} as 0.1.

4.1 Video reconstruction for different compressive sensing systems

This section evaluates the performance of various existing state-of-the-art video extraction algorithms for compressive sensing along with the proposed method. Two different sets of test videos were used for this analysis, Set-1 is the test set used for evaluation in Yoshida *et al.* (2018) consisting of 14 test videos (16 frames each) and Set-2 consists of 15 test videos (16 frames each) randomly selected from GoPro test data (Nah *et al.*, 2017).

For pixel-wise coded exposure sensing, three different algorithms are evaluated - the proposed method, deep learning based method DNN (Yoshida *et al.*, 2018) and analytical method GMM (Yang *et al.*, 2014), all using the exposure pattern *optimized*

SBE mask proposed in Yoshida *et al.* (2018). Yoshida *et al.* (2018) proposed video reconstruction using a fully-connected deep neural network by jointly optimizing the exposure mask and the reconstruction network. The proposed network on the other hand is fully-convolutional and was trained using the optimized SBE mask from Yoshida *et al.* (2018). For Coded-two-bucket system, there is no existing deep learning based algorithm that incorporates the information from the second bucket to recover the video. Therefore, the evaluation is done only on the proposed method, using the same exposure pattern *optimized SBE mask*.

Table 4.1: Quantitative results for different reconstruction algorithms. The table lists average PSNR(dB) and SSIM of reconstructed videos.

Exposure	**Test data**	**Algorithm**		
		GMM (32)	DNN (33)	Proposed
Pixel-wise coded	Set-1 (33)	29.31, 0.898	30.21, 0.905	**31.14, 0.925**
	Set-2 (16)	29.94, 0.887	30.27, 0.890	**31.76, 0.914**
			Proposed	
Coded-two-bucket (C2B)	Set-1 (33)		**32.23, 0.935**	
	Set-2 (16)		**32.34, 0.920**	

Table 4.1 provides a quantitative analysis of the various reconstruction algorithms while Figures 4.1 and 4.2 provide a visual comparison between the reconstructed videos for above mentioned algorithms. The reconstruction results shown in the figures are for test videos from Set-1. Through this analysis, it can be concluded that the proposed fully-convolutional model for video reconstruction from a coded exposure image is able to perform better than the existing state-of-the art algorithms for video reconstruction from the same input, considering the same exposure pattern. Also, by including complementary information in the blurred image along with the coded image as the input, it is observed that the proposed algorithm is able to produce further improved reconstruction results.

Yoshida *et al.* (2018) proposes a joint optimization method for optimizing the exposure pattern jointly with the reconstruction weights. Although the proposed network architecture does not cover optimizing the exposure pattern, since the network is designed to be end-to-end trainable, it can be easily extended for this task. However, this is not covered in this project, but can be explored as a part of future work.

4.2 Ablation study on proposed architecture

Table 4.2: Ablation study on proposed architecture. The table lists PSNR(dB) and SSIM of reconstructed videos for different architecture changes.

Exposure		**Set-1** (33)		**Set-2** (16)	
		PSNR	SSIM	PSNR	SSIM
Pixel-wise	U-Net	30.86	0.919	31.43	0.907
coded	SVC(16)+U-Net	30.89	0.921	31.56	0.910
	SVC(64)+U-Net	**31.14**	**0.925**	**31.76**	**0.914**
Coded-two-	coded & complement	32.19	0.935	32.31	0.919
bucket (C2B)	coded & blurred	**32.23**	**0.935**	**32.34**	**0.920**

This section investigates some of the architectural choices that were made in developing the proposed network. For the case of pixel-wise coded exposure system, Table 4.2 shows the performance of the proposed algorithm under different architecture changes namely – U-Net, SVC(16) + U-Net and SVC(64) + U-Net. SVC refers to shift-variant convolution and the following value specifies the number of channels in the feature-map extracted by the SVC layer. The first architecture U-Net consists of the refinement stage alone, without the inversion stage. As seen in the table 4.2, since U-Net is a deep network, it is able to perform well, but with a room for improvement. In the second architecture SVC(16) + U-Net, the shift-variant convolution layer extracts features of 16 channels from the input, which can be considered as an intermediate reconstruction $\tilde{X}$ obtained from the inversion stage. There is an additional inversion loss $\mathcal{L}_{inv}$ imposed on the intermediate reconstruction, computed similar to $\mathcal{L}_{ref}$ described in (3.10) as follows.

$$\mathcal{L}_{inv} = \|\tilde{X} - X\|_1 \tag{4.1}$$

Therefore for the architecture SVC(16) + U-Net, the overall loss during training becomes

$$\mathcal{L} = \mathcal{L}_{ref} + \lambda_{inv}\mathcal{L}_{inv} + \lambda_{tv}\mathcal{L}_{tv} \tag{4.2}$$

where λ_{inv} and λ_{tv} are hyperparameters that weigh the loss function. For this analysis, λ_{inv} was set to 0.5 and λ_{tv} to 0.1. The third architecture SVC(64) + U-Net is the proposed architecture, where the shift-variant convolution layer extracts a feature-map of 64 channels as described in section 3.2.1.

For the case of C2B exposure system, the best architecture from the above analysis SVC(64) + U-Net was used to investigate video reconstruction from a pair of coded exposure image I^0 and complementary coded image I^1, and reconstruction from a pair of coded exposure image I^0 and fully exposed image I^b. I^0, I^1 and I^b are measurements obtained as described in section 3.1. As seen in the table 4.2, it was observed that reconstruction from a pair of coded and blurred images produces the best results.

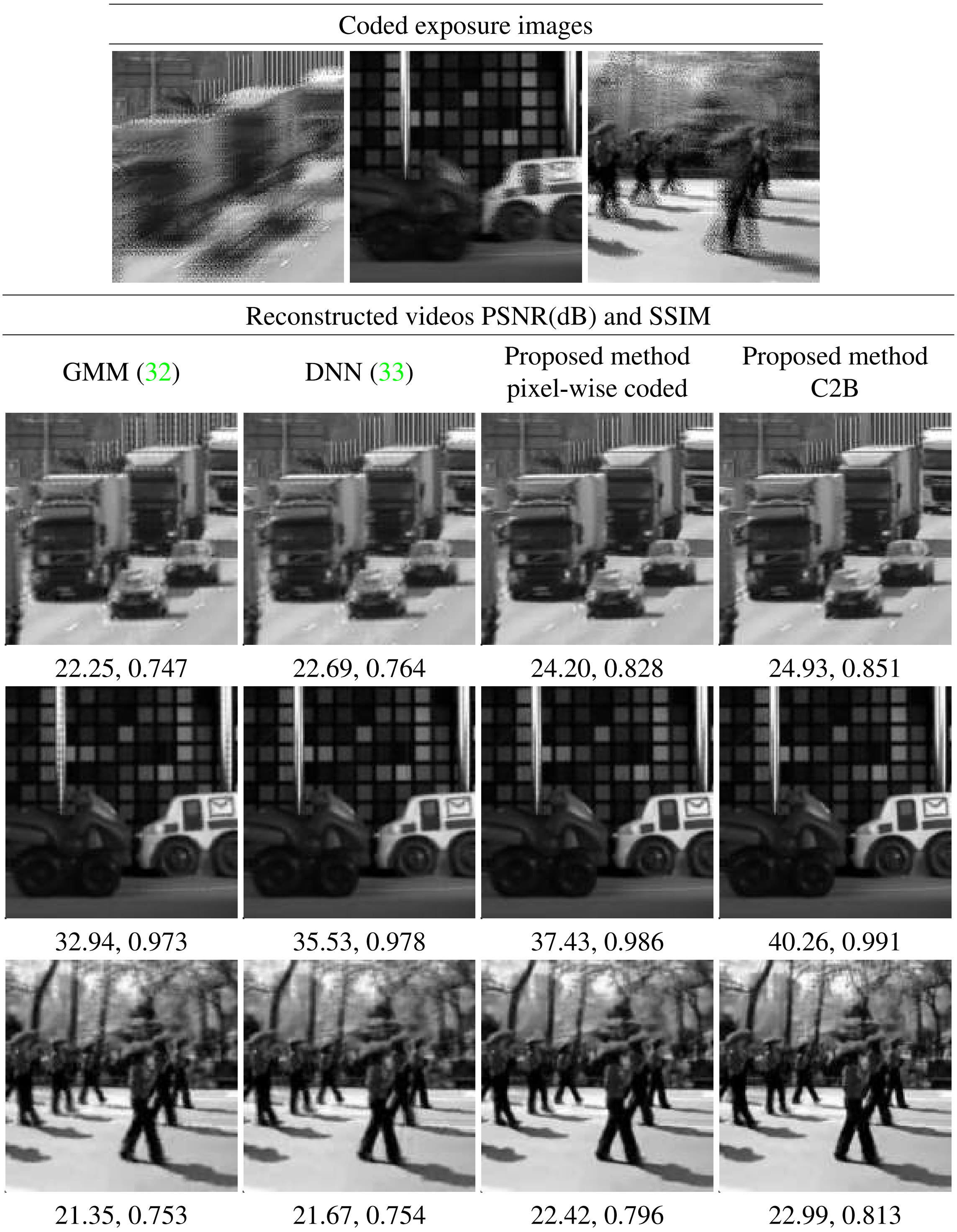


Figure 4.1: Visual comparison of video reconstruction results for different algorithms. The figure shows reconstruction results for test videos from Set-1. Videos can be viewed here.

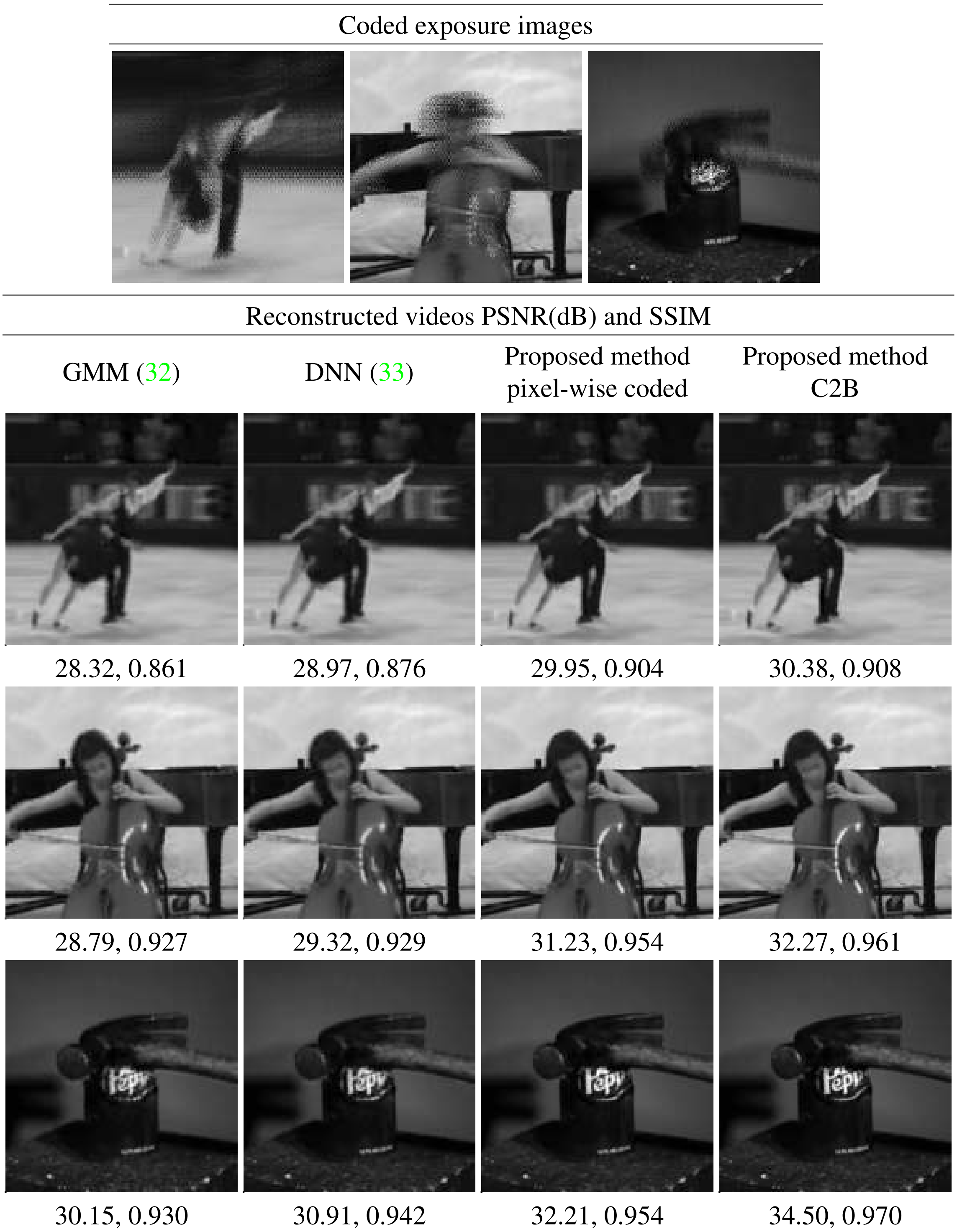


Figure 4.2: Visual comparison of video reconstruction results for different algorithms. The figure shows reconstruction results for test videos from Set-1. Videos can be viewed here.

CHAPTER 5

CONCLUSION

In this project, a unified and flexible deep network architecture was proposed, that can easily be adapted to various compressive imaging systems like flutter shutter, pixel-wise coded exposure and coded-two-bucket systems. The proposed model is capable of producing high-quality video reconstruction results, better than various existing state-of-the-art reconstruction algorithms, as demonstrated through the figures and tables in section 4. This project also analysed and evaluated various existing compressive sensing systems and video reconstruction algorithms based on deep learning as well as analytical optimization methods. Through the study, it was proved that a fully-convolutional deep network model is able to effectively reconstruct videos from the measurements provided by various compressive sensing systems. While extracting a video from a coded exposure image is a better-posed inversion problem compared to extracting video from a fully exposed image, it is found that, by adding complementary information found in the fully exposed image to the input, we are able to achieve further improved reconstruction results. Coded-two-bucket exposure provides the essential complementary information required for superior video reconstruction, which is otherwise lost in pixel-wise coded exposure sensors. The fact that the input pair of coded and blurred images can be obtained using an existing compressive sensing architecture, makes this proposed algorithm possible to be implemented for practical applications.

REFERENCES

1. **Antipa, N.**, **P. Oare**, **E. Bostan**, **R. Ng**, and **L. Waller**, Video from stills: Lensless imaging with rolling shutter. *In 2019 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2019.

2. **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.

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

4. **Gupta, M.**, **A. Agrawal**, **A. Veeraraghavan**, and **S. G. Narasimhan**, Flexible voxels for motion-aware videography. *In European Conference on Computer Vision*. Springer, 2010.

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

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

7. **Iliadis, M.**, **L. Spinoulas**, and **A. K. Katsaggelos** (2020). Deepbinarymask: Learning a binary mask for video compressive sensing. *Digital Signal Processing*, **96**, 102591.

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

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

10. **Karim, H. A.**, **M. Bister**, and **M. U. Siddiqi**, Low rate video frame interpolation-challenges and solution. *In 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 3. IEEE, 2003.

11. **Kaviani, H. R.** and **S. Shirani** (2015). Frame rate upconversion using optical flow and patch-based reconstruction. *IEEE Transactions on Circuits and Systems for Video Technology*, **26**(9), 1581–1594.

12. **Li, Y.**, **M. Qi**, **R. Gulve**, **M. Wei**, **R. Genov**, **K. N. Kutulakos**, and **W. Heidrich**, End-to-end video compressive sensing using anderson-accelerated unrolled networks. *In 2020 IEEE International Conference on Computational Photography (ICCP)*. 2020.

13. **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.

14. **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.

15. **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.

16. **Nah, S.**, **T. H. Kim**, and **K. M. Lee**, Deep multi-scale convolutional neural network for dynamic scene deblurring. *In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

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

18. **Pan, L.**, **C. Scheerlinck**, **X. Yu**, **R. Hartley**, **M. Liu**, and **Y. Dai**, Bringing a blurry frame alive at high frame-rate with an event camera. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

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

20. **Raskar, R.**, **A. Agrawal**, and **J. Tumblin**, Coded exposure photography: motion deblurring using fluttered shutter. *In ACM transactions on graphics (TOG)*, volume 25. ACM, 2006.

21. **Rebecq, H.**, **R. Ranftl**, **V. Koltun**, and **D. Scaramuzza**, Events-to-video: Bringing modern computer vision to event cameras. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

22. **Reddy, D.**, **A. Veeraraghavan**, and **R. Chellappa**, P2c2: Programmable pixel compressive camera for high speed imaging. *In CVPR 2011*. IEEE, 2011.

23. **Reinbacher, C.**, **G. Graber**, and **T. Pock** (2016). Real-time intensity-image reconstruction for event cameras using manifold regularisation. *arXiv preprint arXiv:1607.06283*.

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

25. **Scheerlinck, C.**, **N. Barnes**, and **R. Mahony**, Continuous-time intensity estimation using event cameras. *In Asian Conference on Computer Vision*. Springer, 2018.

26. **Shechtman, E.**, **Y. Caspi**, and **M. Irani**, Increasing space-time resolution in video. *In European Conference on Computer Vision*. Springer, 2002.

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

28. **Takeda, H.**, **P. Milanfar**, **M. Protter**, and **M. Elad** (2009). Super-resolution without explicit subpixel motion estimation. *IEEE Transactions on Image Processing*, **18**(9), 1958–1975.

29. **Veeraraghavan, A.**, **D. Reddy**, and **R. Raskar** (2010). Coded strobing photography: Compressive sensing of high speed periodic videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**(4), 671–686.

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

31. **Wilburn, B.**, **N. Joshi**, **V. Vaish**, **M. Levoy**, and **M. Horowitz**, High-speed videography using a dense camera array. *In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2. IEEE, 2004.

32. **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.

33. **Yoshida, M.**, **A. Torii**, **M. Okutomi**, **K. Endo**, **Y. Sugiyama**, **R.-i. Taniguchi**, and **H. Nagahara**, Joint optimization for compressive video sensing and reconstruction under hardware constraints. *In Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

34. **Zhang, Y.**, **D. Zhao**, **X. Ji**, **R. Wang**, and **W. Gao** (2009). A spatio-temporal auto regressive model for frame rate upconversion. *IEEE Transactions on Circuits and Systems for Video Technology*, **19**(9), 1289–1301.