Computer Vision from Spatial-Multiplexing Cameras at Low Measurement Rates

by

Kuldeep Sharad Kulkarni

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved July 2017 by the
Graduate Supervisory Committee:

Pavan Turaga, Chair
Baoxin Li
Chaitali Chakrabarti
Aswin Sankaranarayanan
Robert LiKamWa

ARIZONA STATE UNIVERSITY

August 2017

ABSTRACT

In UAVs and parking lots, it is typical to first collect an enormous number of pixels using conventional imagers. This is followed by employment of expensive methods to compress by throwing away redundant data. Subsequently, the compressed data is transmitted to a ground station. The past decade has seen the emergence of novel imagers called spatial-multiplexing cameras, which offer compression at the sensing level itself by providing an arbitrary linear measurements of the scene instead of pixel-based sampling. In this dissertation, I discuss various approaches for effective information extraction from spatial-multiplexing measurements and present the trade-offs between reliability of the performance and computational/storage load of the system. In the first part, I present a reconstruction-free approach to high-level inference in computer vision, wherein I consider the specific case of activity analysis, and show that using correlation filters, one can perform effective action recognition and localization directly from a class of spatial-multiplexing cameras, called compressive cameras, even at very low measurement rates of 1%. In the second part, I outline a deep learning based non-iterative and real-time algorithm to reconstruct images from compressively sensed (CS) measurements, which can outperform the traditional iterative CS reconstruction algorithms in terms of reconstruction quality and time complexity, especially at low measurement rates. To overcome the limitations of compressive cameras, which are operated with random measurements and not particularly tuned to any task, in the third part of the dissertation, I propose a method to design spatial-multiplexing measurements, which are tuned to facilitate the easy extraction of features that are useful in computer vision tasks like object tracking. The work presented in the dissertation provides sufficient evidence to high-level inference in computer vision at extremely low measurement rates, and hence allows us to think about the possibility of revamping the current day computer systems.

ACKNOWLEDGMENTS

Dedicated to my parents, my grandparents, and my brother.

I would like to express my deepest gratitude to my advisor Dr. Pavan Turaga, whose constant guidance and encouragement have steered this dissertation to see the light of day. During my early grad days, as I swam in the world of uncertainty suffering from many a rejection at various conferences, talking to him helped calm my nerves and gave me the impetus and inspiration to fight back. He has not only been a great advisor, but also has been a great friend from whom I have learnt many a life lesson, which I believe will be useful throughout the rest of my life. I would like to thank Prof. Baoxin Li, Prof. Chaitali Chakrabarti, Dr. Aswin Sankaranarayanan, and Dr. Robert LiKamWa for graciously agreeing to serve in my committee and providing useful feedback on my dissertation.

I would like to thank the administrative staff of the School of Electrical, Computer and Energy Engineering (ECEE) and the School of Arts, Media and Engineering (AME), in particular Darleen Mandt and Toni Mengert for helping me navigate the administrative maze with great ease. I am also grateful to Charlie of HPC for educating me on how to utilize the saguaro cluster in the most efficient manner.

I would like to thank Dr. Aswin Sankaranarayan, Prof. Amit Ashok, and Prof. Al Molnar for graciously hosting me at their labs for our collaborative projects. My three internship stints at Bausch and Lomb (B&L), Rochester, Xerox Research Centre India (XRCI), Bangalore, and Mitsubishi Electric Research Labs (MERL), Cambridge provided me with wonderful opportunities to take a peek at how industry research works, and understand how different it is from the academic research. I would like thank my mentors Dr. Nishant Mohan (B&L), Dr. Arijit Biswas (XRCI), and Dr. Mike Jones (MERL) for being excellent mentors and introducing me to problems which I had not got a chance to work on at ASU.

I would like to thank my collaborators, Suhas L, who being my co-author on many papers has made several contributions to my dissertation, Ronan for providing us their single-pixel camera to obtain real data for the ReconNet work, Sohil with whom I have had many fruitful discussions during our internship at XRCI, Suren for explaining me the workings of light-fields and ASPs, Mayank and Arjun for performing the deep learning experiments for light-field reconstructions which helped us generalize the ReconNet framework for 4D data, and Bharat for explaining all the intricacies of the action detection pipeline.

Being one of the first students of the group, with no seniors, there was an air of apprehension that accompanied me when I joined the group in October 2011, and it is fair to say that the group has come a long way since then. Along the way, I have had a lot of assistance from all the members of the group, and I would like to thank the members (in the chronological order of joining), Rushil, Vinay, Qiao, Suhas L, Anirudh, Mayank, Kowshik, and our honorary member, Aaron.

I am very fortunate to have many friends during all my years in Tempe - Rushil, Vinay, Prithwish, Suhas, Charan, Vimala and Mayank. I would like this opportunity to thank the first responders of the events of 4/3, Rohit and Sridhar who stayed with me that day, Charan and Vimala, who were very gracious in letting me stay at their apartment, making me feel at home all along and being the wonderful friends they have been, and Dr. Pavan Turaga and Prof. Grisha Coleman who went out of their way in visiting me, and the members of our group who called and offered me whatever assistance I needed. I greatly appreciate their generosity and help during my difficult times. I would like to thank my friends back home in India, Aditya, Sarvendranath, Rakshith and Pallavi with whom I have had enjoyable conversations over the years.

The game of Cricket has given me many inspirational heroes, who set the benchmark for commitment and desire to succeed. Growing up in a little town with hardly any decent English education on offer, my craze for cricket drove me to read literally every single cricket article that I could get my hands and eyes on, and those reading exercises helped hone my writing skills. As strange as it may sound, I owe it to the sport for being responsible, to a large extent, for whatever rudimentary English writing I can do today. Life in grad school would have been far more stressful without my cricket buddies, some of whom I have had many a discussion with, and many others with whom I have played. I would like to thank my cricket buddies - Vinayak, Anand, Praveen, Vinay Reddy, and my brother, Anudeep. Discussing cricket all these years, and celebrating the victories together has been immense fun. I would like to thank all members of the Desert Palm Village Cricket Club for all the intense and competitive games we have played together, and I will truly cherish all the memorable moments we have had together. In addition, I thank Chennai Express for the Chole Bathura, Idlis, Dosas, and lots of free *Cha*.

While it is impossible to put it into words the contribution of my family to whatever little I have done, I would like to write down few things about how they shaped my formative years in Ilkal. Realizing the far from ideal educational scene at my hometown, my parents created a parallel environment at home, wherein I was strongly encouraged to be fearless and take on challenges well beyond the routine tests and exams at school. I owe my strong background and immense interest in mathematics to my paternal grandfather. He taught me how to speak and ‘think’ the language of mathematics by constantly challenging my ability by posing difficult problems in algebra and geometry, outside of what the school textbooks had to offer. The mathematical skills I gained from those rigorous exercises held me in good stead during my PhD.

The work in this dissertation was supported by ONR Grant N00014-12-1-0124 sub award Z868302.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

Persistent surveillance from camera networks, such as at parking lots, UAVs, etc., often results in large amounts of video data, resulting in significant challenges for inference in terms of storage, communication and computation. All these applications are heavily resource-constrained and require low communication overheads in order to achieve real-time implementation. Consider the application of UAVs which provide real-time video and high resolution aerial images on demand. In these scenarios, it is typical to collect an enormous amount of data, followed by transmission of the same to a ground station using a low-bandwidth communication link. This results in expensive methods being employed for video capture, compression, and transmission implemented on the aircraft. The transmitted video is decompressed at a central station and then fed into a computer vision pipeline. Similarly, a video surveillance system which typically employs many high-definition cameras, gives rise to a prohibitively large amount of data, making it very challenging to store, transmit and extract meaningful information. Thus, there is a growing need to acquire as little data as possible and yet be able to perform high-level inference tasks like action recognition, object tracking reliably.

Recent advances in the areas of compressive sensing (CS) [23, 11, 12] have led to the development of new sensors like compressive cameras (also called single-pixel cameras (SPCs)) [65, 80], which enable the acquisition of **'more for less'** by greatly reducing the amount of sensed data while preserving most of its information. Another compelling application of SPC is in the area of infrared imaging. It is well known that short-wave infrared (SWIR) cameras have applications in military surveillance and maritime navigation because of their ability to ‘see-through’ in environmental conditions like fog, smoke, haze etc. However, the cost of a SWIR pixel is prohibitively expensive, and this has prevented infrared cameras from being employed in the applications outlined above. SPCs provide a cost-effective solution for image acquisition in such spectral regions. The SPC employs just a single photodiode sensitive to wavelengths of interest and a micro-mirror array to acquire images. This greatly reduces the cost of the camera. Current CS imaging systems, such as the commercially available short-wave infrared single pixel camera from Inview Technology Corporation, provide the luxury of reduced and fast acquisition of the image by optically computing only a small number random projections of the scene, thus enabling compression at the sensing level itself. Such characteristics of the acquisition system are highly sought-after in a) resource-constrained environments like UAVs where generally, computationally expensive methods are employed as a post-acquisition step to compress the fully acquired images, and b) applications such as Magnetic Resonance Imaging (MRI) [61] where traditional imaging methods are very slow. As an undesirable consequence, the computational load is now transferred to the decoding algorithm which reconstructs the image from the CS measurements or the random projections. ***The goal of this dissertation is to provide methods for effective information extraction from compressive cameras for computer vision applications and study the tradeoffs between reliability of performance and computational/storage load of the system in a resource constrained setting, that these methods offer***. SPCs differ from the conventional cameras in that they integrate the process of acquisition and compression by acquiring a small number of linear projections of the original images. More formally, when a sequence of images is acquired by a compressive camera, the measurements are generated by a sensing strategy which maps the space of $P \times Q$ images, $I \in \mathbb{R}^{PQ}$ to an observation space

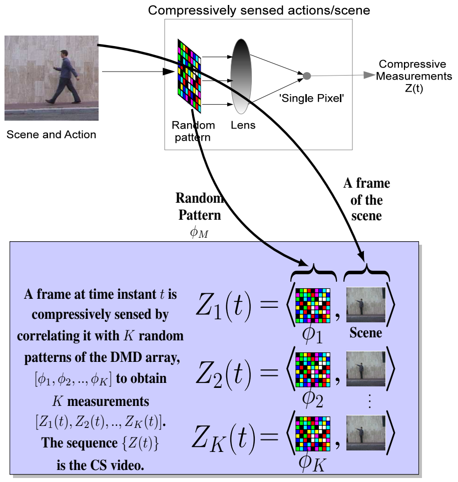

Compressively sensed actions/scene
Scene and Action
Random pattern
Lens
'Single Pixel'
Compressive Measurements Z(t)
Random Pattern ϕ_M
A frame of the scene
A frame at time instant t is compressively sensed by correlating it with K random patterns of the DMD array, $[\phi_1, \phi_2, .., \phi_K]$ to obtain K measurements $[Z_1(t), Z_2(t), .., Z_K(t)]$. The sequence $\{Z(t)\}$ is the CS video.
$Z_1(t) =$
ϕ_1
Scene
$Z_2(t) =$
ϕ_2
$\vdots$
$Z_K(t) =$
ϕ_K

Figure 1.1: Compressive Sensing (CS) of a scene: Every frame of the scene is compressively sensed by optically correlating random patterns with the frame to obtain CS measurements. The temporal sequence of such CS measurements is the CS video.

$Z \in \mathbb{R}^K$,

$$Z(t) = \phi I(t) + w(t), \tag{1.1}$$

where ϕ is a $K \times PQ$ measurement matrix, $w(t)$ is the noise, and $K \ll PQ$. The process is pictorially shown in Figure 1.1.

Difference between CS and video codecs It is worth noting at this point that the manner in which compression is achieved by SPCs differs fundamentally from the manner in which compression is achieved in JPEG images or MPEG videos. In the case of JPEG, the images are fully sensed and then compressed by applying wavelet transform or DCT to the sensed data, and in the case of MPEG, a video after having been sensed fully is compressed using a motion compensation technique. However, in the case of SPCs, at the outset one does not have direct access to full blown images, $\{I(t)\}$. SPCs instead provide us with compressed measurements $\{Z(t)\}$ directly by optically calculating inner products of the images, $\{I(t)\}$, with a set of test functions given by the rows of the measurement matrix, ϕ, implemented using a programmable micro-mirror array [65]. While this helps avoid the storage of a large amount of data and expensive computations for compression, it often comes at the expense of employing high computational load at the central station to reconstruct the video data perfectly. Moreover, for perfect reconstruction of the images, given a sparsity level of s, state-of-the-art algorithms require $O(s \log(PQ/s))$ measurements [12], which still amounts to a large fraction of the original data dimensionality. Hence, using SPCs may not always provide advantage with respect to communication resources since compressive measurements and transform coding of data require comparable bandwidth [13].

In this dissertation, firstly we present two approaches to information extraction from CS measurements, 1) a reconstruction-free approach to action recognition from compressive cameras, and 2) a non-iterative algorithm to reconstruct images from CS measurements, and secondly we present a method to design spatial-multiplexing measurements which are tuned to facilitate the easy extraction of visual features that are useful in computer vision applications like object tracking.

Spatio-temporal Smashed Filtering: First, we propose reconstruction-free methods for action recognition from compressive cameras at high compression ratios of 100 and above. Recognizing actions directly from CS measurements requires features which are mostly nonlinear and thus not easily applicable. This leads us to search for such properties that are preserved in compressive measurements. To this end, we propose the use of spatio-temporal smashed filters, which are compressive domain versions of pixel-domain matched filters. We conduct experiments on publicly available databases and show that one can obtain recognition rates that are comparable to the oracle method in uncompressed setup, even for high compression ratios.

A non-iterative CS reconstruction algorithm: Next, we present a non-iterative and more importantly an extremely fast algorithm to reconstruct images from compressively sensed (CS) random measurements. To this end, we propose a novel convolutional neural network (CNN) architecture which takes in CS measurements of an image as input and outputs an intermediate reconstruction. We call this network, ReconNet. The intermediate reconstruction is fed into an off-the-shelf denoiser to obtain the final reconstructed image. On a standard dataset of images we show significant improvements in reconstruction results (both in terms of PSNR and time complexity) over state-of-the-art iterative CS reconstruction algorithms at various measurement rates. Further, through qualitative experiments on real data collected using our block single pixel camera (SPC), we show that our network is highly robust to sensor noise and can recover visually better quality images than competitive algorithms at extremely low sensing rates of 0.1 and 0.04.

Reconstruction-free integral image estimation: Next, we propose a framework called **ReFInE** to directly obtain integral image estimates from a very small number of spatially multiplexed measurements of the scene without iterative reconstruction of any auxiliary image, and demonstrate their practical utility in visual object tracking. Specifically, we design measurement matrices which are tailored to facilitate extremely fast estimation of the integral image, by using a single-shot linear operation on the measured vector. Leveraging a prior model for the images, we formulate a nuclear norm minimization problem with second order conic constraints to jointly obtain the measurement matrix and the linear operator. Through qualitative and quantitative experiments, we show that high quality integral image estimates can be obtained using our framework at very low measurement rates. Further, on a standard dataset of 50 videos, we present object tracking results which are comparable to the state-of-the-art methods, even at an extremely low measurement rate of 1%.

A summary of contributions:

- We propose a correlation-based framework for action recognition and localization directly from compressed measurements, thus avoiding the costly reconstruction process.

- We provide principled ways to achieve quasi view-invariance in a spatio-temporal smashed filtering based action recognition setup.

- We further show that a single MACH filter for a canonical view is sufficient to generate MACH filters for all affine transformed views of the canonical view.

- Next, we propose a **non-iterative** and extremely fast reconstruction algorithm for block CS imaging [31]. To the best of our knowledge, there exists no published work which achieves these desirable features.

- We introduce a novel class of CNN architectures called **ReconNet** which takes in CS measurements of an image block as input and outputs the reconstructed image block. Further, the reconstructed image blocks are arranged appropriately and fed into an off-the-shelf denoiser to recover the full image.

- Through experiments on a standard dataset of images, we show that, in terms of mean PSNR of reconstructed images, our algorithm beats the nearest competitor by considerable margins at measurement rates of 0.1 and below. Further, we validate the robustness of ReconNet to arbitrary sensor noise by conducting qualitative experiments on real-data collected using our block SPC. We achieve visually superior quality reconstructions than the traditional CS algorithms.

- We demonstrate that the reconstructions retain rich semantic content even at a low measurement rate of 0.01. To this end, we present a proof of concept real-time application, wherein object tracking is performed on-the-fly as the frames are recovered from the CS measurements.

- Next, we propose a novel framework to recover estimates of integral images from a small number of spatially multiplexed measurements without iterative reconstruction of any auxillary image. We dub the framework **ReFInE** (Reconstruction-Free Integral Image Estimation).

- Leveraging the MGGD (multivariate generalized Gaussian distribution) prior model for the vector of detailed wavelet coefficients of natural images, we propose a nuclear norm minimization formulation to obtain a new specialized measurement matrix. We term the measurements acquired with such a measurement matrix, as **ReFInE** measurements.

- On a large dataset of 4952 images, we present qualitative and quantitative results to show that high quality estimates of integral images and box-filtered outputs can be recovered from **ReFInE** measurements in real-time.

- We show object tracking results, which are comparable to state-of-the-art methods, on a challenging dataset of 50 videos to demonstrate the utility of the box-filtered output estimates in tackling inference problems from SMCs at 1% measurement rate.

Chapter 2

RECONSTRUCTION-FREE ACTION INFERENCE FROM COMPRESSIVE IMAGERS

While a great body of work has focused on the theory and algorithms for signal recovery, much less attention has been paid to the question of whether it is possible to perform high-level inference directly on CS measurements without reconstruction. This question is interesting due to the following reasons: a) very often we want to know some property of the scene rather than the entire scene itself, b) good quality reconstruction results are difficult to achieve at compression ratios of 100 and above, and c) the parameters to be input to the reconstruction algorithm such as signal sparsity, sparsifying basis are not known, and are chosen in an ad-hoc manner. ***In this work, we consider the specific problem of action recognition in videos, and show that it is indeed possible to perform action recognition at extremely higher compression ratios, by bypassing reconstruction.*** We first show that approximate correlational features can be extracted directly from CS measurements. Using this in conjunction with the widely used correlational filters approach to recognition tasks in computer vision, we propose a spatio-temporal smashed filtering approach to action recognition, which results in robust performance at extremely high compression ratios.

2.1 Related work

a) Action Recognition The approaches in human action recognition from cameras can be categorized based on the low level features. Most successful representations of human action are based on features like optical flow, point trajectories, background subtracted blobs and shape, filter responses, etc. The current state-of-the-art approaches [94, 95] to action recognition are based on dense trajectories, which are extracted using dense optical flow. The dense trajectories are encoded by complex, hand-crafted descriptors like histogram of oriented gradients (HOG) [17] , histogram of oriented optical flow (HOOF) [15], HOG3D [51], and motion boundary histograms (MBH) [94]. However, the extraction of the above features involves various non-linear operations. This makes it very difficult to extract such features from compressively sensed images. For a detailed survey of action recognition, the readers are referred to [1].

b) Action recognition in compressed domain Though action recognition has a long history in computer vision, little exists in literature to recognize actions in the compressed domain. Yeo *et al.*[99] and Ozer *et al.*[71] explore compressed domain action recognition from MPEG videos by exploiting the spatiotemporal local structure, induced by the motion compensation technique used for compression. However, as stated above, the compression in CS cameras is achieved by randomly projecting the individual frames of the video onto a much lower dimensional space and hence does not easily allow leveraging motion information of the video. CS imagery acquires global measurements, thereby do not preserve any local information in their raw form, making action recognition much more difficult in comparison.

c) Reconstruction-free inference from CS videos Sankaranarayanan *et al.*[79] attempted to model videos as a LDS (Linear Dynamical System) by recovering parameters directly from compressed measurements, but is sensitive to spatial and view transforms, making it more suitable for recognition of dynamic textures than action recognition. Thirumalai *et al.*[88] introduced a reconstruction-free framework to obtain optical flow based on correlation estimation between two compressively sensed images. However, the method does not work well at very low measurement rates. Calderbank *et al.*[74] theoretically showed that 'learning directly in compressed domain is possible', and that with high probability the linear kernel SVM classifier in the compressed domain can be as accurate as best linear threshold classifier in the data domain. Recently, Kulkarni and Turaga [45] proposed a novel method based on recurrence textures for action recognition from compressive cameras. However, the method is prone to produce very similar recurrence textures even for dissimilar actions for CS sequences and is more suited for feature sequences as in [44].

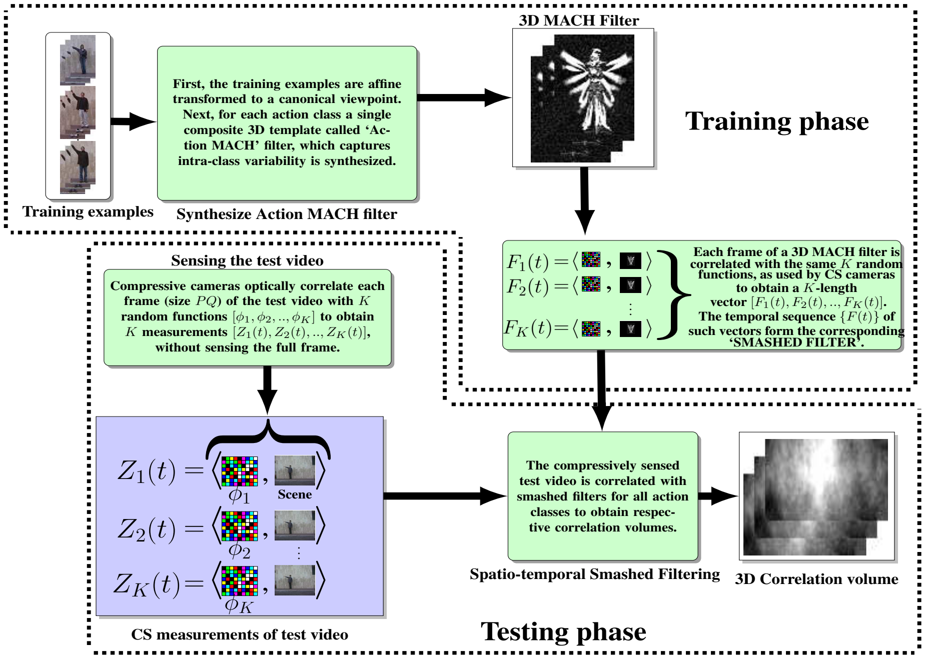

3D MACH Filter
Training phase
First, the training examples are affine transformed to a canonical viewpoint. Next, for each action class a single composite 3D template called 'Action MACH' filter, which captures intra-class variability is synthesized.
Training examples
Synthesize Action MACH filter
$F_1(t) = \langle$, $\rangle$
$F_2(t) = \langle$, $\rangle$
$F_K(t) = \langle$, $\rangle$
Each frame of a 3D MACH filter is correlated with the same K random functions, as used by CS cameras to obtain a K-length vector $[F_1(t), F_2(t), .., F_K(t)]$. The temporal sequence $\{F(t)\}$ of such vectors form the corresponding 'SMASHED FILTER'.
Sensing the test video
Compressive cameras optically correlate each frame (size PQ) of the test video with K random functions $[\phi_1, \phi_2, .., \phi_K]$ to obtain K measurements $[Z_1(t), Z_2(t), .., Z_K(t)]$, without sensing the full frame.
$Z_1(t) = \langle$, $\rangle$
ϕ_1
Scene
$Z_2(t) = \langle$, $\rangle$
ϕ_2
$Z_K(t) = \langle$, $\rangle$
ϕ_K
CS measurements of test video
The compressively sensed test video is correlated with smashed filters for all action classes to obtain respective correlation volumes.
Spatio-temporal Smashed Filtering
3D Correlation volume
Testing phase

Figure 2.1: Overview of our approach to action recognition from a compressively sensed test video. First, MACH [77] filters for different actions are synthesized offline from training examples and then compressed to obtain smashed filters. Next, the CS measurements of the test video are correlated with these smashed filters to obtain correlation volumes which are analyzed to determine the action in the test video.

d) Correlation filters in computer vision Even though, as stated above, the approaches based on dense trajectories extracted using optical flow information have yielded state-of-the-art results, it is difficult to extend such approaches while dealing with compressed measurements. Earlier approaches to action recognition were based on correlation filters, which were obtained directly from pixel data [49, 82, 81, 77, 18, 78]. The filters for different actions are correlated with the test video and the responses thus obtained are analyzed to recognize and locate the action in the test video. Davenport *et al.*[63] proposed a CS counterpart of the correlation filter based framework for target classification. Here, the trained filters are compressed first to obtain 'smashed filters', then the compressed measurements of the test examples are correlated with these smashed filters. Concisely, smashed filtering hinges on the fact that correlation between a reference signal and an input signal is nearly preserved even when they are projected onto a much lower-dimensional space. In this chapter, we show that spatio-temporal smashed filters provide a natural solution to reconstruction-free action recognition from compressive cameras. Our framework (shown in Figure 2.1) for classification includes synthesizing Action MACH (Maximum Average Correlation Height) filters [77] offline and then correlating the compressed versions of the filters with compressed measurements of the test video, instead of correlating raw filters with full-blown video, as is the case in [77]. Action MACH involves synthesizing a single 3D spatiotemporal filter which captures information about a specific action from a set of training examples. MACH filters can become ineffective if there are viewpoint variations in the training examples. To effectively deal with this problem, we also propose a quasi view-invariant solution, which can be used even in uncompressed setup.

2.2 Compressive action recognition

To devise a reconstruction-free method for action recognition from compressive cameras, we need to exploit such properties that are preserved robustly even in the compressed domain. One such property is the distance preserving property of the measurement matrix ϕ used for compressive sensing [12, 43]. Stated differently, the correlation between any two signals is nearly preserved even when the data is compressed to a much lower dimensional space. This makes correlation filters a natural choice to adopt. 2D correlation filters have been widely used in the areas of automatic target recognition and biometric applications like face recognition [106], palm print identification [73], etc., due to their ability to capture intraclass variabilities. Recently, Rodriguez *et al.*[77] extended this concept to 3D by using a class of correlation filters called MACH filters to recognize actions. As stated earlier, Davenport *et al.*[63] introduced the concept of smashed filters by implementing matched filters in the compressed domain. In the following section, we generalize this concept of smashed filtering to the space-time domain and show how 3D correlation filters can be implemented in the compressed domain for action recognition.

2.2.1 Spatio-temporal smashed filtering (STSF)

This section forms the core of our action recognition pipeline, wherein we outline a general method to implement spatio-temporal correlation filters using compressed measurements without reconstruction and subsequently, recognize actions using the response volumes. To this end, consider a given video $s(x, y, t)$ of size $P \times Q \times R$ and let $H_i(x, y, t)$ be the optimal 3D matched filter for actions $i = 1, .., N_A$, with size $L \times M \times N$ and N_A is the number of actions. First, the test video is correlated with the matched filters of all actions $i = 1, ..N_A$ to obtain respective 3D response volumes as in (2.1).

$$c_i(l,m,n) = \sum_{t=0}^{N-1}\sum_{y=0}^{M-1}\sum_{x=0}^{L-1} s(l+x, m+y, n+t)H_i(x,y,t). \tag{2.1}$$

Next, zero-padding each frame in H_i upto a size $P \times Q$ and changing the indices, (2.1) can be rewritten as:

$$c_i(l,m,n) = \sum_{t=0}^{N-1}\sum_{\beta=0}^{Q-1}\sum_{\alpha=0}^{P-1} s(\alpha, \beta, n+t)H_i(\alpha-l, \beta-m, t). \tag{2.2}$$

This can be written as the summation of N correlations in the spatial domain as follows:

$$c_i(l,m,n) = \sum_{t=0}^{N-1} \langle S_{n+t}, H_i^{l,m,t}\rangle, \tag{2.3}$$

where, $\langle , \rangle$ denotes the dot product, S_{n+t} is the column vector obtained by concatenating the Q columns of the $(n+t)^{th}$ frame of the test video. To obtain $H_i^{l,m,t}$, we first shift the t^{th} frame of the zeropadded filter volume H_i by l and m units in x and y respectively to obtain an intermediate frame and then rearrange it to a column vector by concatenating its Q columns. Due to the distance preserving property of measurement matrix ϕ, the correlations are nearly preserved in the much lower dimensional compressed domain. To state the property more specifically, using JL Lemma [43], the following relation can be shown:

$$c_i(l,m,n) - N\epsilon \leq \sum_{t=0}^{N-1} \langle \phi S_{n+t}, \phi H_i^{l,m,t}\rangle \leq c_i(l,m,n) + N\epsilon. \tag{2.4}$$

The derivation of this relation and the precise form of ϵ is as follows. In the following, we derive the relation between the response volume from uncompressed data and response volume obtained using compressed data. According to JL Lemma [43], given $0 < \epsilon < 1$, a set $\mathcal{S}$ of $2V$ points in $\mathbb{R}^{PQ}$, each with unit norm, and $K > \mathcal{O}(\frac{\log(V)}{\epsilon^2})$, there exists a Lipschitz function $f : \mathbb{R}^{PQ} \to \mathbb{R}^K$ such that

$$\begin{aligned}(1-\epsilon)\|S_{n+t} - H_i^{l,m,t}\|^2 &\leq \|f(S_{n+t}) - f(H_i^{l,m,t})\|^2 \\ &\leq (1+\epsilon)\|S_{n+t} - H_i^{l,m,t}\|^2, \end{aligned} \tag{2.5}$$

and

$$\begin{aligned}(1-\epsilon)\|S_{n+t}+H_i^{l,m,t}\|^2 &\leq \|f(S_{n+t})+f(H_i^{l,m,t})\|^2 \\ &\leq (1+\epsilon)\|S_{n+t}+H_i^{l,m,t}\|^2, \end{aligned} \tag{2.6}$$

$\forall$ S_{n+t} and $H_i^{l,m,t} \in \mathcal{S}$. Now we have:

$$\begin{aligned} & 4\langle f(S_{n+t}), f(H_i^{l,m,t})\rangle \\ &= \|f(S_{n+t})+f(H_i^{l,m,t})\|^2 - \|f(S_{n+t})-f(H_i^{l,m,t})\|^2 \\ &\geq (1-\epsilon)\|S_{n+t}+H_i^{l,m,t}\|^2 - (1+\epsilon)\|S_{n+t}-H_i^{l,m,t}\|^2 \\ &= 4\langle S_{n+t}, H_i^{l,m,t}\rangle - 2\epsilon(\|S_{n+t}\|^2+\|H_i^{l,m,t}\|^2) \\ &\geq 4\langle S_{n+t}, H_i^{l,m,t}\rangle - 4\epsilon. \end{aligned} \tag{2.7}$$

We can get a similar relation for opposite direction, which when combined with (2.7), yields the following:

$$\begin{aligned}\langle S_{n+t}, H_i^{l,m,t}\rangle - \epsilon &\leq \langle f(S_{n+t}), f(H_i^{l,m,t})\rangle \\ &\leq \langle S_{n+t}, H_i^{l,m,t}\rangle + \epsilon. \end{aligned} \tag{2.8}$$

However, JL Lemma does not provide us with a embedding, f which satisfies the above relation. As discussed in [19], f can be constructed as a matrix, ϕ with size $K \times PQ$, whose entries are either independent realizations of Gaussian random variables or independent realizations of ± Bernoulli random variables. Now, if ϕ constructed as explained above is used as measurement matrix, then we can replace f in (2.8) by ϕ, leading us to

$$\begin{aligned}\langle S_{n+t}, H_i^{l,m,t}\rangle - \epsilon &\leq \langle \phi S_{n+t}, \phi H_i^{l,m,t}\rangle \\ &\leq \langle S_{n+t}, H_i^{l,m,t}\rangle + \epsilon. \end{aligned} \tag{2.9}$$

Hence, we have,

$$\sum_{t=0}^{N-1}\langle S_{n+t}, H_i^{l,m,t}\rangle - N\epsilon \leq \sum_{t=0}^{N-1}\langle \phi S_{n+t}, \phi H_i^{l,m,t}\rangle$$

$$\leq \sum_{t=0}^{N-1}\langle S_{n+t}, H_i^{l,m,t}\rangle + N\epsilon. \tag{2.10}$$

Using equations (4) and (2.10), we arrive at the following desired equation.

$$c_i(l,m,n) - N\epsilon \leq \sum_{t=0}^{N-1}\langle \phi S_{n+t}, \phi H_i^{l,m,t}\rangle \leq c_i(l,m,n) + N\epsilon. \tag{2.11}$$

Now allowing for the error in correlation, we can compute the response from compressed measurements as below:

$$c_i^{comp}(l,m,n) = \sum_{t=0}^{N-1}\langle \phi S_{n+t}, \phi H_i^{l,m,t}\rangle. \tag{2.12}$$

The above relation provides us with the 3D response volume for the test video with respect to a particular action, without reconstructing the frames of the test video. To reduce computational complexity, the 3D response volume is calculated in frequency domain via 3D FFT.

Feature vector and Classification using SVM For a given test video, we obtain N_A correlation volumes. For each correlation volume, we adapt three level volumetric max-pooling to obtain a 73 dimensional feature vector [78]. In addition, we also compute peak-to-side-lobe-ratio for each of these 73 maxpooled values. PSR is given by $PSR_k = \frac{peak_i - \mu_i}{\sigma_i}$,where $peak_k$ is the k^{th} max-pooled value, and μ_k and σ_k are the mean and standard deviation values in its small neighbourhood. Thus, the feature vector for a given test video is of dimension, $N_A \times 146$. This framework can be used in any reconstruction-free application from compressive cameras which can be implemented using 3D correlation filtering. Here, we assume that there exists an optimal matched filter for each action and outline a way to recognize actions from compressive measurements. In the next section, we show how these optimal filters are obtained for each action.

2.2.2 Training filters for action recognition

The theory of training correlation filters for any recognition task is based on synthesizing a single template from training examples, by finding an optimal tradeoff between certain performance measures. Based on the performance measures, there exist a number of classes of correlation filters. A MACH filter is a single filter that encapsulates the information of all training examples belonging to a particular class and is obtained by optimizing four performance parameters, the Average Correlation Height (ACH), the Average Correlation Energy (ACE), the Average Similarity Measure (ASM), and the Output Noise Variance (ONV). Until recently, this was used only in two dimensional applications like palm print identification [73], target recognition [84] and face recognition problems [106]. For action recognition, Rodriguez et al. [77] introduced a generalized form of MACH filters to synthesize a single action template from the spatio-temporal volumes of the training examples. Furthermore, they extended the notion for vector-valued data. In our framework for compressive action recognition, we adopt this approach to train matched filters for each action. Here, we briefly give an overview of 3D MACH filters which was first described in [77].

First, temporal derivatives of each pixel in the spatio-temporal volume of each training sequence are computed and the frequency domain representation of each volume is obtained by computing a 3D-DFT of that volume, according to the following:

$$F(\mathbf{u}) = \sum_{t=0}^{N-1} \sum_{x_2=0}^{M-1} \sum_{x_1=0}^{L-1} f(\mathbf{x}) e^{(-j2\pi(\mathbf{u}\cdot\mathbf{x}))}, \tag{2.13}$$

where, $f(\mathbf{x})$ is the spatio-temporal volume of L rows, M columns and N frames, $F(\mathbf{u})$

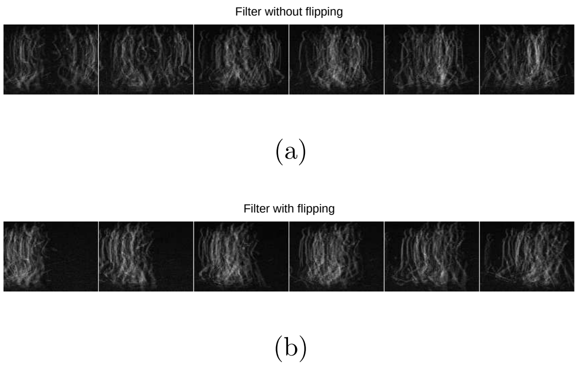
is its spatio-temporal representation in the frequency domain and $\mathbf{x} = (x_1, x_2, t)$ and $\mathbf{u} = (u_1, u_2, u_3)$ denote the indices in space-time and frequency domain respectively. If N_e is the number of training examples for a particular action, then we denote their 3D DFTs by $X_i(\mathbf{u}), i = 1, 2, .., N_e$, each of dimension, $d = L \times M \times N$. The average spatio-temporal volume of the training set in the frequency domain is given by $M_x(\mathbf{u}) = \frac{1}{N_e} \sum_{i=1}^{N_e} X_i(\mathbf{u})$. The average power spectral density of the training set is given by $D_x(\mathbf{u}) = \frac{1}{N_e} \sum_{i=1}^{N_e} |X_i(\mathbf{u})|^2$, and the average similarity matrix of the training set is given by $S_x(\mathbf{u}) = \frac{1}{N_e} \sum_{i=1}^{N_e} |X_i(\mathbf{u}) - M_x(\mathbf{u})|^2$. Now, the MACH filter for that action is computed by minimizing the average correlation energy, average similarity measure, output noise variance and maximizing the average correlation height. This is done by computing the following:

$$h(\mathbf{u}) = \frac{1}{[\alpha C(\mathbf{u}) + \beta D_x(\mathbf{u}) + \gamma S_x(\mathbf{u})]} M_x(\mathbf{u}), \tag{2.14}$$

where, $C(\mathbf{u})$ is the noise variance at the corresponding frequency. Generally, it is set to be equal to 1 at all frequencies. The corresponding space-time domain representation $H(x, y, t)$ is obtained by taking the inverse 3D DFT of h. A filter with response volume H and parameters α, β and γ is compactly written as $\mathbf{H} = \{H, \alpha, \beta, \gamma\}$.

2.3 Affine Invariant Smashed Filtering

Even though MACH filters capture intra-class variations, the filters can become ineffective if viewpoints of the training examples are different or if the viewpoint of the test video is different from viewpoints of the training examples. Filters thus obtained may result in misleading correlation peaks. Consider the case of generating a filter of a translational action, walking, wherein the training set is sampled from two different views. The top row in Fig 2.2 depicts some frames of the filter, say ‘Type-1’ filter, generated out of such a training set. The bottom row depicts some frames of the filter, say ‘Type-2’ filter, generated by affine transforming all examples in the training set to a canonical viewpoint. Roughly speaking, the ‘Type-2’ filter can be interpreted as



(a)

(b)

Figure 2.2: a) ‘Type-1’ filter obtained for walking action where the training examples were from different viewpoints b) ‘Type-2’ filter obtained from the training examples by bringing all the training examples to the same viewpoint. In (a), two groups of human move in opposite directions and eventually merge into each other, thus making the filter ineffective. In (b), the merging effect is countered by transforming the training set to the same viewpoint.

many humans walking in the same direction, whereas the ‘Type-1’ filter, as 2 groups of humans, walking in opposite directions. One can notice that some of the frames in the ‘Type-1’ do not represent the action of interest, particularly the ones in which the two groups merge into each other. This kind of merging effect will become more prominent as the number of different views in the training set increases. The problem is avoided in the ‘Type-2’ filter because of the single direction of movement of the whole group. Thus, it can be said that the quality of information about the action in the ‘Type-2’ filter is better than that in the ‘Type-1’ filter. As we show in experiments, this is indeed the case. Assuming that all views of all training examples are affine transforms of a canonical view, we can synthesize a MACH filter generated after transforming all training examples to a common viewpoint and avoid the merging effect. However, different test videos may be in different viewpoints, which makes it impractical to synthesize filters for every viewpoint. Hence it is desirable that a single representative filter be generated for all affine transforms of a canonical view. The following proposition asserts that, from a MACH filter defined for the canonical view, it is possible to obtain a compensated MACH filter for any affine transformed view.

Proposition 1 *Let $\mathbf{H} = \{H, \alpha, \beta, \gamma\}$ denote the MACH filter in the canonical view, then for any arbitrary view V, related to the canonical view by an affine transformation, $[A|\mathbf{b}]$, there exists a MACH filter, $\hat{\mathbf{H}} = \{\hat{H}, \hat{\alpha}, \hat{\beta}, \hat{\gamma}\}$ such that: $\hat{H}(\mathbf{x_s}, t) = |\Delta|^2 H(A\mathbf{x_s} + \mathbf{b}, t)$, $\hat{\alpha} = |\Delta|^2\alpha$, $\hat{\beta} = \beta$ and $\hat{\gamma} = \gamma$ where $\mathbf{x_s} = (x_1, x_2)$ denote the horizontal and vertical axis indices and Δ is the determinant of A.*

Proof: Consider the frequency domain response $\hat{h}$ for view V, given by the following.

$$\hat{h}(\mathbf{u}) = \frac{1}{(\alpha\hat{C}(\mathbf{u}) + \beta\hat{D}_x(\mathbf{u}) + \gamma\hat{S}_x(\mathbf{u}))}\hat{M}_x(\mathbf{u}). \tag{2.15}$$

For the sake of convenience, we let $\mathbf{u} = (\mathbf{u_s}, u_3)$ where $\mathbf{u_s} = (u_1, u_2)$ denotes the spatial frequencies and u_3, the temporal frequency. Now using properties of the Fourier transform [7], we have,

$$\hat{M}_x(\mathbf{u_s}, u_3) = \frac{1}{N_e}\sum_{i=1}^{N_e}\hat{X}_i(\mathbf{u_s}, \mathbf{u_3})$$

$$= \frac{1}{N_e}\sum_{i=1}^{N_e}\frac{e^{j2\pi\mathbf{b}\cdot(A^{-1})^T\mathbf{u_s}}X_i((A^{-1})^T\mathbf{u_s}, u_3)}{|\Delta|}.$$

Using the relation $M_x(\mathbf{u}) = \frac{1}{N_e}\sum_{i=1}^{N_e} X_i(\mathbf{u})$, we get,

$$\hat{M}_x(\mathbf{u_s}, u_3) = \frac{e^{j2\pi\mathbf{b}\cdot(A^{-1})^T\mathbf{u_s}}M_x((A^{-1})^T\mathbf{u_s}, u_3)}{|\Delta|}. \tag{2.16}$$

Now,

$$\hat{D}_x(\mathbf{u_s}, u_3) = \frac{1}{N_e}\sum_{i=1}^{N_e} |\hat{X}_i(\mathbf{u_s}, u_3)|^2$$

$$= \frac{1}{N_e}\sum_{i=1}^{N_e} |\frac{e^{j2\pi\mathbf{b}\cdot(A^{-1})^T\mathbf{u_s}} X_i((A^{-1})^T\mathbf{u_s}, u_3)}{|\Delta|}|^2$$

$$= \frac{1}{N_e}\sum_{i=1}^{N_e} |\frac{X_i((A^{-1})^T\mathbf{u_s}, u_3)}{|\Delta|}|^2 . (\because |e^{j2\pi\mathbf{b}\cdot(A^{-1})^T\mathbf{u_s}}| = 1)$$

Hence, using the relation $D_x(\mathbf{u}) = \frac{1}{N_e}\sum_{i=1}^{N_e} |X_i(\mathbf{u})|^2$, we have

$$\hat{D}_x(\mathbf{u_s}, u_3) = \frac{1}{|\Delta|^2} D_x((A^{-1})^T\mathbf{u_s}, u_3). \tag{2.17}$$

Similarly, it can be shown that

$$\hat{S}_x(\mathbf{u_s}, u_3) = \frac{1}{|\Delta|^2} S_x((A^{-1})^T\mathbf{u_s}, u_3). \tag{2.18}$$

Using (2.16), (2.17) and (2.18) in (2.15), we have,

$$\hat{h}(\mathbf{u}) = (e^{j2\pi\mathbf{b}\cdot(A^{-1})^T\mathbf{u_s}} M_x((A^{-1})^T\mathbf{u_s}, u_3))\Delta \\ \frac{1}{(\hat{\alpha}|\Delta|^2\hat{C}(\mathbf{u}) + \hat{\beta}D_x((A^{-1})^T\mathbf{u_s}, u_3) + \hat{\gamma}S_x((A^{-1})^T\mathbf{u_s}, u_3)}. \tag{2.19}$$

Now letting, $\alpha = \hat{\alpha}|\Delta|^2$, $\beta = \hat{\beta}$, $\gamma = \hat{\gamma}$, $\hat{C}(\mathbf{u}) = C(\mathbf{u}) = C((A^{-1})^T\mathbf{u_s}, u_3))$ (since C is usually assumed to be equal to 1 at all frequencies if noise model is not available) and using (2.14), we have,

$$\hat{h}(\mathbf{u}) = \Delta h((A^{-1})^T\mathbf{u_s}, u_3))e^{j2\pi\mathbf{b}\cdot(A^{-1})^T\mathbf{u_s}}. \tag{2.20}$$

Now taking the inverse 3D-FFT of $\hat{h}(\mathbf{u})$, we have,

$$\hat{H}(\mathbf{x_s}, t) = |\Delta|^2 H(A\mathbf{x_s} + \mathbf{b}, t). \tag{2.21}$$

Thus, a compensated MACH filter for the view V is given by $\hat{\mathbf{H}} = \{\hat{H}, \hat{\alpha}, \hat{\beta}, \hat{\gamma}\}$. This completes the proof of the proposition. Thus a MACH filter for view V, with parameters $|\Delta|^2\alpha$, β and γ can be obtained just by affine transforming the frames of the MACH filter for the canonical view. Normally $|\Delta| \approx 1$ for small view changes. Thus, even though in theory, $\hat{\alpha}$ is related to α by a scaling factor of $|\Delta|^2$, for small view changes, $\hat{h}$ is the optimal filter with essentially the same parameters as those for the canonical view. This result shows that for small view changes, it is possible to build robust MACH filters from a single canonical MACH filter.

Robustness of affine invariant smashed filtering To corroborate the need of affine transforming the MACH filters to the viewpoint of the test example, we conduct the following two synthetic experiments. In the first, we took all examples in Weizmann dataset and assumed that they belong to the same view, dubbed as the canonical view. We generated five different datasets, each corresponding to a different viewing angle. The different viewing angles from $0°$ to $20°$ in increments of $5°$ were simulated by means of homography. For each of these five datasets, a recognition experiment is conducted using filters for the canonical view as well as the compensated filters for their respective viewpoints, obtained using (2.21). The average PSR in both cases for each viewpoint is shown in Figure 2.3. The mean PSR values obtained using compensated filters are more than those obtained using canonical filters.

In the second experiment, we conducted five independent recognition experiments for the dataset corresponding to fixed viewing angle of $15°$, using compensated filters generated for five different viewing angles. The results are tabulated in table 2.1. It is evident that action recognition rate is highest when the compensated filters used correspond to the viewing angle of the test videos. These two synthetic experiments clearly suggest that it is essential to affine transform the filters to the viewpoint of the test video before performing action recognition.

Viewing angle	Canonical	5°	10°	15°	20°
Recognition rate	65.56	68.88	67.77	**72.22**	66.67

Table 2.1: Action recognition rates for the dataset corresponding to fixed viewing angle of 15° using compensated filters generated for various viewing angles. As expected, action recognition rate is highest when the compensated filters used correspond to the viewing angle of the test videos.

2.4 Experimental results

For all our experiments, we use a measurement matrix, ϕ whose entries are drawn from i.i.d. standard Gaussian distribution, to compress the frames of the test videos. We conducted extensive experiments on the widely used Weizmann [6], UCF sports [77], UCF50 [76] and HMDB51 [54] datasets to validate the feasibility of action recognition from compressive cameras. Before we present the action recognition results, we briefly discuss the baseline methods to which we compare our method, and describe a simple to perform action localization in those videos in which the action is recognized successfully.

Baselines As noted earlier, this is the first work to tackle the problem of action recognition from compressive cameras. The absence of precedent approach to this problem makes it difficult to decide on the baseline methods to compare with. The state-of-the-art methods for action recognition from traditional cameras rely on dense trajectories [95], derived using highly non-linear features, HOG [17], HOOF [15], and MBH [94]. At the moment, it is not quite clear on how to extract such features directly from compressed measurements. Due to these difficulties, we fixate on two baselines. The first baseline method is the Oracle MACH, wherein action recognition

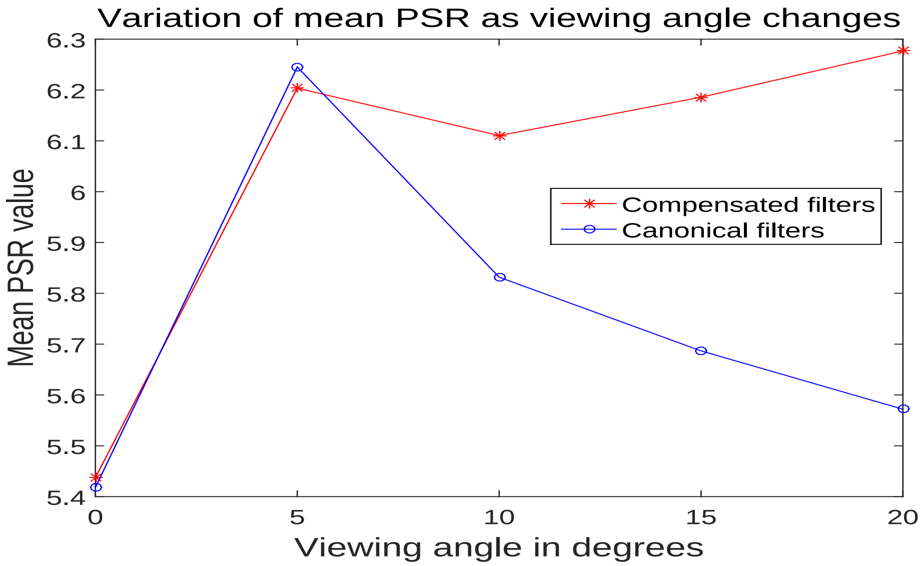


Figure 2.3: The mean PSRs for different viewpoints for both canonical filters and compensated filters are shown. The mean PSR values obtained using compensated filters are more than those obtained using canonical filters, thus corroborating the need of affine transforming the MACH filters to the viewpoint of the test example.

is performed as in [77] and for the second baseline, we first reconstruct the frames from the compressive measurements, and then apply the improved dense trajectories (IDT) method [95], which is the most stable state-of-the-art method, on the reconstructed video to perform action recognition. There are two approaches that one can follow to reconstruct the frames of a CS video. One of them is the naive frame-by-frame reconstruction approach, and the other one, a more sophisticated approach dubbed as video compressive sensing, involves alternating between motion estimation and motion-compensated signal recovery. We note that even the best performing video CS reconstruction algorithms [80] take about 2-3 hours to recover the video clips we deal with in this work. We have around 7000 clips in each of the two datasets, UCF50 and HMDB51. We realized that adopting video CS reconstruction for such a large dataset is computationally infeasible. Hence, we adopt the former approach, more specifically the CoSamP algorithm [70] to reconstruct the frames of the video. We use the code made publicly available by the authors, and set all the parameters to default to obtain improved dense trajectory (IDT) features. The features thus obtained are encoded using Fisher vectors, and a linear SVM is used for classification. Henceforth, we refer this method as Recon+IDT.

Spatial Localization of action from compressive cameras without reconstruction Action localization in each frame is determined by a bounding box centred at location (l^{max}) in that frame, where l^{max} is determined by the peak response (response corresponding to the classified action) in that frame and the size of the filter corresponding to the classified action. To determine the size of the bounding box for a particular frame, the response values inside a large rectangle of the size of the filter, and centred at l^{max} in that frame are normalized so that they sum up to unity. Treating this normalized rectangle as a 2D probability density function, we determine the bounding box to be the largest rectangle centred at l^{max}, whose sum is less than a value, $\lambda \leq 1$. For our experiments, we use λ equal to 0.7.

Computational complexity In order to show the substantial computational savings achievable in our STSF framework of reconstruction-free action recognition from compressive cameras, we compare the computational time of the framework with that of Recon+IDT. We ran our experiments on a Intel i7 quad core machine with 16GB RAM to report the timing numbers.

Compensated Filters In section 3, we experimentally showed that better action recognition results can be obtained if compensated filters are used instead of canonical view filters (table 2.1). However, to generate compensated filters, one requires the information regarding the viewpoint of the test video. Generally, the viewpoint of the test video is not known. This difficulty can be overcome by generating compensated filters corresponding to various viewpoints. In our experiments, we restrict our filters to two viewpoints described in section 3, i.e we use 'Type-1' and 'Type-2' filters.

2.4.1 Reconstruction-free recognition on Weizmann dataset

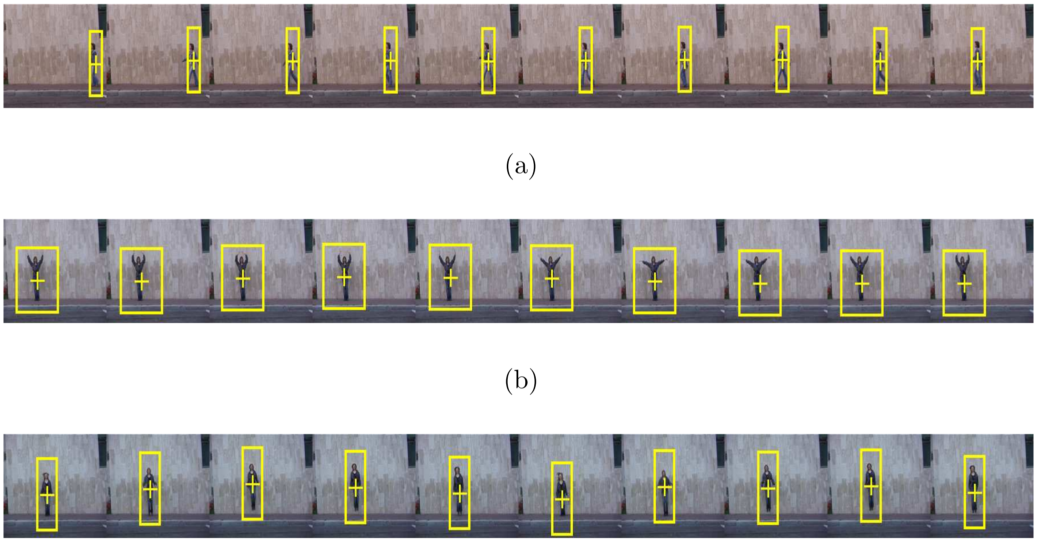
Even though it is widely accepted in the computer vision community that Weizmann dataset is an easy dataset, with many methods achieving near perfect action recognition rates, we believe that working with compressed measurements precludes the use of those well-established methods, and obtaining such high action recognition rates at compression ratios of 100 and above even for a simple dataset as Weizmann is not straightforward. The Weizmann dataset contains 10 different actions, each performed by 9 subjects, thus making a total of 90 videos. For evaluation, we used the leave-one-out approach, where the filters were trained using actions performed by 8 actors and tested on the remaining one. The results shown in table 2.2 indicate that our method clearly outperforms the Recon+IDT. It is quite evident that with full-blown frames (indicated in table 2.2) that Recon+IDT method performs much better than STSF method. However, at compression ratios of 100 and above, recognition rates are very stable for our STSF framework, while Recon+IDT fails completely. This is due to the fact that Recon+IDT operates on reconstructed frames, which are of poor quality at such high compression ratios, while STSF operates directly on compressed measurements. The recognition rates are stable even at high compression ratios and are comparable to the recognition accuracy for the Oracle MACH (OM) method [2]. The average time taken by STSF and Recon+IDT to process a video of size 144 × 180 × 50 are shown in parentheses in table 1. Recon+IDT takes about 20-35 minutes to process one video, with the frame-wise reconstruction of the video being the dominating component in the total computational time, while STSF frame-

Compression factor	STSF	Recon + IDT
1	81.11 (3.22s) (OM [2, 77])	100 (3.1s)
100	81.11 (3.22s)	5.56 (1520s)
200	81.11 (3.07s)	10 (1700s)
300	76.66 (3.1s)	10 (1800s)
500	78.89 (3.08s)	7.77 (2000s)

Table 2.2: Weizmann dataset: Recognition rates for reconstruction-free recognition from compressive cameras for different compression factors are stable even at high compression factors of 500. Our method clearly outperforms Recon+IDT method and achieves a recognition rate which is comparable to the recognition rate of 81.11 in the case of Oracle MACH [2, 77].

work takes only a few seconds for the same sized video since it operates directly on compressed measurements.

Spatial localization of action from compressive cameras without reconstruction Further, to validate the robustness of action detection using the STSF framework, we quantified action localization in terms of error in estimation of the subject's centre from its ground truth. The subject's centre in each frame is estimated as the centre of the fixed sized bounding box with location of the peak response (only the response corresponding to the classified action) in that frame as it left-top corner. Figure 2.4 shows action localization in a few frames for various actions of the dataset. Figure 2.5 shows that using these raw estimates, on average, the error from the ground truth is less than or equal to 15 pixels in approximately 70% of the frames, for compression ratios of 100, 200 and 300. It is worth noting that using our framework it is possible to obtain robust action localization results without reconstructing the images, even at extremely high compression ratios.

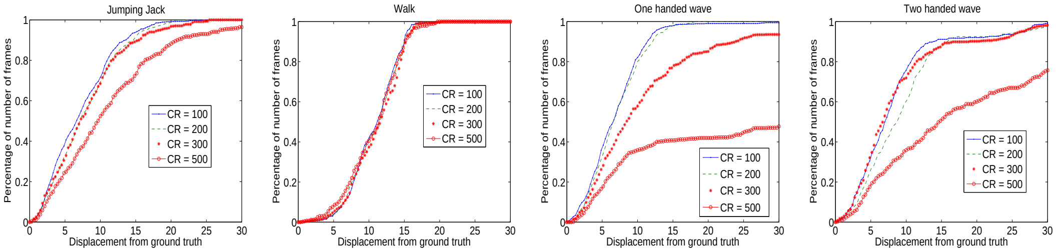


(a)

(b)

(c)

Figure 2.4: Reconstruction-free spatial localization of subject at compression ratio = 100 for different actions in Weizmann dataset. a) Walking b) Two handed wave c) Jump in place



(a)

(b)

(c)

(d)

Figure 2.5: Localization error for Weizmann dataset. X-axis : Displacement from ground truth. Y-axis: Fraction of total number of frames for which the displacement of subject's centre from ground truth is less than or equal to the value in x-axis. On average, for approximately 70% of the frames, the displacement of ground truth is less than or equal to 15 pixels, for compression ratios of 100, 200 and 300.

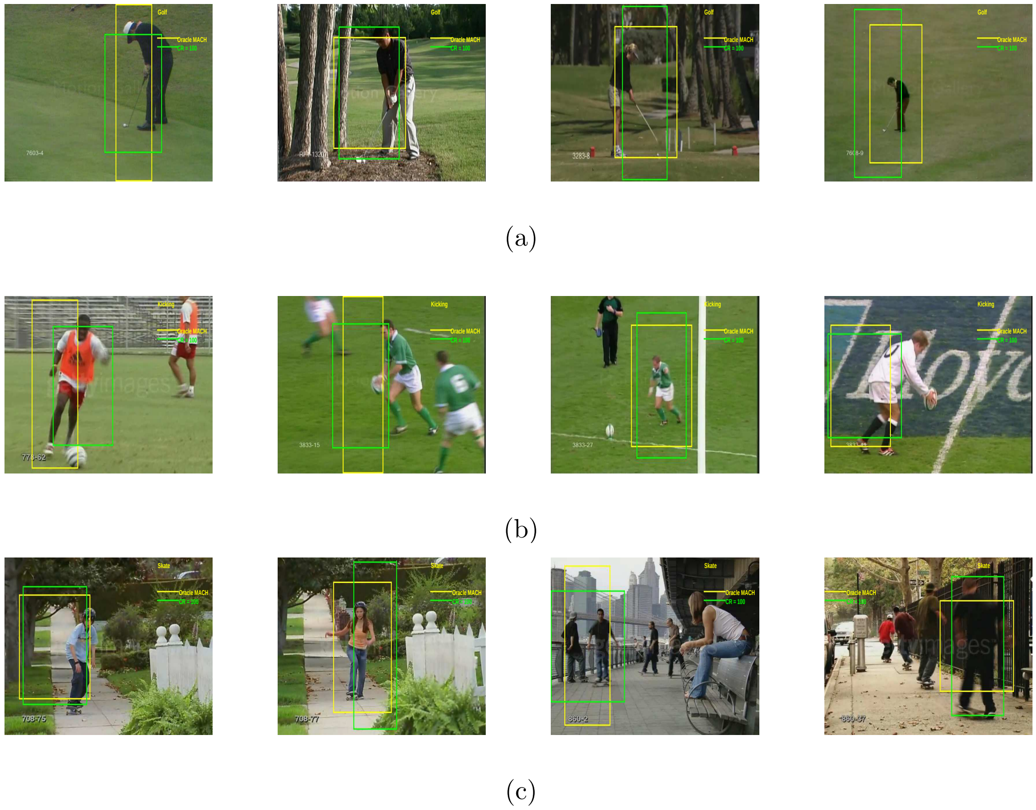
2.4.2 Reconstruction-free recognition on UCF sports dataset

The UCF sports action dataset [77] contains a total of 150 videos across 9 different actions. The dataset is a challenging dataset with scale and viewpoint variations. For testing, we use leave-one-out cross validation. At compression ratio of 100 and 300, the recognition rates are 70.67% and 68% respectively. The rates obtained are comparable to those obtained in Oracle MACH set-up [77] (69.2%). Considering the difficulty of the dataset, these results are very encouraging. The confusion matrix for compression ratios 100 is shown in table 2.3.

Action	Golf-Swing	Kicking	Riding Horse	Run-Side	Skate-Boarding	Swing	Walk	Diving	Lifting
Golf-Swing	**77.78**	16.67	0	0	0	0	5.56	0	0
Kicking	0	**75**	0	5	5	10	5	0	0
Riding Horse	16.67	16.67	**41.67**	8.33	8.33	0	8.33	0	0
Run-Side	0	0	0	**61.54**	7.69	15.38	7.69	7.69	0
Skate-Boarding	0	8.33	8.33	25	**50**	0	5	0	0
Swing	0	3.03	12.12	0.08	3.03	**78.79**	3.03	0	0
Walk	0	9.09	4.55	4.55	9.09	9.09	**63.63**	0	0
Diving	0	0	0	0	7.14	0	0	**92.86**	0
Lifting	0	0	0	0	0	0	0	16.67	**83.33**

Table 2.3: Confusion matrix for UCF sports database at a compression factor = 100. Recognition rate for this scenario is 70.67 %, which is comparable to Oracle MACH [77] (69.2%).

Spatial localization of action from compressive cameras without reconstruction Figure 2.6 shows action localization for some correctly classified instances across various actions in the dataset, for Oracle MACH and compression ratio = 100. It can be seen that action localization is estimated reasonably well despite large scale variations and extremely high compression ratio.



(a)

(b)

(c)

Figure 2.6: Reconstruction-free spatial localization of subject for Oracle MACH (shown as yellow box) and STSF (shown as green box) at compression ratio = 100 for some correctly classfied instances of various actions in the UCF sports dataset. a) Golf b) Kicking c) Skate-Boarding. Action localization is estimated reasonably well directly from CS measurements even though the measurements themselves do not bear any explicit information regarding pixel locations.

2.4.3 Reconstruction-free recognition on UCF50 dataset

To test the scalability of our approach, we conduct action recognition on large datasets, UCF50 [76] and HMDB51 [54]. Unlike the datasets considered earlier, these two datasets have large intra-class scale variability. To account for this scale variability, we generate about 2-6 filters per action. To generate MACH filters, one requires bounding box annotations for the videos in the datasets. Unfortunately frame-wise bounding box annotations are not available for these two datasets. Hence, we selected 190 video clips from UCF50 dataset with 2-6 video clips per action. We manually annotated these clips with frame-wise bounding boxes. Each MACH filter is generated with just one of these videos as a training example. In total we generate 380 filters (190 canonical filters, i.e 'Type-1 filters' + 190 their flipped versions, i.e 'Type-2' filters). The UCF50 database consists of 50 actions, with around 120 clips per action, totalling upto 6681 videos. The database is divided into 25 groups with each group containing between 4-7 clips per action. We use leave-one-group cross-validation to evaluate our framework. The recognition rates at different compression ratios, and the mean time taken for one clip (in parentheses) for our framework and Recon+IDT are tabulated in table 2.4. Table 2.4 also shows the recognition rates for various state-of-the-art action recognition methods, while operating on the full-blown images, as indicated in the table by (FBI). Two conclusions follow from the table. 1) Our approach outperforms the baseline method, Recon+IDT at very high compression ratios of 100 and above, and 2) the mean time per clip is less than that for Recon+IDT method. This clearly suggests that when operating at high compression ratios, it is better to perform action recognition without reconstruction than reconstructing the frames and then applying a state-of-the-art method. The recognition rates for individual classes for Oracle MACH (OM), and compression ratios, 100 and 400 are given in table 2.5. The action localization results for various actions are shown in figure 2.7. The bounding boxes in most instances correspond to the human or the moving part of the human or the object of interest. Note how the sizes of the bounding boxes are commensurate with the area of the action in each frame. For example, for the fencing action, the bounding box covers both the participants, and for the playing piano action, the bounding box covers just the hand of the participant. In the case of breaststroke action, where human is barely visible, action localization results are impressive. We emphasize that action localization is achieved directly from compressive measurements without any intermediate reconstruction, even though the measurements do not bear any explicit information regarding pixel locations. We note that the procedure outlined above is by no means a full-fledged procedure for action localization and is fundamentally different from the those in [56, 89], where sophisticated models are trained jointly on action labels and the location of person in each frame, and action and its localization are determined simultaneously by solving one computationally intensive inference problem. While our method is simplistic in nature and does not always estimate localization accurately, it relies only on minimal post-processing of the correlation response, which makes it an attractive solution for action localization in resource-constrained environments where a rough estimate of action location may serve the purpose. However, we do note that action localization is not the primary goal of the work and that the purpose of this exercise is to show that reasonable localization results directly from compressive measurements are possible, even using a rudimentary procedure as outlined above. This clearly suggests that with more sophisticated models, better reconstruction-free action localization results can be achieved. One possible option is to co-train models jointly on action labels and annotated bounding boxes in each frame similar to [56, 89], while extracting spatiotemporal features such as HOG3D [51] features for correlation response volumes, instead of the input video.

2.4.4 Reconstruction-free recognition on HMDB51 dataset

The HMDB51 database consists of 51 actions, with around 120 clips per action, totalling upto 6766 videos. The database is divided into three train-test splits. The average recognition rate across these splits is reported here. For HMDB51 dataset, we use the same filters which were generated for UCF50 dataset. The recognition rates at different compression ratios, and mean time taken for one clip (in parentheses)

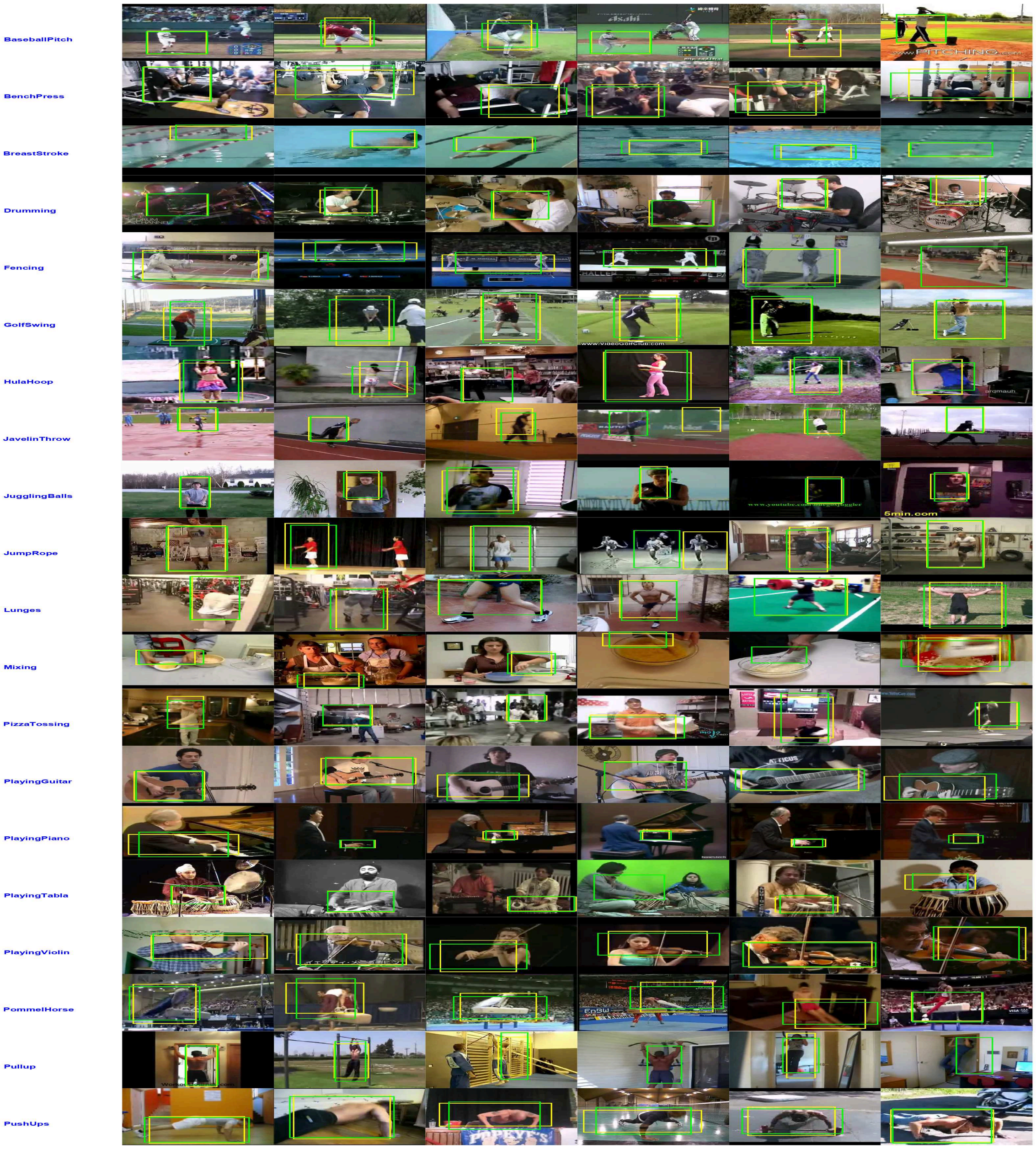


Figure 2.7: Action localization: Each row corresponds to various instances of a particular action, and action localization in one frame for each of these instances is shown. The bounding boxes (yellow for Oracle MACH, and green for STSF at compression ratio = 100) in most cases correspond to the human, or the moving part. Note that these bounding boxes shown are obtained using a rudimentary procedure, without any training, as outlined earlier in the section. This suggests that joint training of features extracted from correlation volumes and annotated bounding boxes can lead to more accurate action localization results.

Method	CR = 1	CR = 100	CR =400
Our method (‘Type 1’ + ‘Type 2’)	60.86 (2300s) (OM)	54.55 (2250s)	46.48 (2300s)
Recon + IDT	91.2 (FBI)	21.72 (3600s)	12.52 (4000s)
Action Bank [78]	57.9 (FBI)	NA	NA
Jain *et al.*[39]	59.81 (FBI)	NA	NA
Kliper-Gross *et al.*[52]	72.7 (FBI)	NA	NA
Reddy *et al.*[76]	76.9 (FBI)	NA	NA
Shi *et al.*[83]	83.3 (FBI)	NA	NA

Table 2.4: UCF50 dataset: The recognition rate for our framework is stable even at very high compression ratios, while in the case of Recon + IDT, recognition rates are much lower. The mean time per clip (given in parentheses) for our method is less than that for the baseline method (Recon + IDT).

Action	CR =1 (OM)	CR = 100	CR = 400	Action	CR =1 (OM)	CR = 100	CR = 400	Action	CR =1 (OM)	CR = 100	CR = 400	Action	CR =1 (OM)	CR = 100	CR = 400
BaseballPitch	58.67	57.05	50.335	HorseRiding	77.16	60.4	60.4	PlayingPiano	65.71	60.95	58.1	Skiing	35.42	34.72	29.86
Basketball	41.61	38.2353	25.7353	HulaLoop	55.2	56	55.2	PlayingTabla	73.88	56.75	36.94	Skijet	44	37	29
BenchPress	80	73.75	65.63	Javelin Throw	41.0256	41.0256	32.48	PlayingViolin	59	52	43	SoccerJuggling	42.31	31.61	28.38
Biking	60	42.07	33.01	Juggling Balls	64.75	67.21	65.57	PoleVault	56.25	58.12	53.75	Swing	54.01	35.03	19.7
Billiards	94.67	89.33	79.33	JumpRope	71.53	75	74.31	PommelHorse	86.07	81.3	69.1	TaiChi	66	68	61
Breaststroke	81.19	46.53	17.82	JumpingJack	80.49	80.49	72.357	PullUp	64	59	49	TennisSwing	46.11	41.92	30.53
CleanAndJerk	56.25	59.82	41.96	Kayaking	58.6	47.14	43.12	Punch	80.63	73.12	62.5	ThrowDiscus	62.6	51.14	45
Diving	76.47	71.24	51.63	Lunges	44.68	36.17	32.62	PushUps	66.67	60.78	61.76	TrampolineJumping	45.39	28.57	18.48
Drumming	63.35	50.93	44.1	MilitaryParade	80.32	78.74	59.05	RockClimbing	65.28	58.33	63.2	VolleyBall	60.34	48.27	39.65
Fencing	71.171	64.86	62.16	Mixing	51.77	56.02	48.93	RopeClimbing	36.92	34.61	29.23	WalkingwithDog	31.71	27.64	25.4
GolfSwing	71.13	58.86	48.93	Nunchucks	40.9	34.1	31.82	Rowing	55.47	40.14	29.2	YoYo	54.69	58.59	47.65
HighJump	52.03	52.84	47.15	Pizza Tossing	30.7	33.33	22.8	Salsa	69.92	63.16	46.62				
HorseRace	73.23	66.92	59.84	PlayingGuitar	73.75	64.37	60.62	SkateBoarding	55.82	46.67	38.33				

Table 2.5: UCF50 dataset: Recognition rates for individual classes at compression ratios, 1 (Oracle MACH), 100 and 400.

for our framework and Recon+IDT are tabulated in table 2.6. Table 2.6 also shows the recognition rates for various state-of-the-art action recognition approaches, while operating on full-blown images. The table clearly suggests that while operating at compression ratios of 100 and above, to perform action recognition, it is better to work in compressed domain rather than reconstructing the frames, and then applying a state-of-the-art method. While the recognition rates obtained using our method at different compression ratios are lower than state-of-the-art methods, they are very much comparable with Action Bank [78]. Action Bank method is the only filter based approach compared with in table 2.6, where linear features are extracted like in our method, whereas in the other methods highly non-linear features were extracted, which boosted action recognition accuracy substantially. The above mentioned trend can also be seen in the case of UCF50 dataset in table 2.4. This greatly underlines the limitations of linear features and the need to devise methods to extract non-linear features from CS videos.

Method	CR = 1	CR = 100	CR =400
Our method ('Type 1' + 'Type 2')	22.5 (2200s) (OM)	21.125 (2250s)	17.02 (2300s)
Recon + IDT	57.2 (FBI)	6.23 (3500s)	2.33 (4000s)
Action Bank [78]	26.9 (FBI)	NA	NA
Jain *et al.*[40]	52.1 (FBI)	NA	NA
Kliper-Gross *et al.*[52]	29.2 (FBI)	NA	NA
Jiang *et al.*[42]	40.7 (FBI)	NA	NA

Table 2.6: HMDB51 dataset: The recognition rate for our framework is stable even at very high compression ratios, while in the case of Recon+IDT, it is much lower.

2.4.5 Comments on computational complexity and storage

From tables 2.2, 2.4 and 2.6, it is evident that time taken for our framework is substantially less than that for Recon+IDT. In the case of Recon+IDT, the computational bottleneck stems from the reconstruction of the frames. Further, we note that for most frames, the reconstruction algorithm did not converge, owing to the high compression ratio. To avoid this, we ran the reconstruction algorithm for a fixed number of iterations. We also compared the storage and communication requirements of full blown videos and their compressed counterparts. It was observed that the raw data of a full blown video of size $240 \times 320 \times 106$ occupies 64873 KB, whereas the CS video at CR = 100 occupies 589 KB, leading to memory savings of 99.1%. Similarly, the CS video at CR = 400 occupies 147 KB, leading to memory savings of 99.77%.

Chapter 3

RECONNET: NON-ITERATIVE RECONSTRUCTION OF IMAGES FROM COMPRESSIVELY SENSED MEASUREMENTS

The easy availability of vast amounts of image data and the ever increasing computational power has triggered the resurgence of convolutional neural networks (CNNs) in the past three years and consolidated their position as one of the most powerful machineries in computer vision. Researchers have shown CNNs to break records in the two broad categories of long-standing vision tasks, namely: 1) high-level inference tasks such as image classification , object detection, scene recognition , fine-grained categorization and pose estimation [53, 32, 105, 103, 104] and 2) pixel-wise output tasks like semantic segmentation, depth mapping, surface normal estimation, image super resolution and dense optical flow estimation [60, 27, 96, 21, 93]. However, the benefits of CNNs have not been explored for one such important task belonging to the latter category, namely reconstruction of images from compressively sensed measurements. In this work we adapt CNNs to develop an algorithm to recover images from block CS measurements.

Motivation: Over the past decade, a plethora of reconstruction algorithms [9, 25, 72, 5, 57, 50, 100, 85, 66, 22] have been proposed. However, almost all of them are plagued by a number of similar drawbacks. Firstly, current approaches solve an optimization problem to reconstruct the images from the CS measurements. Very often, the iterative nature of the solutions to the optimization problems renders the algorithms computationally expensive with some of them even taking as many as 20 minutes to recover just one image, thus making real-time reconstruction impossible. Secondly, in many resource-constrained applications, one may be interested only in some property of the scene like ‘Where is a particular object in the image?’ or ‘What is the person in the image doing?’, rather than the exact values of all pixels in the image. In such scenarios, there is a great urge to acquire as few measurements as possible, and still be able to recover an image which retains enough information regarding the property of the scene that one is interested in. The current approaches, although slow, are capable of delivering high quality reconstructions at high measurement rates. However, their performance degrades appreciably as measurement rate decreases, yielding reconstructions which are not useful for any image understanding task. Motivated by these, in this chapter we present a CS image recovery algorithm which has the desired features of being computationally light as well as being capable of delivering reasonable quality reconstructions useful for image understanding tasks, even at extremely low measurement rates of 0.01.

Background: As stated earlier in this dissertation, compressive Sensing (CS) is a signal acquisition paradigm which provides the ability to sample a signal at sub-Nyquist rates. Unlike traditional sensing methods, in CS, one acquires a small number of random linear measurements, instead of sensing the entire signal, and a reconstruction algorithm is used to recover the original signal from the measurements. Mathematically, the measurements are given by $\mathbf{y} = \Phi\mathbf{x} + \mathbf{e}$, where $\mathbf{x} \in \mathbb{R}^n$ is the signal, $\mathbf{y} \in \mathbb{R}^m$, known as the measurement vector, denotes the set of sensed projections, $\Phi \in \mathbb{R}^{m \times n}$ is called the measurement matrix defined by a set of random patterns, and $\mathbf{e} \in \mathbb{R}^m$ is the measurement noise. Reconstructing $\mathbf{x}$ from $\mathbf{y}$ when $m < n$ is an ill-posed problem. However, CS theory [23, 11] states that the signal $\mathbf{x}$ can be recovered perfectly from a small number of $m = \mathcal{O}(s \log(\frac{n}{s}))$ random linear measurements by solving the optimization problem in Eq. 3.1, provided the signal is s-sparse in some sparsifying domain, Ψ.

$$\min_{\mathbf{x}} \quad ||\mathbf{\Psi x}||_1 \qquad s.t \qquad ||\mathbf{y} - \mathbf{\Phi x}||_2 \leq \epsilon. \tag{3.1}$$

Variants of the optimization problem with relaxed sparsity assumption in Eq. 3.1 have been proposed for the compressible signals as well. However, all such algorithms suffer from drawbacks as already discussed.

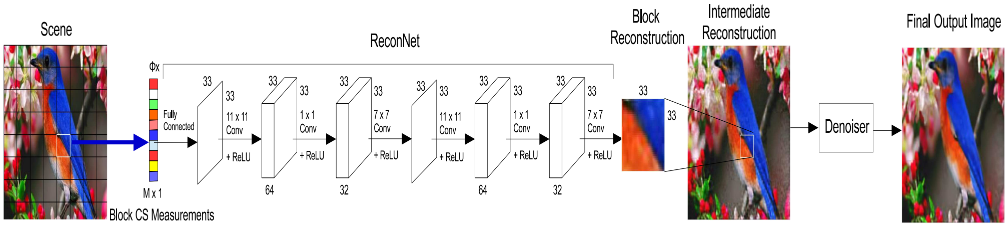


Figure 3.1: Overview of our non-iterative block CS image recovery algorithm.

3.1 Related Work

We can divide related work into two broad categories, namely CS image reconstruction algorithms and CNNs for per-pixel output tasks.

CS image reconstruction: Several algorithms have been proposed to reconstruct images from CS measurements. The earliest algorithms leveraged traditional CS theory described above [23, 11, 9] and solved the l_1-minimization in Eq. 3.1 with the assumption that the image is sparse in some transform-domain like wavelet, DCT, or gradient. However, such sparsity-based algorithms did not work well, since images, though compressible, are not exactly sparse in the transform domain. This heralded an era of model-based CS recovery methods, wherein more complex image models that go beyond simple sparsity were proposed. Model-based CS recovery methods come in two flavors. In the first, the image model is enforced explicitly [25, 5, 50, 85], where in each iteration the image estimate is projected onto the solution set defined by the model. These models, often considered under the class of 'structured-sparsity' models, are capable of capturing the higher order dependencies between the wavelet coefficients. However, generally a computationally expensive optimization is solved to obtain the projection. In the second, the algorithms enforce the image model implicitly through a non-local regularization term in the objective function [72, 100, 22]. Recently, a new class of recovery methods called approximate message passing (AMP) algorithms [24, 87, 66] have been proposed in which the image estimate is refined in each iteration using an off-the-shelf denoiser. To the best of our knowledge there exists no published work which proposes a non-iterative solution to the CS image recovery problem. However, there has been one concurrent and independent investigation ([67]) that presents stacked denoising auto-encoders (SDAs) based non-iterative approach for this problem. Different from this, in this chapter we present a convolutional architecture, which has fewer parameters, and is more easily scalable to larger block-size at the sensing stage, and also results in better performance than SDAs.

CNNs for per-pixel prediction tasks: Computer vision researchers have applied CNNs to per-pixel output tasks like semantic segmentation [60], depth estimation [27], surface normal estimation [96], image super-resolution [21] and dense optical flow estimation from a single image[93]. However, these tasks differ fundamentally from the one tackled in this dissertation in that they map a full-blown image to a similar-sized feature output, while in the CS reconstruction problem, one is required to map a small number of random linear measurements of an image to its estimate. Hence, we cannot use any of the standard CNN architectures that have been proposed so far. Motivated by this, we introduce a novel class of CNN architectures for the CS recovery problem at any arbitrary measurement rate.

3.2 Overview of Our Algorithm

Unlike most computer vision tasks like recognition or segmentation to which CNNs have been successfully applied, in the CS recovery problem, the images are not inputs but rather outputs or labels which we seek to obtain from the networks. Hence, the typical CNN architectures which can map images to rich hierarchical visual features are not applicable to our problem of interest. How does one design a network architecture for the CS recovery problem? To answer this question, one can seek inspiration from the CNN-based approach for image super-resolution proposed in [21]. Similar to the character of our problem, the outputs in image super-resolution are images, and the inputs – lower-resolution images – are of lower dimension. In [21], initial estimates of the high-resolution images are first obtained from low-resolution input images using bicubic interpolation, and then a 3-layered CNN is trained with the initial estimates as inputs and the ground-truth of the desired outputs as labels. If we were to adapt the same architecture for the CS recovery problem, we will have to first generate the initial estimates of the reconstructions from CS measurements. A straightforward option would be to run one of the several existing CS recovery algorithms and obtain initial estimates. But how many iterations do we need to run to ensure a good initial estimate? Running for too many increases computational load, defeating the very goal of this work of developing a fast algorithm, but running for too few could lead to extremely poor estimates.

Due to the aforementioned reasons, we relinquish the idea of obtaining initial estimates of the reconstructions, and instead propose a novel class of CNN architectures called ReconNet which can directly map CS measurements to image blocks. The overview of our ReconNet driven algorithm is given in Figure 3.1. The scene is divided into **non-overlapping** blocks. Each block is reconstructed by feeding in the corresponding CS measurements to 'ReconNet'. The reconstructed blocks are arranged appropriately to form an intermediate reconstruction of the image, which is input to an off-the-shelf denoiser to remove blocky artifacts and obtain the final output image.

Network architecture: Here, we describe the proposed CNN architecture, 'ReconNet' shown as part of Figure 3.1. The input to the network is an m-dimensional vector of compressive measurements, denoted by $\Phi\mathbf{x}$, where Φ is the measurement operator of size $m \times n$, m is the number of measurements and $\mathbf{x}$ is the vectorized input image block. In our case, we train networks capable of reconstructing blocks of size 33×33, hence $n = 1089$. This block size is chosen so as to reduce the network complexity and hence, the training time, while ensuring a good reconstruction quality.

The first layer is a fully connected layer that takes compressive measurements as input and outputs a feature map of size 33×33. The subsequent layers are all convolutional layers inspired by [21]. Except the final convolutional layers, all the other layers use ReLU following convolution. All feature maps produced by all convolutional layers are of size 33×33, which is equal to the block size. The first and the fourth convolutional layers use kernels of size 11×11 and generate 64 feature maps each. The second and the fifth convolutional layers use kernels of size 1×1 and generate 32 feature maps each. The third and the last convolutional layer use a 7×7 kernel and generate a single feature map, which, in the case of the last layer, is also the output of the network. We use appropriate zero padding to keep the feature map size constant in all layers.

Denoising the intermediate reconstruction: The intermediate reconstruction (see Figure 3.1) is denoised to remove the artifacts resulting due to block-wise processing. We choose BM3D [16] as the denoiser since it gives a good trade-off between computational complexity and reconstruction quality.

3.3 Learning the ReconNet

In this section, we discuss in detail training of deep networks for reconstruction of CS measurements. We use the network architecture shown in Figure 3.1 for all the cases.

Ground truth for training: We use the same set of 91 images as in [21] and can be downloaded from their website [1] . We uniformly extract patches of size 33 × 33 from these images with a stride equal to 14 to form a set of 21760 patches. We retain only the luminance component of the extracted patches (During test time, for RGB images, we use the same network to recover the individual channels). These form the labels of our training set. We obtain the corresponding CS measurements of the patches. These form the inputs of our training set. Experiments indicate that this training set is sufficient to obtain very competitive results compared to existing CS reconstruction algorithms, especially at low measurement rates.

Input data for training: To train our networks, we need CS measurements corresponding to each of the extracted patches. To this end, we simulate noiseless CS as follows. For a given measurement rate, we construct a measurement matrix, Φ by first generating a random Gaussian matrix of appropriate size, followed by orthonormalizing its rows. Then, we apply $\mathbf{y} = \Phi\mathbf{x}$ to obtain the set of CS measurements, where $\mathbf{x}$ is the vectorized version of the luminance component of an image patch. Thus, an

[1] `http://mmlab.ie.cuhk.edu.hk/projects/SRCNN/SRCNN_train.zip`

input-label pair in the training set can be represented as $(\Phi\mathbf{x}, \mathbf{x})$. We train networks for four different measurement rates (MR) $= 0.25, 0.10, 0.04$ and 0.01. Since, the total number of pixels per block is $n = 1089$, the number of measurements $n = 272, 109, 43$ and 10 respectively.

Learning algorithm details: All the networks are trained using Caffe [41]. The loss function is the average reconstruction error over all the training image blocks, given by $L(\{W\}) = \frac{1}{T}\sum_i^T ||f(\mathbf{y_i}, \{W\}) - x_i||^2$, and is minimized by adjusting the weights and biases in the network, $\{W\}$ using backpropagation. T is the total number of image blocks in the training set, x_i is the i^{th} patch and $f(\mathbf{y_i}, \{W\})$ is the network output for i^{th} patch. For gradient descent, we set the batch size to 128 for all the networks. For each measurement rate, we train two networks, one with random Gaussian initialization for the fully connected layer, and one with a deterministic initialization, and choose the network which provides the lower loss on a validation test. For the latter network, the j^{th} weight connecting the i^{th} neuron of the fully connected layer is initialized to be equal to $\Phi^T_{i,j}$. In each case, weights of all convolutional layers are initialized using a random Gaussian with a fixed standard deviation. The learning rate is determined separately for each network using a linear search. All networks are trained on a Nvidia Tesla K40 GPU for about a day each.

3.4 Experimental Results

In this section, we conduct extensive experiments on both simulated data and real data, and compare the performance of our CS recovery algorithm with state-of-the-art CS image recovery algorithms, both in terms of reconstruction quality and time complexity.

Image Name	Algorithm	MR = 0.25		MR = 0.10		MR = 0.04		MR = 0.01	
		w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D	w/o BM3D	w/ BM3D
Monarch	TVAL3 [57]	**27.77**	**27.77**	**21.16**	21.16	16.73	16.73	11.09	11.11
	NLR-CS [22]	25.91	26.06	14.59	14.67	11.62	11.97	6.38	6.71
	D-AMP [66]	26.39	26.55	19.00	19.00	14.57	14.57	6.20	6.20
	SDA [67]	23.54	23.32	20.95	21.04	18.09	18.19	15.31	15.38
	ReconNet (Ours)	24.31	25.06	21.10	**21.51**	**18.19**	**18.32**	**15.39**	**15.49**
Parrot	TVAL3	**27.17**	**27.24**	23.13	**23.16**	18.88	18.90	11.44	11.46
	NLR-CS	26.53	26.72	14.14	14.16	10.59	10.92	5.11	5.44
	D-AMP	26.86	26.99	21.64	21.64	15.78	15.78	5.09	5.09
	SDA	24.48	24.36	22.13	22.35	**20.37**	20.67	**17.70**	17.88
	ReconNet (Ours)	25.59	26.22	22.63	**23.23**	20.27	**21.06**	17.63	**18.30**
Barbara	TVAL3 [57]	24.19	24.20	21.88	22.21	18.98	18.98	11.94	11.96
	NLR-CS [22]	**28.01**	**28.00**	14.80	14.84	11.08	11.56	5.50	5.86
	D-AMP [66]	25.89	25.96	21.23	21.23	16.37	16.37	5.48	5.48
	SDA [67]	23.19	23.20	22.07	22.39	**20.49**	20.86	18.59	18.76
	Ours	23.25	23.52	**21.89**	**22.50**	20.38	**21.02**	**18.61**	**19.08**
Boats	TVAL3	28.81	28.81	23.86	23.86	19.20	19.20	11.86	11.88
	NLR-CS	29.11	**29.27**	14.82	14.86	10.76	11.21	5.38	5.72
	D-AMP	**29.26**	29.26	21.95	21.95	16.01	16.01	5.34	5.34
	SDA	26.56	26.25	24.03	**24.18**	21.29	21.54	**18.54**	18.68
	ReconNet (Ours)	27.30	27.35	**24.15**	24.10	**21.36**	**21.62**	18.49	**18.83**
Cameraman	TVAL3	**25.69**	**25.70**	**21.91**	**21.92**	18.30	18.33	11.97	12.00
	NLR-CS	24.88	24.96	14.18	14.22	11.04	11.43	5.98	6.31
	D-AMP	24.41	24.54	20.35	20.35	15.11	15.11	5.64	5.64
	SDA	22.77	22.64	21.15	21.30	**19.32**	19.55	17.06	17.19
	ReconNet (Ours)	23.15	23.59	21.28	21.66	19.26	**19.72**	**17.11**	**17.49**
Fingerprint	TVAL3	22.70	22.71	18.69	18.70	16.04	16.05	10.35	10.37
	NLR-CS	23.52	23.52	12.81	12.83	9.66	10.10	4.85	5.18
	D-AMP	25.17	23.87	17.15	16.88	13.82	14.00	4.66	4.73
	SDA	24.28	23.45	20.29	20.31	16.87	16.83	**14.83**	14.82
	Ours	**25.57**	**25.13**	**20.75**	**20.97**	**16.91**	**16.96**	14.82	**14.88**
Flintstones	TVAL3	24.05	24.07	18.88	18.92	14.88	14.91	9.75	9.77
	NLR-CS	22.43	22.56	12.18	12.21	8.96	9.29	4.45	4.77
	D-AMP	**25.02**	**24.45**	16.94	16.82	12.93	13.09	4.33	4.34
	SDA	20.88	20.21	18.40	18.21	16.19	16.18	13.90	13.95
	Ours	22.45	22.59	**18.92**	**19.18**	**16.30**	**16.56**	**13.96**	**14.08**
Foreman	TVAL3	35.42	35.54	**28.69**	**28.74**	20.63	20.65	10.97	11.01
	NLR-CS	**35.73**	**35.90**	13.54	13.56	9.06	9.44	3.91	4.25
	D-AMP	35.45	34.04	25.51	25.58	16.27	16.78	3.84	3.83
	SDA	28.39	28.89	26.43	27.16	23.62	24.09	**20.07**	20.23
	ReconNet (Ours)	29.47	30.78	27.09	28.59	**23.72**	**24.60**	20.04	**20.33**
House	TVAL3	32.08	32.13	26.29	26.32	20.94	20.96	11.86	11.90
	NLR-CS	**34.19**	**34.19**	14.77	14.80	10.66	11.09	4.96	5.29
	D-AMP	33.64	32.68	24.84	24.71	16.91	17.37	5.00	5.02
	SDA	27.65	27.86	25.40	26.07	22.51	22.94	**19.45**	**19.59**
	ReconNet (Ours)	28.46	29.19	**26.69**	**26.66**	**22.58**	**23.18**	19.31	19.52
Lena	TVAL3	28.67	28.71	**24.16**	24.18	19.46	19.47	11.87	11.89
	NLR-CS	**29.39**	**29.67**	15.30	15.33	11.61	11.99	5.95	6.27
	D-AMP	28.00	27.41	22.51	22.47	16.52	16.86	5.73	5.96
	SDA	25.89	25.70	23.81	24.15	21.18	21.55	17.84	17.95
	Ours	26.54	26.53	23.83	**24.47**	**21.28**	**21.82**	**17.87**	**18.05**
Peppers	TVAL3	29.62	29.65	**22.64**	22.65	18.21	18.22	11.35	11.36
	NLR-CS	28.89	29.25	14.93	14.99	11.39	11.80	5.77	6.10
	D-AMP	**29.84**	**28.58**	21.39	21.37	16.13	16.46	5.79	5.85
	SDA	24.30	24.22	22.09	22.34	**19.63**	19.89	**16.93**	**17.02**
	ReconNet (Ours)	24.77	25.16	22.15	**22.67**	19.56	**20.00**	16.82	16.96
Mean PSNR	TVAL3	27.84	27.87	**22.84**	22.86	18.39	18.40	11.31	11.34
	NLR-CS	28.05	**28.19**	14.19	14.22	10.58	10.98	5.30	5.62
	D-AMP	**28.17**	27.67	21.14	21.09	15.49	15.67	5.19	5.23
	SDA	24.72	24.55	22.43	22.68	19.96	20.21	**17.29**	17.40
	Ours	25.54	25.92	22.68	**23.23**	**19.99**	**20.44**	17.27	**17.55**

Table 3.1: PSNR values in dB of the test images using different algorithms at different measurement rates. At low measurement rates of 0.1, 0.04 and 0.01, our algorithm yields superior quality reconstructions than the traditional iterative CS reconstruction algorithms, TVAL3, NLR-CS, and D-AMP. It is evident that the reconstructions are very stable for our algorithm with a decrease in mean PSNR of only 8.37 dB as the measurement rate decreases from 0.25 to 0.01, while the smallest corresponding dip in mean PSNR for classical reconstruction algorithms is in the case of TVAL3, which is equal to 16.53 dB.

Baselines: We compare our algorithm with three iterative CS image reconstruction algorithms, TVAL3 [57], NLR-CS [22] and D-AMP [66]. We use the code made available by the respective authors on their websites. Parameters for these algorithms, including the number of iterations, are set to the default values. We use BM3D [16] denoiser since it gives a good trade-off between time complexity and reconstruction quality. The code for NLR-CS provided on author's website is implemented only for random Fourier sampling. The algorithm first computes an initial estimate using a DCT or wavelet based CS recovery algorithm, and then solves an optimization problem to get the final estimate. Hence, obtaining a good estimate is critical to the success of the algorithm. However, using the code provided on the author's website, we failed to initialize the reconstruction for random Gaussian measurement matrix. Similar observation was reported by [66]. Following the procedure outlined in [66], the initial image estimate for NLR-CS is obtained by running D-AMP (with BM3D denoiser) for 8 iterations. Once the initial estimate is obtained, we use the default parameters and obtain the final NLR-CS reconstruction. We also compare with a concurrent work [67] which presents an SDA based non-iterative approach to recover from block-wise CS measurements. Here, we compare our algorithm with our own implementation of SDA, and show that our algorithm outperforms the SDA. For fair comparison, we denoise the image estimates recovered by baselines as well. The only parameter to be input to the BM3D algorithm is the estimate of the standard Gaussian noise, σ. To estimate σ, we first compute the estimates of the standard Gaussian noise for each block in the intermediate reconstruction given by $\sigma_i = \sqrt{\frac{||y_i - \Phi x_i||^2}{m}}$, and then take the median of these estimates.

3.4.1 Simulated data

For our simulated experiments, we use a standard set of 11 grayscale images, compiled from two sources [2] , [3] . We conduct both noiseless and noisy block-CS image reconstruction experiments at four different measurement rates 0.25, 0.1, 0.04 and 0.01.

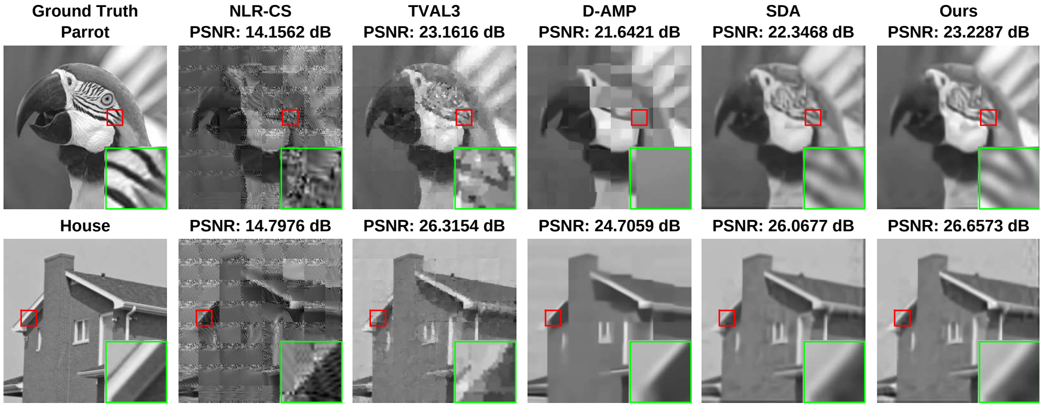


Figure 3.2: Reconstruction results for parrot and house images from noiseless CS measurements at measurement rate of 0.1. It is evident that our algorithm recovers more visually appealing images than other competitors. Notice how fine structures are recovered by our algorithm.

Reconstruction from noiseless CS measurements: To simulate noiseless block-wise CS, we first divide the image of interest into non-overlapping blocks of size 33×33, and then compute CS measurements for each block using the same random Gaussian measurement matrix as was used to generate the training data for the network corresponding to the measurement rate. The PSNR values in dB for both intermediate reconstruction (indicated by w/o BM3D) as well as final denoised versions (indicated by w/ BM3D) for the measurement rates are presented in Table 3.1. It is clear

[2] `http://dsp.rice.edu/software/DAMP-toolbox`

[3] `http://see.xidian.edu.cn/faculty/wsdong/NLR_Exps.htm`

from the PSNR values that our algorithm outperforms traditional reconstruction algorithms at low measurement rates of $0.1, 0.04$ and 0.01. Also, the degradation in performance with lower measurement rates is more graceful.

Further, in Figure 3.2, we show the final reconstructions of parrot and house images for various algorithms at measurement rate of 0.1. From the reconstructed images, one can notice that our algorithm, as well as SDA are able to retain the finer features of the images while other algorithms fail to do so. NLR-CS and DAMP provide poor quality reconstruction. Even though TVAL3 yields PSNR values comparable to our algorithm, it introduces undesirable artifacts in the reconstructions.

Algorithm	MR = 0.25	MR = 0.10	MR = 0.04	MR = 0.01
TVAL3	2.943	3.223	3.467	7.790
NLR-CS	314.852	305.703	300.666	314.176
D-AMP	27.764	31.849	34.207	54.643
ReconNet	0.0213	0.0195	0.0192	0.0244
SDA	0.0042	0.0029	0.0025	0.0045

Table 3.2: Time complexity (in seconds) of various algorithms (without BM3D) for reconstructing a single 256 × 256 image. By taking only about 0.02 seconds at any given measurement rate, ReconNet can recover images from CS measurements in real-time, and is 3 orders of magnitude faster than traditional reconstruction algorithms.

Time complexity: In addition to competitive reconstruction quality, for our algorithm without the BM3D denoiser, the computation is real-time and is about **3** orders of magnitude faster than traditional reconstruction algorithms. To this end, we compare various algorithms in terms of the time taken to produce the intermediate reconstruction of a 256×256 image from noiseless CS measurements at various measurement rates. For traditional CS algorithms, we use an Intel Xeon E5-1650 CPU to run the implementations provided by the respective authors. For ReconNet and SDA, we use a Nvidia GTX 980 GPU to compute the reconstructions. The average time taken for the all algorithms of interest are given in table 3.2. Depending on the measurement rate, the time taken for block-wise reconstruction of a 256 × 256 for our algorithm is about 145 to 390 times faster than TVAL3, 1400 to 2700 times faster than D-AMP, and 15000 times faster than NLR-CS. It is important to note that the speedup achieved by our algorithm is not solely because of the utilization of the GPU. It is mainly because unlike traditional CS algorithms, our algorithm being CNN based relies on much simpler convolution operations, for which very fast implementations exist. More importantly, the non-iterative nature of our algorithm makes it amenable to parallelization. SDA, also a deep-learning based non-iterative algorithm shows significant speedups over traditional algorithms at all measurement rates.

Performance in the presence of noise: To demonstrate the robustness of our algorithm to noise, we conduct reconstruction experiments from noisy CS measurements. We perform this experiment at three measurement rates - $0.25, 0.10$ and 0.04. We emphasize that for ReconNet and SDA, we **do not** train separate networks for different noise levels but use the same networks as used in the noiseless case. To first obtain the noisy CS measurements, we add standard random Gaussian noise of increasing standard deviation to the noiseless CS measurements of each block. In each case, we test the algorithms at three levels of noise corresponding to $\sigma = 10, 20, 30$, where σ is the standard deviation of the Gaussian noise distribution. The intermediate reconstructions are denoised using BM3D. The mean PSNR for various noise levels for different algorithms at different measurement rates are shown in Figure 3.4. It can be observed that our algorithm beats all other algorithms at high noise levels.

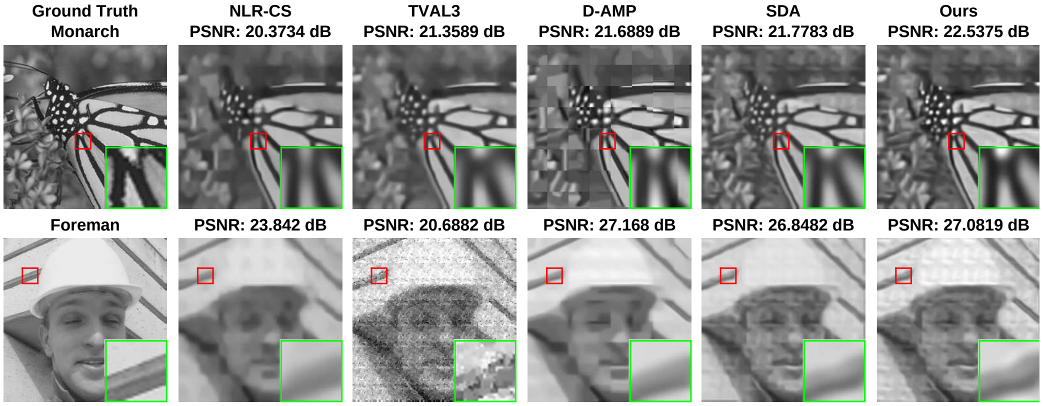


Figure 3.3: Reconstruction results for monarch and foreman images from 25% noisy CS measurements with noise standard deviation equal to 30. One can observe that our algorithm provides visually appealing reconstructions despite high noise level.

This shows that the method proposed in this work is extremely robust to all levels of noise. Further, in figure 3.3, we show the final reconstructions of monarch and foreman images for various algorithms at measurement rate of 0.25 and noise standard deviation equal to 30. From the reconstructed images, one can notice that our algorithm is extremely robust to noise and provides visually appealing reconstructions despite the very large amount of noise. On the other hand, NLR-CS and TVAL3 provide poor quality reconstruction.

3.4.2 Experiments with real data

The previous section demonstrated the superiority of our algorithm over traditional algorithms for simulated CS measurements. Here, we show that our networks trained on simulated data can be readily applied for real world scenario by reconstructing images from CS measurements obtained from our block SPC. We compare our reconstruction results with other algorithms.

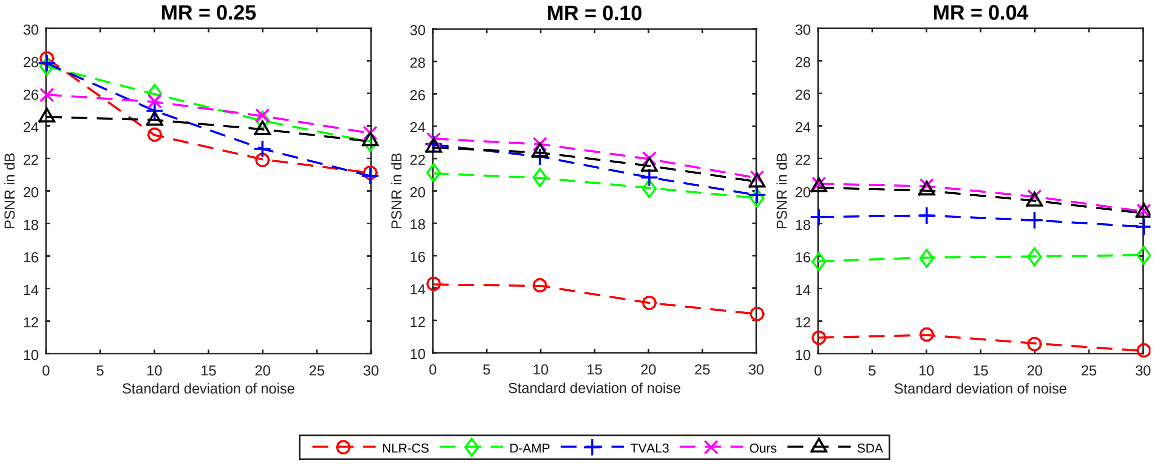


Figure 3.4: Comparison of different algorithms in terms of mean PSNR (in dB) for the test set in presence of Gaussian noise of different standard deviations at MR = 0.25, 0.10 and 0.04.

Scalable Optical Compressive Imager Testbed: We implement a scalable optical compressive imager testbed similar to the one described in [48, 47]. It consists of two optical arms and a discrete micro-mirror device (DMD) acting as a spatial light modulator as shown in Figure 3.5. The first arm, akin to an imaging lens in a traditional system, forms an optical image of the scene in the DMD plane. It has a 40° field of view and operates at F/8. The DMD has a resolution of 1920 × 1080 micro-mirror elements, each of size $10.8\mu m$. However, in our system the field of view (FoV) is limited to an image circle of 7.5mm, which is approximately 700 DMD pixels. The DMD micro-mirrors are bi-stable and each is either oriented half-way toward the second arm or in the opposite direction (when the flux is discarded). The micro-mirrors can be switched in either direction at a very high rate to effectively achieve 8 bits gray-scale modulation via pulse width modulation. The optically modulated scene on the DMD plane is then imaged (by the second arm) and spatially integrated by a 1/3", 640 × 480 CCD focal plane array with a measurement depth of 12 bits.

In the CCD plane, the field of view is 3mm in diameter (≈ 400 CCD pixels). Thus, in effect, this testbed implements several single pixel cameras [86] in parallel. Each block on the DMD effectively maps to a super pixel (e.g. 2×2 binned pixels) on the CCD. The DMD sequences (in time) through m projections, implementing the m rows of the $m \times n$ projection matrix Φ, where each projection vector appears as a $\sqrt{n} \times \sqrt{n}$ block pattern, replicated across the scene FoV. Before data acquisition, a calibration step is performed to map the DMD blocks to CCD detector pixels to characterize any deviation from the idealized system model.

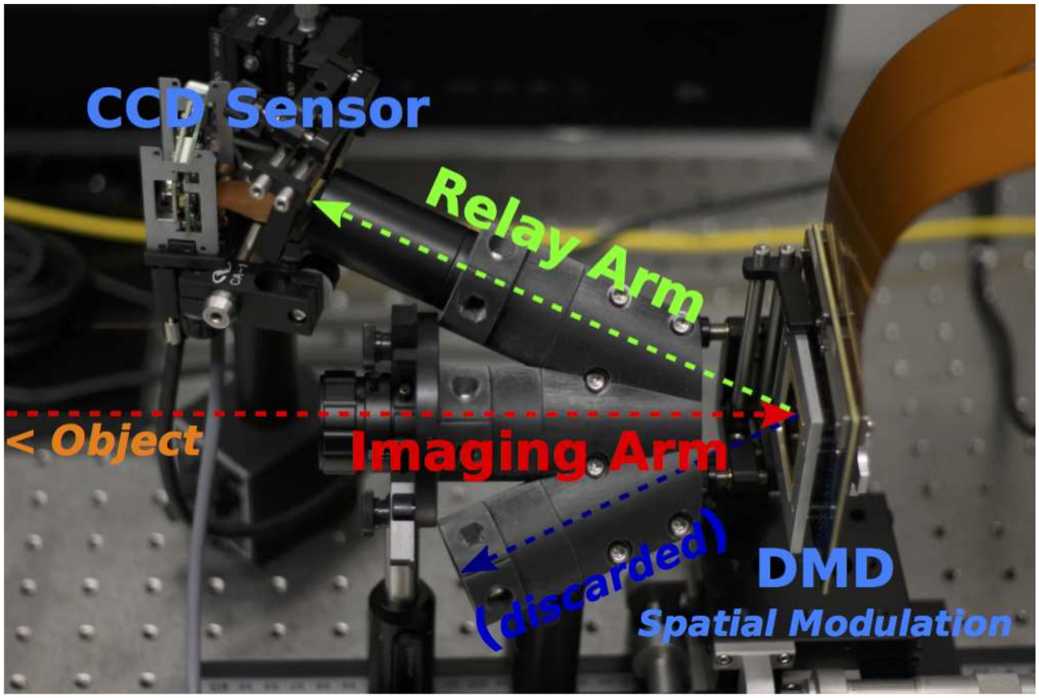


Figure 3.5: Compressive imager testbed layout with the object imaging arm in the center, the two DMD imaging arms are on the sides.

Reconstruction experiments: We use the set up described above to obtain the CS measurements for 383 blocks (size of 33×33) of the scene. Operating at MR's of 0.1 and 0.04, we implement the 8-bit quantized versions of measurement matrices

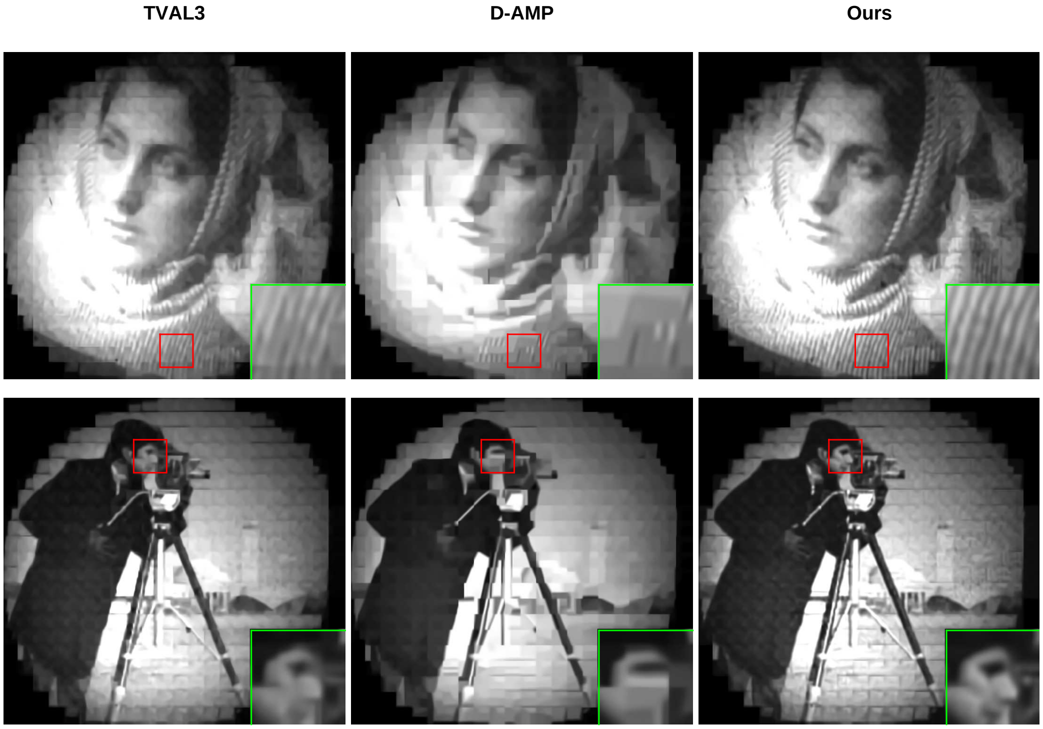


Figure 3.6: The figure shows reconstruction results on 3 images collected using our block SPC operating at measurement rate of 0.1. The reconstructions of our algorithm are qualitatively better than those of TVAL3 and D-AMP.

(orthogonalized random Gaussian matrices). The measurement vectors are input to the corresponding networks trained on the simulated CS measurements to obtain the block-wise reconstructions as before and the intermediate reconstruction is denoised using BM3D. Figures 3.6 and 3.7 show the reconstruction results using TVAL3, D-AMP and our algorithm for three test images at MR = 0.10 and 0.04 respectively. It can be observed that our algorithm yields visually good quality reconstruction and preserves more detail compared to others, thus demonstrating the robustness of our algorithm.

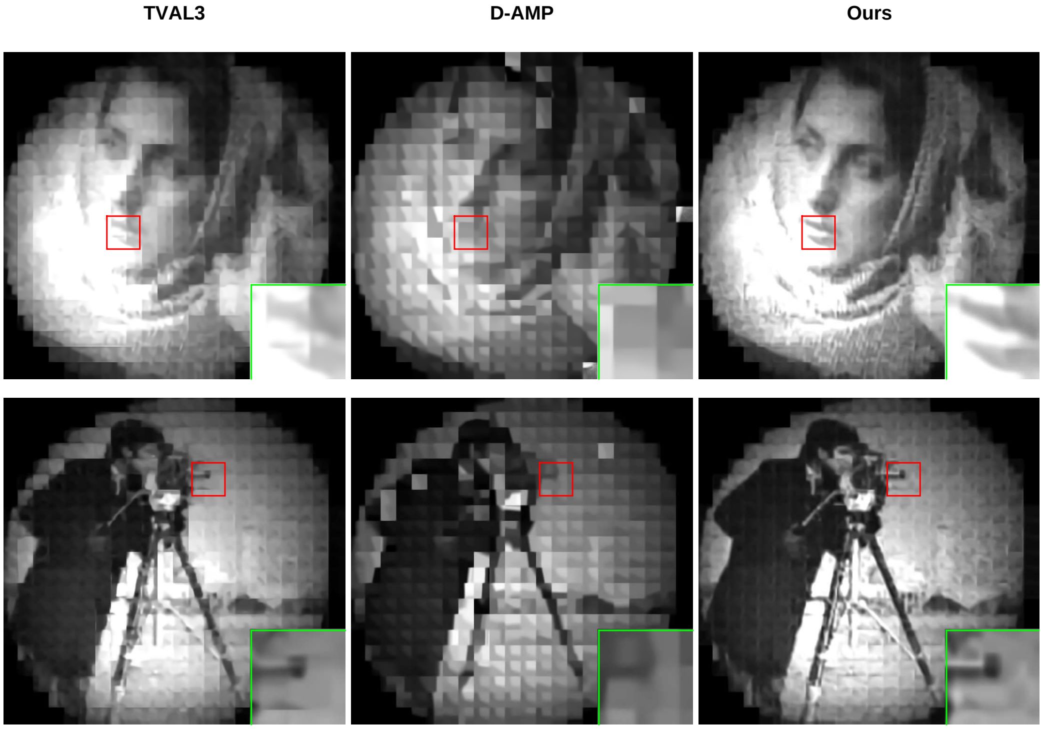


Figure 3.7: The figure shows reconstruction results on 3 images collected using our block SPC operating at measurement rate of 0.04. The reconstructions of our algorithm are qualitatively better than those of TVAL3 and D-AMP.

3.4.3 Training strategy for a different Φ

We surmise that for a new Φ of a desired measurement rate, one **does not** need to train the network from scratch, and that it may be sufficient to follow a suboptimal, yet effective and computationally light training strategy outlined below, ideally suited to practical scenarios. We adapt the convolutional layers (C1-C6) of a pre-trained network for the same or slightly higher MR, henceforth referred to as the *base network*, and train **only** the fully connected (FC) layer with random initialization for 1000 iterations (or equivalent time of around 2 seconds on a Titan X GPU), while keeping C1-C6 **fixed**. The mean PSNR (without BM3D) for the test-set at various MRs, the time taken to train models and the MR of the base network are given in table 3.3.

From the table, it is clear that the overhead in computation for new Φ is trivial, while the mean PSNR values are comparable to the ones presented in Table 3.1. We note that one can obtain better quality reconstructions at the cost of more training time if C1-C6 layers are also fine-tuned along with FC layer.

New Φ MR	**0.1**	**0.08**	**0.04**	**0.01**
Base network MR	0.25	0.1	0.1	0.25
Mean PSNR (dB)	21.73	20.99	19.66	16.60
Training Time (seconds)	2	2	2	2

Table 3.3: Networks for a new Φ can be obtained by training only the FC layer of the base network at minimal computational overhead, while maintaining comparable PSNRs.

3.5 Relation to super resolution

Although both reconstruction of CS measurements and super resolution (SR) can be cast as inversion of a undetermined linear-system $\mathbf{y} = \Phi\mathbf{x}$, in practice, they are not considered under the same umbrella for the following reason. Φ in the case of SR has a special structure with uniformly spaced 1s and 0s. This allows access to measurements of all patches (non-overlapping or otherwise), which are directly available as certain pixel values. Most techniques, including SRCNN [21], exploit this by averaging out the pixels which belong to multiple patches, thus obtaining far superior results than can be obtained if they are applied to only non-overlapping blocks. However, in the CS recovery problem, Φ is a random (Gaussian) matrix, which is devoid of a structure similar to the one, Φ in SR possesses. If we wish to process overlapping patches, then we need to **sense** each patch explicitly and obtain corresponding measurements. This would require a moving part in the optical system, such as a moving lens! These adjustments are not only difficult, but also beyond the scope of this work. Moreover, sensing overlapping patches amounts to increase in measurement rate. Hence, we consider only non-overlapping patches, which is also optically implemented in our prototype.

3.6 Real-time high level vision from CS imagers

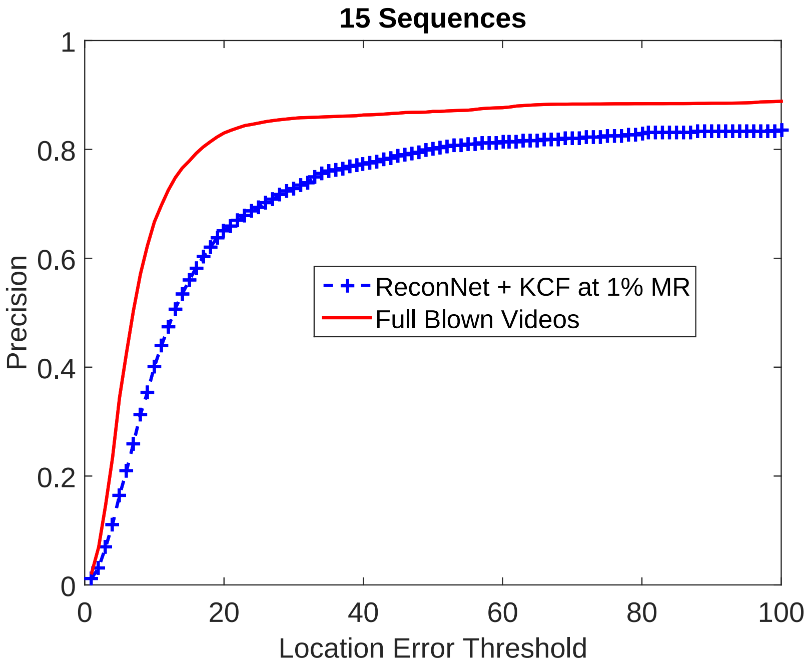


Figure 3.8: The figure shows the variation of average precision with location error threshold for ReconNet+KCF and original videos. For a location error threshold of 20 pixels, ReconNet+KCF achieves an impressive average precision of 65.02%.

In the previous section, we have shown how our approach yields good quality reconstruction results in terms of PSNR over a broad range of measurement rates. Despite the expected degradation in PSNR as the measurement rate plummets to 0.01, our algorithm still yields reconstructions of 15-20 dB PSNR and rich semantic content is still retained. As stated earlier, in many resource-constrained inference applications the goal is to acquire the least amount of data required to perform high-level image understanding. To demonstrate how CS imaging can applied in such scenarios, we present an example proof of concept real-time high level vision application - tracking. To this end we simulate video CS at a measurement rate of 0.01 by obtaining frame-wise block CS measurements on 15 publicly available videos [98] (BlurBody, BlurCar1, BlurCar2, BlurCar4, BlurFace, BlurOwl, Car2, CarDark, Dancer, Dancer2, Dudek, FaceOcc1, FaceOcc2, FleetFace, Girl2) used to benchmark tracking algorithms. Further, we perform object tracking on-the-fly as we recover the frames of the video using our algorithm without the denoiser. For object tracking we use a state-of-the-art algorithm based on kernelized correlation filters [38]. We call the aforementioned pipeline, ReconNet+KCF. For comparison, we conduct tracking on original videos as well. Figure 3.8 shows the average precision curve over the 15 videos, in which each datapoint is the mean percentage of frames that are tracked correctly for a given location error threshold. Using a location error threshold of 20 pixels, the average precision over 15 videos for ReconNet+KCF at 1% MR is 65.02%, whereas tracking on the original videos yields an average precision value of 83.01%. ReconNet+KCF operates at around 10 Frames per Second (FPS) for a video with frame size of 480×720 to as high as 56 FPS for a frame size of 240×320. This shows that even at an extremely low MR of 1%, using our algorithm, effective and real-time tracking is possible by using CS measurements. In figure 3.9 we present qualitative results for 8 of those videos by overlaying on the original frames, the bounding boxes predicted for ReconNet+KCF (in red) and original videos+KCF (in blue). It can be seen that for the videos where the target object is of reasonably large size, ReconNet+KCF performs nearly as well as original videos + KCF. This indicates that the reconstruction output by ReconNet retain enough semantic information to reliably track medium to large sized targets. However, for very small sized targets, ReconNet+KCF performs poorly indicating that at measurement rate of 0.01, the reconstructed frames do not retain fine-grained information in the image

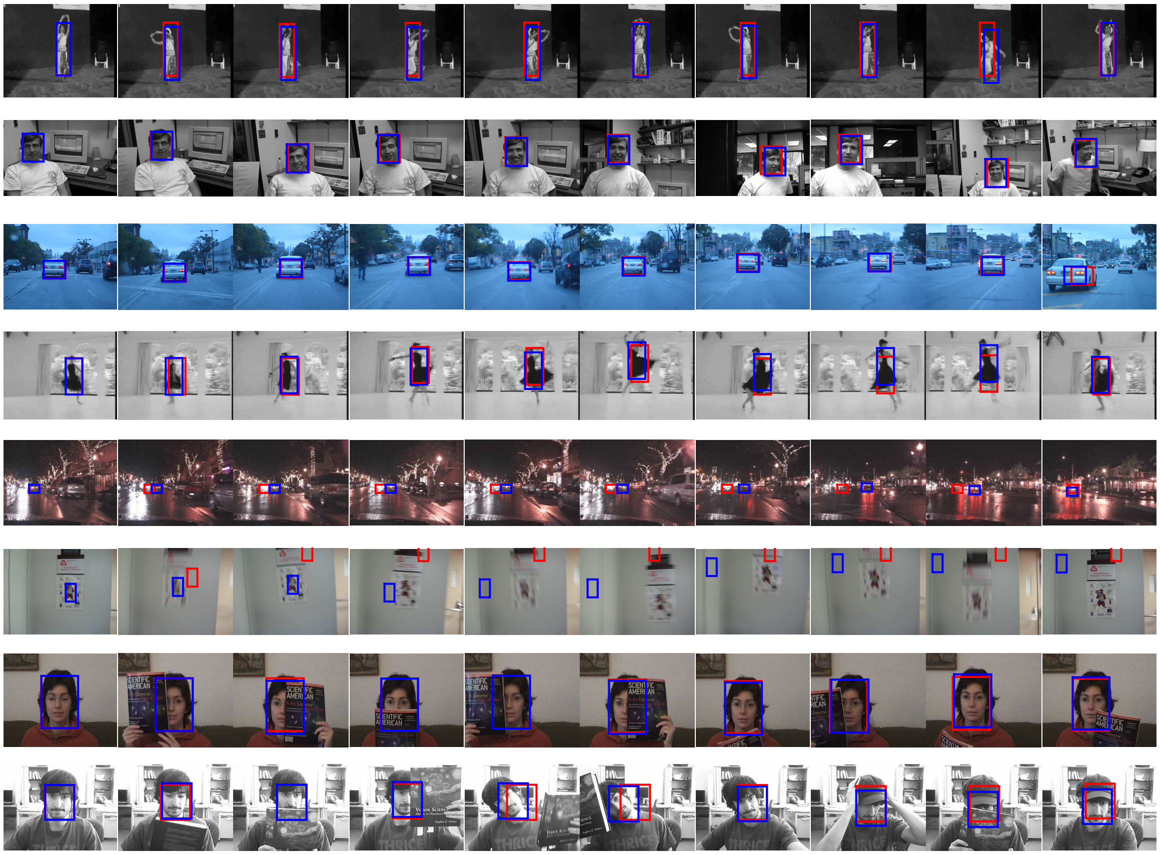


Figure 3.9: The figure shows the variation of average precision with location error threshold for ReconNet+KCF and original videos. For a location error threshold of 20 pixels, ReconNet+KCF achieves an impressive average precision of 65.02%.

Chapter 4

FAST INTEGRAL IMAGE ESTIMATION AT 1% MEASUREMENT RATE

In this chapter, we study the problem of obtaining integral image estimates from emerging flexible programmable imaging devices. These novel imaging devices, often considered under the broad umbrella of spatial-multiplexing cameras (SMCs)[68, 65, 80] provide a number of benefits in reducing the amount of sensing for portable and resource constrained acquisition. The imaging architectures in these cameras employ spatial light modulators like digital micromirror arrays to optically compute projections of the scene. Mathematically, the projections are given by $y = \phi x$, where $x \in \mathbb{R}^n$ is the image, $y \in \mathbb{R}^m$, known as the measurement vector, denotes the set of sensed projections and $\phi \in \mathbb{R}^{m \times n}$ is called measurement matrix defined by the set of multiplexing patterns. The nature of the acquisition framework enables us to deploy SMCs in resource constrained settings, wherein one can employ $m << n$ number of photon detectors to sense otherwise high-resolution imagery [65, 79] and obtain a very small number of measurements. Later, a reconstruction algorithm is used to recover the image x. However, reconstructing x from y when $m < n$ is an ill-posed problem. Researchers in the past have attempted to provide solutions by carefully designing the measurement matrix ϕ in the hope of easier recovery of x from y. Recent compressive sensing (CS) theory provides one possible solution to tackle the above mentioned ill-posed problem. According to CS theory, a signal can be recovered perfectly from a small number of $m = \mathcal{O}(s \log(\frac{n}{s}))$ such pseudo-random (PR) multiplexed measurements, where s is the sparsity of the signal. However, a significant research shows that high-quality reconstruction is computationally intensive [23, 11, 90, 69]. Hence, despite the promise of CS-based SMCs [65, 80], the computational bottleneck of non-linear, iterative reconstruction has withheld their wide-spread adoption in applications which require fast inference of objects, actions, and scenes. This has led to researchers exploring the option of tackling inference problems directly from these pseudo-random multiplexed measurements [79, 88, 45, 63, 74, 55, 59] (more on these later in the section in related work). However, the 'universal' nature of such measurements has made it challenging to devise new or adopt existing computer vision algorithms to solve the inference problem at hand.

The need to acquire as less data as possible combined with the limitations of pseudo-random multiplexers props us to outline the following goal. The goal is to propose a novel sensing framework for SMCs such that acquired measurements satisfy the following properties. 1) The measurements are not random in nature but are tailored for a particular application. 2) The number of measurements is 2 orders less than the number of pixels, so that SMCs based on our framework can be employed in resource constrained applications. 3) A simple linear operation on the measurement vector y yields a 'proxy' representation (e.g integral images, gradient images) from which the required features are extracted for the application in hand, thus avoiding the computationally expensive iterative and non-linear reconstruction.

In this chapter, we focus on one such 'proxy' representation, integral images. Integral images are extremely attractive representation since Haar-like features and box-filtered image outputs can be computed from integral images with a small and fixed number of floating point operations in constant time [91]. These advantages have led to their widespread use in real time applications like face detection [91], pedestrian detection [20], object tracking [34, 102, 4, 46] and object segmentation [75].

Instead of setting a fixed number of measurements, we formulate an optimization problem to minimize the number of measurements while incorporating the other two (1 and 3) properties of measurements (as mentioned above) in the constraints. Minimizing the number of measurements is akin to minimizing the rank of the measurement matrix. In more concrete terms, the problem is posed to jointly minimize the rank of the measurement matrix, ϕ and learn the linear operator, $\mathcal{L}$ which when applied on the measurement vector yields the approximate integral image, with the probabilistic constraint that the error between the approximate integral image and the exact integral image is within allowable limits with high probability. By controlling the allowable error limit, we can obtain measurement matrix of the desired rank. Incorporating a wavelet domain prior model for natural images combined with a relaxation (explained in section 2) allows us to convert the probabilistic constraint into a series of second conic constraints. Rank minimization is a NP-hard problem. Relaxing the objective function to nuclear norm allows to use off-the-shelf convex optimization tools to solve the problem and obtain the measurement matrix and the linear operator.

Related Work: The related previous works in literature follow one of the two themes. Some attempt to tackle inference problems directly from PR measurements without optimizing for the measurement matrix for the inference task at hand, some others attempt to optimize measurement matrix for a particular signal model so as to minimize reconstruction error.

a) Design of measurement matrix: A closely related work can be found in [26], wherein a framework is proposed to jointly optimize for a measurement matrix and an overcomplete sparsifying dictionary for small patches of images. Results suggest that better reconstruction results can be obtained using this strategy. However, learning global dictionaries for entire images is not possible, and hence the framework is not scalable. Goldstein *et al.*[33] designed measurement matrices called 'STOne' Transform which facilitate fast low resolution 'previews' just by direct reconstruction, and the hope is that 'previews' are of high enough quality so that conventional methods for inference tasks can be applied. Assuming a multi-resolutional signal model for natural images, Chang *et al.*[14] proposed an algorithm to obtain measurements which have the maximum mutual information with natural images.

b) Inference problems from CS videos: A LDS (Linear Dynamical System) based approach was proposed by Sankaranarayanan *et al.*[79] to model CS videos and recover the LDS parameters directly from PR measurements. However, the method is sensitive to spatial and view transforms. Calderbank *et al.*[74] theoretically proved that one can learn classifiers directly from PR measurements, and that with high probability the performance of the linear kernel support vector machine (SVM) classifier operating on the CS measurements is similar to that of the the best linear threshold classifier operating on the original data. A reconstruction-free framework was proposed by Thirumalai *et al.*[88] to compute optical flow based on correlation estimation between two images, directly from PR measurements. Davenport *et al.*[63] proposed a measurement domain based correlation filter approach for target classification. Here, the trained filters are first projected onto PR patterns to obtain 'smashed filters', and then the PR measurements of the test examples are correlated with these smashed filters. Recently, Kulkarni *et al.*[55] and Lohit *et al.*[59] extended the 'smashed filter' approach to action recognition and face recognition respectively, and demonstrated the feasibility and scalability of tackling difficult inference tasks in computer vision directly from PR measurements.

4.1 Background

In this section, we provide a brief background on the probability model for natural images, which we rely on, and introduce notations required to set up the optimization problem to derive measurement matrix and above referred linear operator.

Probability Model of natural images: There is a rich body of literature which deals with statistical modeling of natural images. We refer to some works which are related to the probability model we use here. Many successful probability models for wavelet coefficients fall under the broad umbrella of Gaussian scale mixtures (GSM) [3], [92]. Typically the coefficient space is partitioned into overlapping blocks, and each block is modeled independently as a GSM, which captures the local dependencies. This implicitly gives rise to a global model of the wavelet coefficients. Building on this framework, Lyu *et al.*[62] proposed a field of Gaussian scale mixtures (FoGSM) to explicitly model the subbands of wavelet coefficients, while treating each subband independently. However, incorporating such a general model for wavelet coefficient vector makes it very difficult to compute the distribution for even simple functions like a linear function of the wavelet coefficient vector. Therefore, it is vital to assume a prior model which can lead to tractable computation of the distribution. It is well-known that marginal distributions of detailed wavelet coefficients follow generalized Gaussian distribution [64]. We extend this notion to multi-dimensions and model the vector of detailed wavelet coefficients by multivariate generalized Gaussian distribution (MGGD).

To put it formally, let $\mathbf{U}^T \in \mathbb{R}^{n \times n}$ be the orthogonal matrix representing the $\log_2(n)$ level wavelet transform, so that $x = \mathbf{U}w$, where $w \in \mathbb{R}^n$ is the corresponding wavelet coefficient vector. Without loss of generality, we assume that all entries in the first row of $\mathbf{U}^T$ are $1/\sqrt{n}$ so that the first entry in w corresponds to $\sqrt{n}$ times the mean of all entries in x. Also we denote the rest $n-1$ rows in $\mathbf{U}^T$ by $\mathbf{U}^T_{2:n}$. Now we can write $w = [\sqrt{n}\bar{x}, w_d]$, where $\bar{x}$ is the mean of x and w_d is the vector of detailed coefficients. As explained above, the probability distribution of w_d (MGGD) is given by

$$f(w) = K|\mathbf{\Sigma}_{w_d}|^{-0.5} exp(-(w_d^T \mathbf{\Sigma}_{w_d}^{-1} w_d)^{\beta}), \tag{4.1}$$

where $\mathbf{\Sigma}_{w_d}$, commonly known as the scatter matrix, is equal to

rank$(\mathbf{\Sigma}_{w_d})$ $\Gamma((2+n-2)/2\beta)/\Gamma((2+n)/2\beta)$ times the covariance matrix of w_d, $\beta \in (0,1]$, and K is a normalizing constant. For $\beta = 1$, we obtain the probability distribution for the well-known multivariate Gaussian distribution. In the following we briefly provide a background regarding the multivariate generalized Gaussian distribution.

A linear transformation of the multivariate generalized Gaussian random vector is also a multivariate generalized Gaussian random vector.

Proposition 2 *[29] Let u be the a $n \times 1$ multivariate generalized Gaussian random vector with mean $\mu_u \in \mathbb{R}^n$, and scatter matrix, $\mathbf{\Sigma}_u \in \mathbb{R}^{n\times n}$. Let A be a $l \times n$ full rank matrix. Then the $l \times 1$ random vector, $v = Au$ has the multivariate generalized Gaussian distribution with mean $\mu_v = A\mu_u \in \mathbb{R}^l$, and scatter matrix, $\mathbf{\Sigma}_v = A\mathbf{\Sigma}_u A^T \in \mathbb{R}^{l\times l}$.*

If v is a univariate generalized Gaussian random variable, then the probability of v falling in the range of $[\delta - \mu_v, \delta + \mu_v]$, for $\delta \geq 0$, can be found in terms of lower incomplete gamma function.

Proposition 3 *If v is a univariate generalized Gaussian random variable with mean $\mu_v \in \mathbb{R}$ and scatter matrix, $\mathbf{\Sigma}_v \in \mathbb{R}$, then the probability of v falling in the range of $[-\delta + \mu_v, \delta + \mu_v]$, for $\delta \geq 0$, is given by,*

$$\mathbb{P}(|v - \mu_v| \leq \delta) = 2\gamma\left(\frac{1}{2\beta}, \left(\frac{\delta}{\mathbf{\Sigma}_v}\sqrt{\frac{\Gamma(\frac{3}{2\beta})}{\Gamma(\frac{1}{2\beta})}}\right)^{2\beta}\right), \tag{4.2}$$

where $\gamma(.,.)$ is the lower incomplete gamma function, and $\Gamma(.)$ is the ordinary gamma function.

Preliminaries: Let $\mathbf{H} \in \mathbb{R}^{n\times n}$ be the block Toeplitz matrix representing the integral operation so that the integral image, $I = \mathbf{H}x \in \mathbb{R}^n$, and h_i^T for $i = 1,..,n$

be the rows of $\mathbf{H}$. Hence, $I_i = h_i^T x$. We wish to recover the approximate integral image, $\hat{I}$ from the measured vector $y = \phi x$, just by applying a linear operator $\mathcal{L}$ on y, so that $\hat{I} = \mathcal{L}y$. For reasons which will be apparent soon, we assume $\mathcal{L} = \mathbf{H}(\phi^d)^T$, where $\phi^d \in \mathbb{R}^{m\times n}$ such that $\text{rank}(\phi^d) = \text{rank}(\phi)$. We call ϕ^d as the dual of ϕ. Thus by construction $\mathcal{L} \in \mathbb{R}^{n\times m}$. The value at location i in the approximate integral image is given by $\hat{I}_i = h_i^T(\phi^d)^T\phi x$. The distortion in integral image at location i is given by $d_i = \hat{I}_i - I_i = h_i^T((\phi^d)^T\phi x - x)$. Noting $\mathbf{Q} = (\phi^d)^T\phi$, and $n \times n$ identity matrix by $\mathbf{I}$, the distortions can be compactly written as $d_i = h_i^T(\mathbf{Q} - \mathbf{I})x$. We call $\mathbf{d} = [d_1, ..., d_n]$ as distortion vector.

4.2 Optimization problem

Our aim is to search for a measurement matrix ϕ of minimum rank such that distortions, d_i are within allowable limits for all i, jointly, with high probability. $\mathbf{Q}$ by construction is the product of two matrices of identical ranks, ϕ and $(\phi^d)^T$. Hence, we have the relation, $\text{rank}(\mathbf{Q}) = \text{rank}(\phi)$. Inspired by the phase-lifting technique used in [10] and [36], instead of minimizing the rank of ϕ, we minimize the rank of $\mathbf{Q}$. Now, we can formally state the optimization problem as follows.

$$\begin{aligned} &\underset{\mathbf{Q}}{\text{minimize}} \quad \text{rank}(\mathbf{Q}) \\ &\text{s.t} \quad \mathbb{P}(|d_1| \leq \delta_1, .., |d_i| \leq \delta_i .., |d_n| \leq \delta_n) \geq 1 - \epsilon, \end{aligned} \tag{4.3}$$

$$\begin{aligned} &\underset{\mathbf{Q}}{\text{minimize}} \quad \text{rank}(\mathbf{Q}) \\ &\text{s.t} \quad \mathbb{P}(|d_i| \leq \delta_i) \geq 1 - \epsilon, \quad i = 1, .., n. \end{aligned} \tag{4.4}$$

where $\delta_i \geq 0$ denotes the allowable limit of distortion at location i of integral image, and $0 < \epsilon < 1$. Once $\mathbf{Q}^*$ is found, we show later in the section that the SVD decomposition of $\mathbf{Q}^*$ allows us to write $\mathbf{Q}^*$ as a product of two matrices of identical ranks, thus yielding both the measurement matrix, ϕ^* and the desired linear operator, $\mathcal{L}^*$. The constraint in (4.3) is a probabilistic one. Hence to compute it, one needs to assume a statistical prior model for x. Using the model in 4.1, and its properties given in proposition (1) and (2), we arrive at a solvable optimization problem.

Computation of probabilistic constraint in (4.3): Substituting for x, we can write the distortion at location i as $d_i = h_i^T(\mathbf{Q} - \mathbf{I})\mathbf{U}w$. We let all the entries in the first row of ϕ and ϕ^d to be equal to $1/\sqrt{n}$, so that one of the m measurements is exactly equal to $\sqrt{n}\bar{x}$. Further, we denote the rest $m-1$ rows of the two matrices by $\phi_{2:m}$ and $\phi^d_{2:m}$. Now, if we restrict $\phi_{2:m}$ and $\phi^d_{2:m}$ to be respectively equal to $\mathbf{C}\mathbf{U}^T_{2:n}$ and $\mathbf{D}\mathbf{U}^T_{2:n}$ for some $\mathbf{C}$, $\mathbf{D}$ in $\mathbb{R}^{m-1\times n-1}$, then from basic linear algebra we can show that $d_i = h_i^T(\mathbf{P} - \mathbf{I})\mathbf{U}^T_{2:n}w_d$, where $\mathbf{P} = (\phi^d_{2:m})^T\phi_{2:m}$. It is easy to see that $\mathbf{Q} = \mathbf{P} + \frac{1}{n}\mathbf{O}$, and rank($\mathbf{Q}$) = rank($\mathbf{P}$) + 1, where $\mathbf{O}$ is the matrix with all its entries equal to unity. Hence we can replace the objective function in (4.3) by rank($\mathbf{P}$). Rank minimization is a non-convex problem. Hence we relax the objective function to nuclear norm, as is done typically. To compute the constraint in (4.3), one needs to first compute the joint probability of $\mathbf{d} = [d_1, .., d_n]$, and then compute a n dimensional definite integral. Now that d_i's are linear combinations of w_d, it follows from proposition 1, that $\mathbf{d}$ also has a multivariate generalized Gaussian distribution. However, no closed form for the definite integral is known. Hence, we relax the constraint by decoupling it into n independent constraints, each enforcing the constraint that the distortion at a specific location is to be within allowable limits with high probability, independent of the distortions at other locations. The optimization with relaxed constraints is thus given by

$$\begin{aligned} &\underset{\mathbf{P}}{\text{minimize}} \quad \|\mathbf{P}\|_* \\ &\text{s.t} \quad \mathbb{P}(|d_i| \leq \delta_i) \geq 1-\epsilon, \quad i=1,..,n. \end{aligned} \tag{4.5}$$

Now, $d_i = h_i^T(\mathbf{P}-\mathbf{I})\mathbf{U}_{2:n}^T w_d$, is a linear combination of the entries of w_d. From the proposition 2, d_i has a one-dimensional generalized Gaussian distribution with zero mean and scatter parameter, $\left\|\mathbf{\Sigma}_{w_d}^{1/2}\mathbf{U}_{2:n}^T(\mathbf{P}-\mathbf{I})^T h_i\right\|$, and the probability in equation 4.5 can be explicitly written as follows.

$$\begin{aligned} &\mathbb{P}(|d_i| \leq \delta_i) \\ &= 2\gamma\left(\frac{1}{2\beta}, \left(\frac{\delta_i}{\left\|\mathbf{\Sigma}_{w_d}^{1/2}\mathbf{U}_{2:n}^T\mathbf{Q}^T h_i - \mathbf{\Sigma}_{w_d}^{1/2}\mathbf{U}_{2:n}^T h_i\right\|_2}\sqrt{\frac{\Gamma(\frac{3}{2\beta})}{\Gamma(\frac{1}{2\beta})}}\right)^{2\beta}\right). \end{aligned} \tag{4.6}$$

$$\begin{aligned} &\mathbb{P}(|d_i| \leq \delta_i) \\ &= 2\gamma\left(\frac{1}{2\beta}, \left(\frac{\delta_i}{\left\|\mathbf{\Sigma}_{w_d}^{1/2}\mathbf{U}_{2:n}^T\mathbf{P}^T h_i - \mathbf{\Sigma}_{w_d}^{1/2}\mathbf{U}_{2:n}^T h_i\right\|_2}\sqrt{\frac{\Gamma(\frac{3}{2\beta})}{\Gamma(\frac{1}{2\beta})}}\right)^{2\beta}\right). \end{aligned} \tag{4.7}$$

The optimization problem now can be rewritten as

$$\begin{gathered} \underset{\mathbf{P}}{\text{minimize}}\, \|\mathbf{P}\|_* \qquad \text{s.t} \\ 2\gamma\left(\frac{1}{2\beta}, \left(\frac{\delta_i}{\left\|\mathbf{\Sigma}_{w_d}^{1/2}\mathbf{U}_{2:n}^T\mathbf{P}^T h_i - \mathbf{\Sigma}_{w_d}^{1/2}\mathbf{U}_{2:n}^T h_i\right\|_2}\sqrt{\frac{\Gamma(\frac{3}{2\beta})}{\Gamma(\frac{1}{2\beta})}}\right)^{2\beta}\right) \\ \geq 1-\epsilon \qquad\qquad\qquad\qquad . \end{gathered}$$

We compactly write $\mathbf{\Sigma}_{w_d}^{1/2}\mathbf{U}_{2:n}^T\mathbf{P}^T h_i$ as $\mathcal{A}_i(\mathbf{P})$, $\mathbf{\Sigma}_{w_d}^{1/2}\mathbf{U}_{2:n}^T h_i$ as b_i and $\frac{\delta_i}{(\gamma^{-1}(\frac{1}{2\beta},\frac{1-\epsilon}{2}))^{\frac{1}{2\beta}}}\sqrt{\frac{\Gamma(\frac{3}{2\beta})}{\Gamma(\frac{1}{2\beta})}}$ as Δ_i. Plugging above in equation (4.7), and rearranging terms, we have

$$\begin{aligned} &\underset{\mathbf{P}}{\text{minimize}} \quad \|\mathbf{P}\|_* \\ &\text{s.t} \quad \|\mathcal{A}_i(\mathbf{P}) - b_i\|_2 \leq \Delta_i \quad i=1,..,n. \end{aligned} \tag{4.8}$$

We can rewrite the problem in conic form as below.

$$\begin{gathered}(P1) \quad \underset{\mathbf{Q}}{\text{minimize}} \quad \|\mathbf{Q}\|_* \qquad \text{s.t} \\ \begin{bmatrix} b_i - \mathcal{A}_i(\mathbf{Q}) \\ \Delta_i \end{bmatrix} \in \mathcal{K}_i, \quad i = 1, .., n, \end{gathered} \tag{4.9}$$

$$\begin{gathered}(P1) \quad \underset{\mathbf{P}}{\text{minimize}} \quad \|\mathbf{P}\|_* \qquad \text{s.t} \\ \begin{bmatrix} b_i - \mathcal{A}_i(\mathbf{P}) \\ \Delta_i \end{bmatrix} \in \mathcal{K}_i, \quad i = 1, .., n, \end{gathered} \tag{4.10}$$

where $\mathcal{K}_i$ is a second order cone $\mathcal{K}_i = \{(x_i, t_i) \in \mathbb{R}^{n+1} : \|x_i\| \leq t_i\}$. Let $\mathbf{b} = [b_1, .., b_n]$, and $\mathcal{A}(\mathbf{P}) = [\mathcal{A}_1(\mathbf{P}), .., \mathcal{A}_n(\mathbf{P})]$. Let $\mathcal{A}^*$ denote the adjoint of the linear operator $\mathcal{A}$. It is easy to recognize that the optimization above is a convex problem, since nuclear norm is convex and the constraints enforce finite bounds on the norms of affine functions of the decision variable, $\mathbf{P}$ and hence are also convex. Even though the constraints are second-order cone constraints, the standard second order conic programming methods cannot be used to solve $(P1)$ since nuclear norm is non-smooth. The nuclear norm is smoothened by the addition of a square of Forbenius norm of the matrix, and is replaced by $\tau \|\mathbf{P}\|_* + \frac{1}{2} \|\mathbf{P}\|_F^2$, where $\tau > 0$. The optimization problem with the smoothened objective function is given in 4.11.

$$\begin{gathered}(P2) \quad \underset{\mathbf{P}}{\text{minimize}} \quad \tau \|\mathbf{P}\|_* + \frac{1}{2} \|\mathbf{P}\|_F^2 \\ \text{s.t} \begin{bmatrix} b_i - \mathcal{A}_i(\mathbf{P}) \\ \Delta_i \end{bmatrix} \in \mathcal{K}_i, \qquad i = 1, .., n. \end{gathered} \tag{4.11}$$

Recently, many algorithms [58, 8] have been developed to tackle nuclear norm minimization problem of this form in the context of matrix completion. We use SVT (singular value thresholding) algorithm [8] to solve $(P2)$.

SVT iteration to solve $(P2)$: Here, we first briefly describe the SVT algorithm for smoothened nuclear norm minimization with general convex constraints, and later we show how we adapt the same to our problem $P2$ which has n second-order constraints. Let the smoothened nuclear norm with general convex constraints, be given as below.

$$\begin{aligned} \underset{\mathbf{P}}{\text{minimize}} \quad & \tau \left\|\mathbf{P}\right\|_* + \frac{1}{2}\left\|\mathbf{P}\right\|_F^2 \\ \text{s.t} \quad & f_i(\mathbf{P}) \leq 0, \quad i = 1,..,n, \end{aligned} \tag{4.12}$$

where $f_i(\mathbf{P}) \leq 0, i = 1,..,n$ denote the n convex constraints. Let $\mathcal{F}(\mathbf{P}) = [f_1(\mathbf{P}),..,f_n(\mathbf{P})]$. The SVT algorithm for the 4.12 with the modified objective function is given as below.

$$\left.\begin{aligned} \mathbf{P}^k &= \arg\min_{\mathbf{P}} \quad \tau \left\|\mathbf{P}\right\|_* + \tfrac{1}{2}\left\|\mathbf{P}\right\|_F^2 + \langle \mathbf{z}^{k-1}, \mathcal{F}(\mathbf{P})\rangle \\ \mathbf{z_i}^k &= P_i\left(\mathbf{z_i}^{k-1} + \eta^k f_i(\mathbf{P}^k)\right), \quad i = 1,..,n \end{aligned}\right\} \tag{4.13}$$

where $\mathbf{z}^k$ is a short form for $[\mathbf{z_1}^k,..,\mathbf{z_n}^k]$, and $P_i(\mathbf{q})$ denotes the projection of $\mathbf{q}$ onto the convex set defined by the constraint $f_i(\mathbf{P}) \leq 0$. Let $\mathbf{z_i}^k = [\mathbf{y_i}^k, s_i^k]$, so that the vector $\mathbf{y_i}^k$ denotes the first n elements of $\mathbf{z_i}^k$ and s_i^k denotes the last element of $\mathbf{z_i}^k$. Let $\mathbf{y}^k$ be a short form for $[\mathbf{y_1}^k,..,\mathbf{y_n}^k]$, and s^k is a short form for $[s_1^k,..,s_n^k]$. To obtain a explicit form of the update equations in 4.13 for our problem, $P1$, let us consider the first equation of the same. $\mathcal{F}(\mathbf{P})$ for $P1$ is given by $[b_1 - \mathcal{A}_1(\mathbf{P}), \Delta_1,..,b_n - \mathcal{A}_n(\mathbf{P}), \Delta_n]^T$. We substitute for $\mathcal{F}(\mathbf{P})$, and after removal of the terms not involving $\mathbf{P}$, we have

$$\mathbf{P}^k = \arg\min_{\mathbf{P}} \quad \tau \left\|\mathbf{P}\right\|_* + \frac{1}{2}\left\|\mathbf{P}\right\|_F^2 + \langle \mathbf{y}^{k-1}, \mathbf{b} - \mathcal{A}(\mathbf{P})\rangle. \tag{4.14}$$

Equation 4.14 can be rewritten as below.

$$\begin{aligned} \mathbf{P}^k = \arg\min_{\mathbf{P}} \quad & \tau \left\|\mathbf{P}\right\|_* + \frac{1}{2}\left\|\mathbf{P} - \mathcal{A}^*(\mathbf{y}^{k-1})\right\|_F^2 \\ & -\frac{1}{2}\left\|\mathcal{A}^*(\mathbf{y}^{k-1})\right\|_F^2 + \langle \mathbf{P}, \mathcal{A}^*(\mathbf{y}^{k-1})\rangle \\ & +\langle \mathbf{y}^{k-1}, \mathbf{b}\rangle - \langle \mathbf{y}^{k-1}, \mathcal{A}(\mathbf{P})\rangle. \end{aligned}$$

Removing the terms not involving $\mathbf{P}$ and noting that $\langle \mathbf{P}, \mathcal{A}^*(\mathbf{y}^{k-1})\rangle = \langle \mathbf{y}^{k-1}, \mathcal{A}(\mathbf{P})\rangle$, we have the following.

$$\mathbf{P}^k = \arg\min_{\mathbf{P}} \quad \tau \left\|\mathbf{P}\right\|_* + \frac{1}{2}\left\|\mathbf{P} - \mathcal{A}^*(\mathbf{y}^{k-1})\right\|_F^2 . \tag{4.15}$$

Before we write down the solution to equation 4.15, we first define $\mathcal{D}_\tau$, the singular value shrinkage operator. Consider the SVD of a matrix $\mathbf{X}$, given by $\mathbf{X} = \mathbf{W\Sigma V}^T$. Then for $\tau \geq 0$, the singular value shrinkage operator, $\mathcal{D}_\tau$ is given by $\mathcal{D}_\tau(\mathbf{X}) = \mathbf{W}\mathcal{D}_\tau(\mathbf{\Sigma})\mathbf{V}^T, \mathcal{D}_\tau(\mathbf{\Sigma}) = diag(\{(\sigma_i - \tau)_+\})$, where $t_+ = \max(0,t)$. The solution to equation 4.15 is given by $\mathbf{P}^k = \mathcal{D}_\tau(\mathcal{A}^*(\mathbf{y}^{k-1}))$. Now, it remains to calculate $\mathcal{A}^*(\mathbf{y}^{k-1})$. We achieve it according to the following. Consider $\langle \mathcal{A}^*(\mathbf{y}^{k-1}), \mathbf{P}\rangle$.

$$\begin{aligned} &\langle \mathbf{P}, \mathcal{A}^*(\mathbf{y}^{k-1})\rangle \\ = \langle \mathcal{A}(\mathbf{P}), \mathbf{y}^{k-1}\rangle = &\sum_{i=1}^{n} \langle \mathcal{A}_i(\mathbf{P}), \mathbf{y_i}^{k-1}\rangle \\ = \sum_{i=1}^{n} \langle \mathbf{P}, \mathcal{A}_i^*(\mathbf{y_i}^{k-1})\rangle = \langle \mathbf{P}, &\sum_{i=1}^{n} \mathcal{A}_i^*(\mathbf{y_i}^{k-1})\rangle \end{aligned}$$

Hence, we have $\mathcal{A}^*(\mathbf{y}^{k-1}) = \sum_{i=1}^{n} \mathcal{A}_i^*(\mathbf{y_i}^{k-1})$. Thus, the first equation of SVT iteration for our problem is given by

$$\mathbf{P}^k = \mathcal{D}_\tau\left(\sum_{i=1}^{n} \mathcal{A}_i^*(\mathbf{y_i}^{k-1})\right). \tag{4.16}$$

Using basic linear algebra, it can be shown that $\mathcal{A}_i^*(\mathbf{y_i}^{k-1}) = \mathbf{U}_{2:n}(\Sigma_{w_d}^{1/2})^T \mathbf{y_i}^{k-1} h_i^T$.

We now provide the projection onto the convex cone $\mathcal{K}_i$. The projection operator, $P_{\mathcal{K}_i}$ as derived in [30] is given as follows.

$$P_{\mathcal{K}_i} : (x,t) \mapsto \begin{array}{ll} (x,t), & \|x\| \leq t, \\ \frac{\|x\|+t}{2\|x\|}(x, \|x\|), & -\left\|x\right\| \leq t \leq \left\|x\right\|, \\ (0,0), & t \leq -\left\|x\right\|. \end{array} \tag{4.17}$$

To solve $(P2)$, starting with $\begin{bmatrix} \mathbf{y}_i^0 \\ s_i^0 \end{bmatrix} = \mathbf{0}$ for all $i = 1, ..n$, the k^{th} SVT iteration is given by (4.18).

$$\left.\begin{aligned} &\mathbf{P}^k = \mathcal{D}_\tau \left(\sum_{i=1}^{n} \mathbf{U}_{2:n} (\Sigma_{w_d}^{1/2})^T \mathbf{y_i}^{k-1} h_i^T \right) \\ &\begin{bmatrix} \mathbf{y}_i^k \\ s_i^k \end{bmatrix} = P_{\mathcal{K}_i} \left(\begin{bmatrix} \mathbf{y}_i^{k-1} \\ s_i^{k-1} \end{bmatrix} + \eta^k \begin{bmatrix} b_i - \mathcal{A}_i(\mathbf{P}^k) \\ -\Delta_i \end{bmatrix} \right), \end{aligned}\right\} \tag{4.18}$$

where, $\mathcal{A}_i^*$ are the adjoints of linear operators $\mathcal{A}_i$. For the iterations (4.18) to converge, we need to choose the step sizes, $\eta^k \leq \frac{2}{\|\mathcal{A}\|_2^2}$, where $\|\mathcal{A}\|_2$ is the spectral norm of the linear transformation $\mathcal{A}$ [8].

Once the solution $\mathbf{P}^*$ is found, using the relation noted earlier in the section, we have $\mathbf{Q}^* = \mathbf{P}^* + \frac{1}{n}\mathbf{O}$. Having obtained $\mathbf{Q}^*$, the task now is to express it into a product of two matrices of identical ranks. This is done almost trivially as follows. Noting the singular value decomposition of $\mathbf{Q}^*$ as $\mathbf{Q}^* = \mathbf{W}_M \mathbf{\Sigma}_M \mathbf{V}_M^T$, where $\mathbf{\Sigma}_M = diag\{\lambda_1, .., \lambda_M\}$ denote the diagonal matrix with the non-zero singular values arranged along its diagonal, and $\mathbf{W}_M$ and $\mathbf{V}_M^T$ are the matrices whose columns are the left and right singular vectors respectively. We can choose $\phi^* = \mathbf{\Sigma}_M^{1/2} \mathbf{V}_M^T$, so that $((\phi^d)^*)^T = \mathbf{W}_M \mathbf{\Sigma}_M^{1/2}$ and $\text{rank}(\phi^*) = \text{rank}((\phi^d)^*) = \text{rank}(\mathbf{Q}^*)$. We, henceforth refer to ϕ^* as **ReFInE** ϕ, and the corresponding measurements, $y = \phi^* x$ as **ReFInE** measurements. The desired linear operator $\mathcal{L}^*$ is given by $\mathbf{H}\mathbf{W}_M \mathbf{\Sigma}_M^{1/2}$. By construction, the approximate integral image $\hat{I}$ is given by $\hat{I} = \mathcal{L}^* y = (\mathbf{H}\mathbf{W}_M \mathbf{\Sigma}_M^{1/2})y$.

Computational Complexity: Since the length of the image x is n, the number of entries in $\mathbf{P}$ is n^2. Hence, the dimension of the optimization problem $P2$ is n^2. This means that if we are to optimize for a measurement matrix to operate on an image of size, 256 × 256, the dimension of the optimization problem would be 2^{32}! Optimizing over a large number of variables is computationally expensive, if not impractical. Hence we propose the following suboptimal solution to obtain the measurement matrix. We divide the image into non-overlapping blocks of fixed size, and sense each block using a measurement matrix, optimized for this fixed size, independently of other blocks. Let the image, x be divided into B blocks, $x_1, x_2, .., x_B$, each of a fixed size, $f \times f$, and $\phi_f \in \mathcal{R}^{m \times f^2}$ be the measurement matrix optimized for images of size $f \times f$, and $(\phi_f^d)^T$ be the corresponding dual matrix. Then the '**ReFInE**' measurements, y are given by the following,

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_B \end{bmatrix} = \begin{bmatrix} \phi_f & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \phi_f & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \phi_f \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_B \end{bmatrix}. \tag{4.19}$$

Once the measurements, y are obtained, the integral image, $\hat{I}$ is given by

$$\hat{I} = \mathbf{H} \begin{bmatrix} (\phi_f^d)^T & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & (\phi_f^d)^T & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & (\phi_f^d)^T \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_B \end{bmatrix}. \tag{4.20}$$

4.3 Experiments

Before we can conduct experiments to evaluate our framework, we first need to estimate the parameters of the probability model in 4.1. Estimating parameters of the probability model and optimizing measurement matrices for any arbitrary large sized image blocks is not practical since the former task requires an enormous amount of image data and the latter requires prohibitive amount of memory and computational resources. Hence, we fix the block size to be 32×32 images. The scatter matrix $\mathbf{\Sigma}_{w_d}$ is a scalar multiple of the covariance matrix of w_d. Hence it suffices to compute the covariance matrix. To this end, we first downsample all the 5011 training images in

PASCAL VOC 2007 dataset [28] to a size of 32×32, so that $n = 1024$ and then obtain the level 7 Daubechies wavelet coefficient vectors. We compute the sample covariance matrix of thus obtained wavelet coefficient data. For various values of β, we evaluate the χ^2 distance between the histograms of the individual wavelet coefficients and their respective theoretical marginal distributions with the variances computed above. We found for $\beta = 0.68$, the distance computed above is minimum.

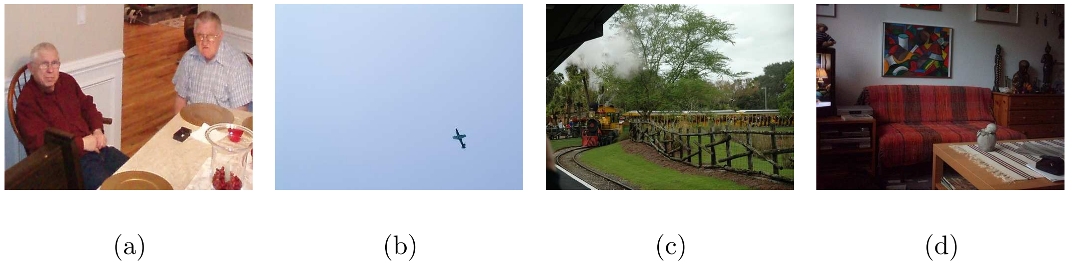
Computing measurement matrix: To obtain a measurement matrix, we need to input a desired distortion vector δ to the optimization problem in $(P2)$. The desired distortion vector is computed according to the following. We first perform principal component analysis (PCA) on the downsampled 5011 training images in the PASCAL VOC 2007 dataset [28]. We use only the top 10 PCA components as ϕ to 'sense' these images. We obtain the desired distortion vector by first assuming $\phi^d = \phi$ and calculating distortions, $|d_i^j|$ at each location for all training images, $j = 1, .., 5011$. Now, the entry in location i of the desired δ is given by the minimum value α, so that 95% of the values, $|d_i^j|, j = 1, .., 5011$ are less than α. We use $\epsilon = 0.95$ and solve $(P2)$ to obtain $\mathbf{P}^*$, and hence also $\mathbf{Q}^*$. The rank of **ReFInE** ϕ is simply the rank of $\mathbf{Q}^*$.

Estimation of integral images: We show that good quality estimates of integral images can be obtained using our framework. To this end, we first construct **ReFInE** measurement matrices of various ranks, M. We achieve this by considering the SVD of $\mathbf{Q}^*$ obtained above. For a particular value of M, the **ReFInE** ϕ is calculated according to $\phi_M = \mathbf{\Sigma}_M^{1/2} \mathbf{V}_M^T$, where $\mathbf{\Sigma}_M = diag\{\lambda_1, .., \lambda_M\}$ is a diagonal matrix with M largest singular values arranged along the diagonal and $\mathbf{V}_M^T$ denote the corresponding rows in $\mathbf{V}^T$. Its dual, ϕ^d, is calculated similarly. For each particular measurement rate, determined by the value of M, the integral image estimates are recovered from M **ReFInE** measurements for all the 4952 test images in the PASCAL VOC 2007 dataset [28]. Similarly integral image estimates are recovered

Method	**ReFInE**	RG-CoSamP	**ReFInE**	RG-CoSamP	**ReFInE**	RG-CoSamP
M (measurement ratio)	20 (0.005)	20 (0.005)	40 (0.01)	40 (0.01)	60 (0.015)	60 (0.015)
Time in s	0.0034	0.38	0.0036	0.58	0.0031	0.97
RSNR in dB	38.95	-16.76	38.96	-11.22	38.96	-10.9

Table 4.1: Comparison of average RSNR and time for recovered integral image estimates obtained using our method with RG-CoSamP. Our framework outperforms RG-CoSamP in terms of both recovery signal-to-noise ratio and time taken to estimate the integral image, at all measurement rates.

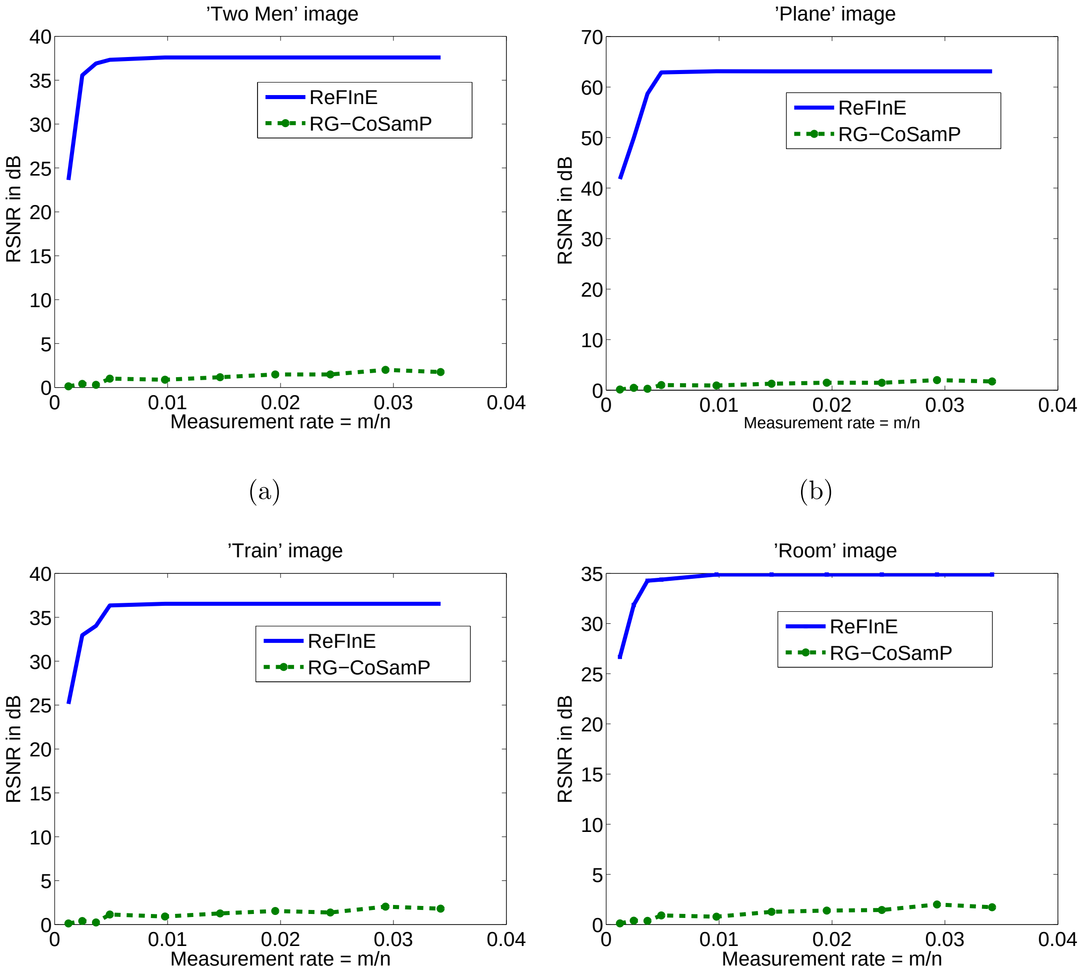
from random Gaussian measurements by first performing non-linear iterative reconstruction using the CoSamP algorithm [69] and then applying the integral operation on the reconstructed images. This pipeline is used as baseline to compare integral estimates, and henceforth is referred to as 'RG-CoSamP'. We then measure the recovered signal-to-noise ratio (RSNR) via $20\ \log_{10}\left(\frac{\|\hat{I}\|_F}{\|\hat{I}-I\|_F}\right)$. The average RSNR for recovered integral image estimates as well as the time taken to obtain integral images are tabulated in the table 4.1. Our framework outperforms RG-CoSamP in terms of both recovery signal-to-noise ratio and time taken to estimate the integral image, at all measurement rates. This shows that **ReFInE** ϕ, the measurement matrices designed by our framework, facilitate faster and more accurate recovery of integral image estimates than the universal matrices. The average time taken to obtain integral image estimates in our framework is about 0.003s, which amounts to a real-time speed of 300 FPS. Further, we randomly select four images ('Two Men', 'Plane', 'Train' and 'Room') from the test set (shown in figure 4.1(a), 4.1(b), 4.1(c), 4.1(d)) and present qualitative and quantitative results for individual images. Image-wise RSNR v/s measurement rate plots are shown in figure 4.2. It is very clear that for all the four images, our framework clearly outperforms RG-CoSamP in terms of RSNR, at all measurement rates.



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

Figure 4.1: Four images (L-R: ‘Two Men’, ‘Plane’, ‘Train’ and ‘Room’) are randomly chosen for presenting qualitative and quantitative results.

Estimation of box-filtered outputs: It is well known that box-filtered outputs of any size can efficiently computed using integral images [91]. To show the capability of our framework in recovering good quality box-filtered output estimates, we conduct the following experiment. For box filters of sizes 3×3, 5×5 and 7×7, we compute the estimates of filtered outputs for the four images using their respective recovered integral image estimates. RSNR v/s measurement rate plots for different filter sizes are shown in figure 4.3. It is evident that even for a remarkably low measurement rate of 1% , we obtain high RSNR box-filtered outputs. For a fixed measurement rate, expectedly the RSNR increases with the size of the filter. This shows the structures which are more global in nature are captured better. This is particularly true in the case of ‘Plane’ image. The high RSNR for this image hints at the absence of fine structures and homogeneous background. Further, for the ‘Two Men’ Image, we also compare the heat maps of the exact box-filtered outputs with the estimated ones. We fix the measurement rate to 1%. For filter sizes 3×3 and 7×7, the exact box-filtered outputs are computed and compared with the box-filtered output estimates obtained using our framework, and RG-CoSamP as well. The heat map visualizations of the outputs are shown in figure 4.4. It is clear that greater quality box-filtered output estimates can be recovered using our framework and the recovered outputs retain the



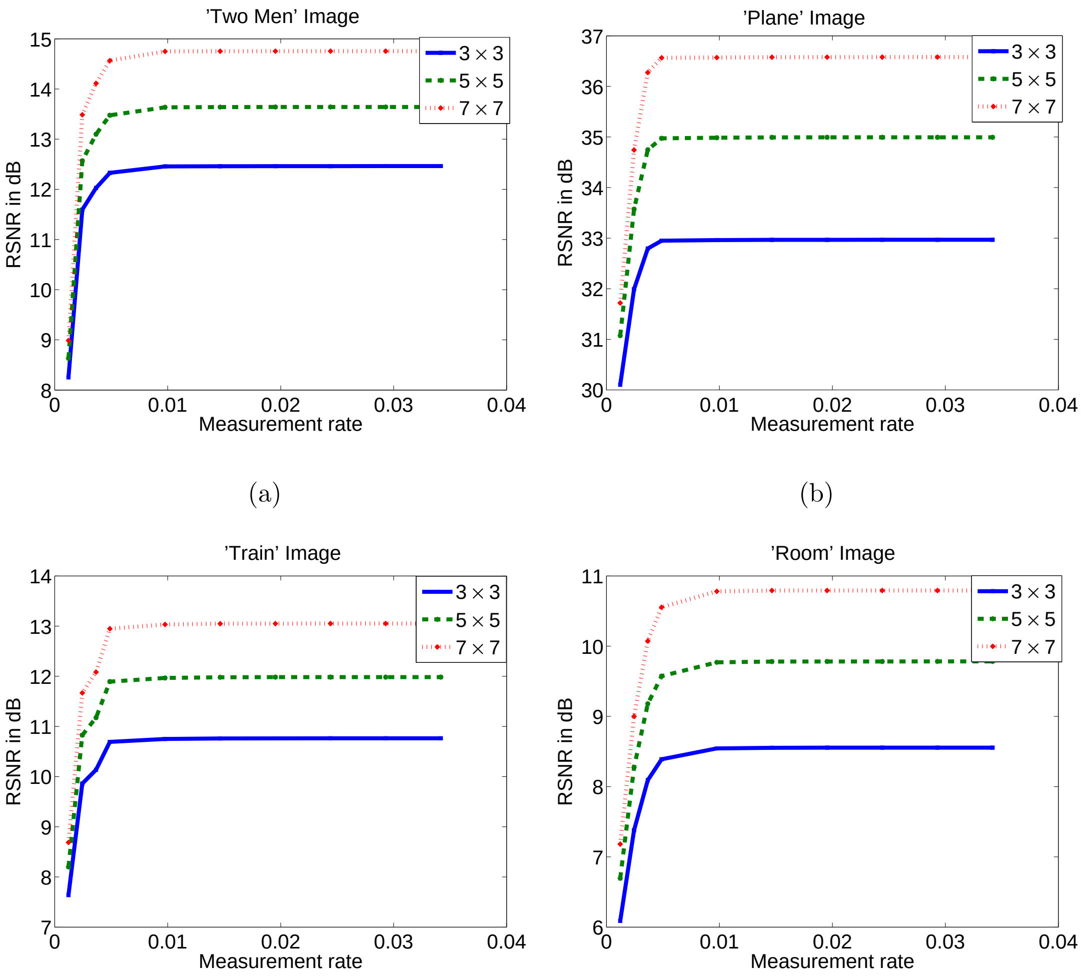
(a)

(b)

(c)

(d)

Figure 4.2: The figure shows variation of image-wise RSNR for recovered integral image estimates for the four images. It is very clear that for all the four images, our framework outperforms 'RG-CoSamP' in terms of RSNR, at all measurement rates.



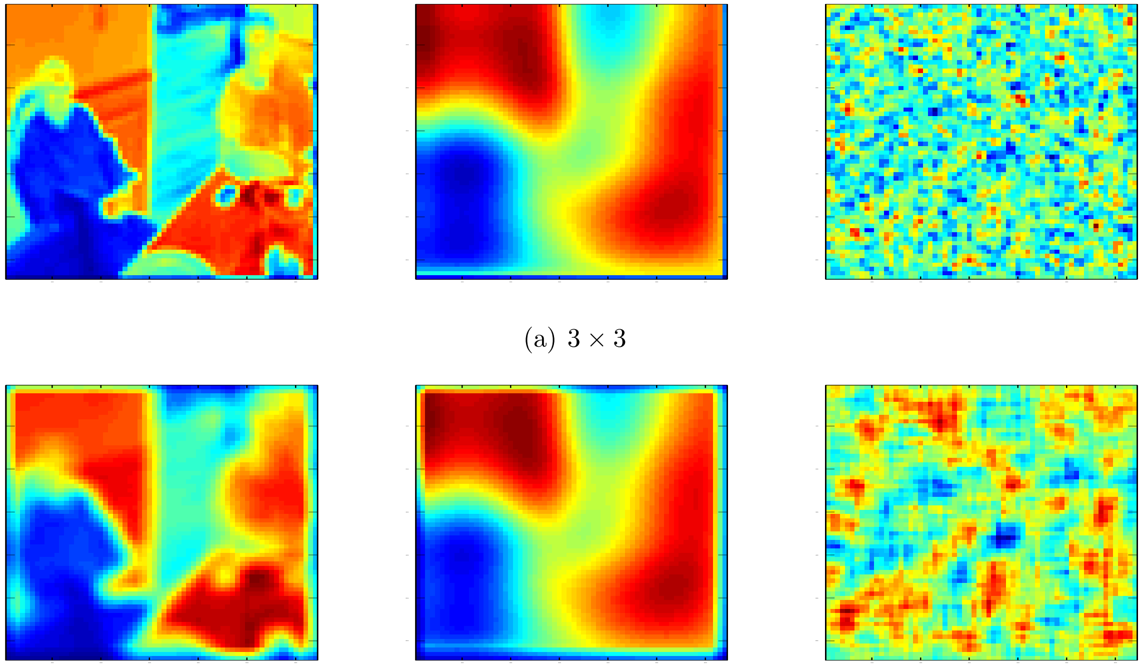
(a)

(b)

(c)

(d)

Figure 4.3: The figure shows the variation of RSNR for the recovered box-filtered outputs using **ReFInE** with measurement rate. It is evident that even for 1% measurement rate, we obtain high RSNR box-filtered outputs. For a fixed measurement rate, the RSNR increases with the size of the filter. This shows the structures global in nature are captured better. This is particularly true in the case of ‘Plane’ image. The high RSNR for this image hints at the absence of fine structures and homogeneous background.



(a) 3 × 3

(b) 7 × 7

Figure 4.4: Heat maps for box-filtered outputs for the 'Two men' image. Left to right: Exact output, **ReFInE** (m/n = 0.01), and RG-CosamP (m/n = 0.01). It is clear that greater quality box-filtered output estimates can be recovered from **ReFInE** measurements and the recovered outputs retain the information regarding medium-sized structures in the images, while in case of RG-CoSamP, the output is all noisy and does not give us any information.

information regarding medium-sized structures in the images, while in the case of RG-CoSamP, the output is all noisy and does not give us any information.

4.4 Tracking Application

In this section, we show the utility of the framework in practical applications. In particular, we show tracking results on 50 challenging videos used in benchmark comparison of tracking algorithms [97]. We emphasize that our aim is not to obtain state-of-the-art tracking results but to show that integral image estimates can be used to obtain robust tracking results at low measurement rates. To this end, we conduct two sets of tracking experiments, one with original resolution videos and one with high definition videos.

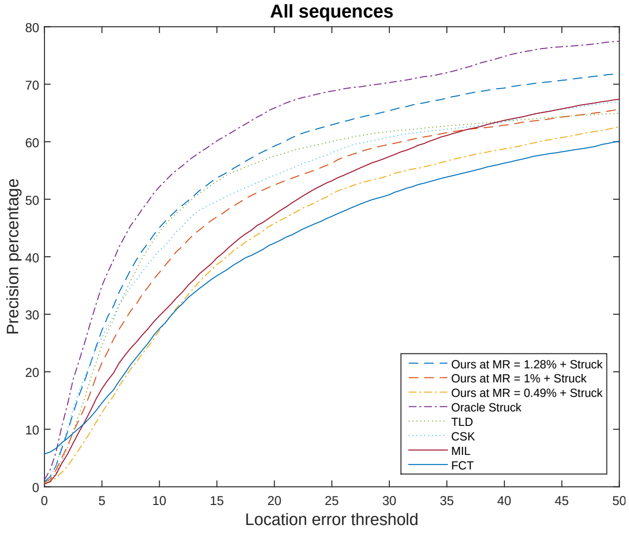
Tracking with original resolution videos: We conduct tracking experiments on original resolution videos at three different measurement rates, viz 1.28%, 1%, and 0.49%. In each case, we use the measurement matrix obtained for block size of 32×32, and obtain **ReFInE** measurements for each frame using the ϕ^* obtained as above. Once, the measurements are obtained, our framework recovers integral image estimates from these measurements in real time. The estimates are subsequently fed into the best performing Haar-feature based tracking algorithm, Struck [35] to obtain the tracking results. Henceforth, we term this tracking pipeline as **ReFInE+Struck**. To evaluate our tracking results, we use the standard criterion of precision curve as suggested by [97]. To obtain the precision curve, we plot the precision scores against a range of thresholds. A frame contributes to the precision score for a particular threshold, α if the distance between the ground truth location of the target and estimated location by the tracker is less than the threshold, α. Precision score for given threshold is calculated as the percentage of frames which contribute to the precision score. When precision scores are required to be compared with other trackers at one particular threshold, generally threshold is chosen to be equal to 20 pixels [97].

Precision curve: The precision curves for our framework at the three different measurement rates are plotted against a wide range of location error thresholds, and are compared with the same for other trackers, Oracle Struck [35], and various other trackers, TLD [46], MIL [4], CSK [37], and FCT [101] in figure 4.5. It is to be noted all the trackers used for comparison utilize full-blown images for tracking and hence operate at 100% measurement rate. As can be seen clearly, 'ReFInE+Struck' at

1.28% performs better than other trackers, MIL, CSK, TLD, and FCT and only a few percentage points worse than Oracle Struck for all thresholds. In particular, the mean precision over all 50 sequences in the dataset [97] for the threshold of 20 pixels is obtained for 'ReFInE+Struck' at three different measurement rates and is compared with other trackers in table 4.2. We obtain a precision of 59.26% at a measurement of 1.28%, which is only a few percentage points less than precision of 65.5% using Oracle Struck and 60.8% using TLD. Even at an extremely low measurement rate of 0.49%, we obtain mean precision of 45.78% which is competitive when compared to other trackers, MIL, and FCT which operate at 100% measurement rate. This clearly suggests that the small number of well-tailored measurements obtained using our framework retain enough information regarding integral images and hence also the Haar-like features which play a critical role in achieving tracking with high precision.

Frame rate: Even though, our framework uses Struck tracker, the frame rates at which 'ReFInE+Struck' operates are potentially less than the frame rate that can be obtained with Oracle Struck, and can even be different at different measurement rates. This is due to the fact that once the measurements are obtained for a particular frame, we first have to obtain an intermediate reconstructed frame before applying the integral operation. However, in the case of Oracle Struck, the integral operation is applied directly on the measured frame. The frame rate for 'Our+Struck' at different measurement rates are compared with the frame rates for other trackers in table 4.2. However, as can be seen, the preprocessing operation to obtain the intermediate reconstructed frame barely affects the speed of tracking since the preprocessing step, being multiplication of small-sized matrices can be efficiently at nearly 1000 frames per second.

Experiments with sequence attributes: Each video sequence in the benchmark dataset is annotated with a set of attributes, indicating the various challenges



(a)

Figure 4.5: 'ReFInE+Struck' at a measurement rate of 1.28% performs better than other trackers, MIL, CSK, TLD, and FCT and only a few percentage points worse than Oracle Struck for all thresholds. Even at a measurement rate of 1%, 'ReFInE+Struck' performs nearly as well as TLD and CSK trackers which operate at 100% measurement rate.

the video sequence offers in tracking. We plot precision percentages against the location error threshold for each of these 10 different kinds of attributes. Figure 4.6 shows the corresponding plots for attributes, 'Illumination Variation', 'Background Clutter', 'Occlusion', and 'Scale Variation'. In the case of 'Illumination Variation' and 'Occlusion' 'Our+Struck' at measurement rate of 1.28% performs better than TLD, CSK, MIL and FCT, whereas in the case of the 'Background Clutter' and 'Scale Variation' attributes, TLD performs slightly better than 'Our+Struck' at measurement rate of 1.28%.

Figure 4.7 shows the corresponding plots for attributes, 'Deformation', 'Fast Motion', 'Motion Blur', and 'Low Resolution'. In the cases of 'Deformation', 'Fast Motion' and 'Motion Blur', 'ReFInE+Struck' at measurement rate of 1.28% performs

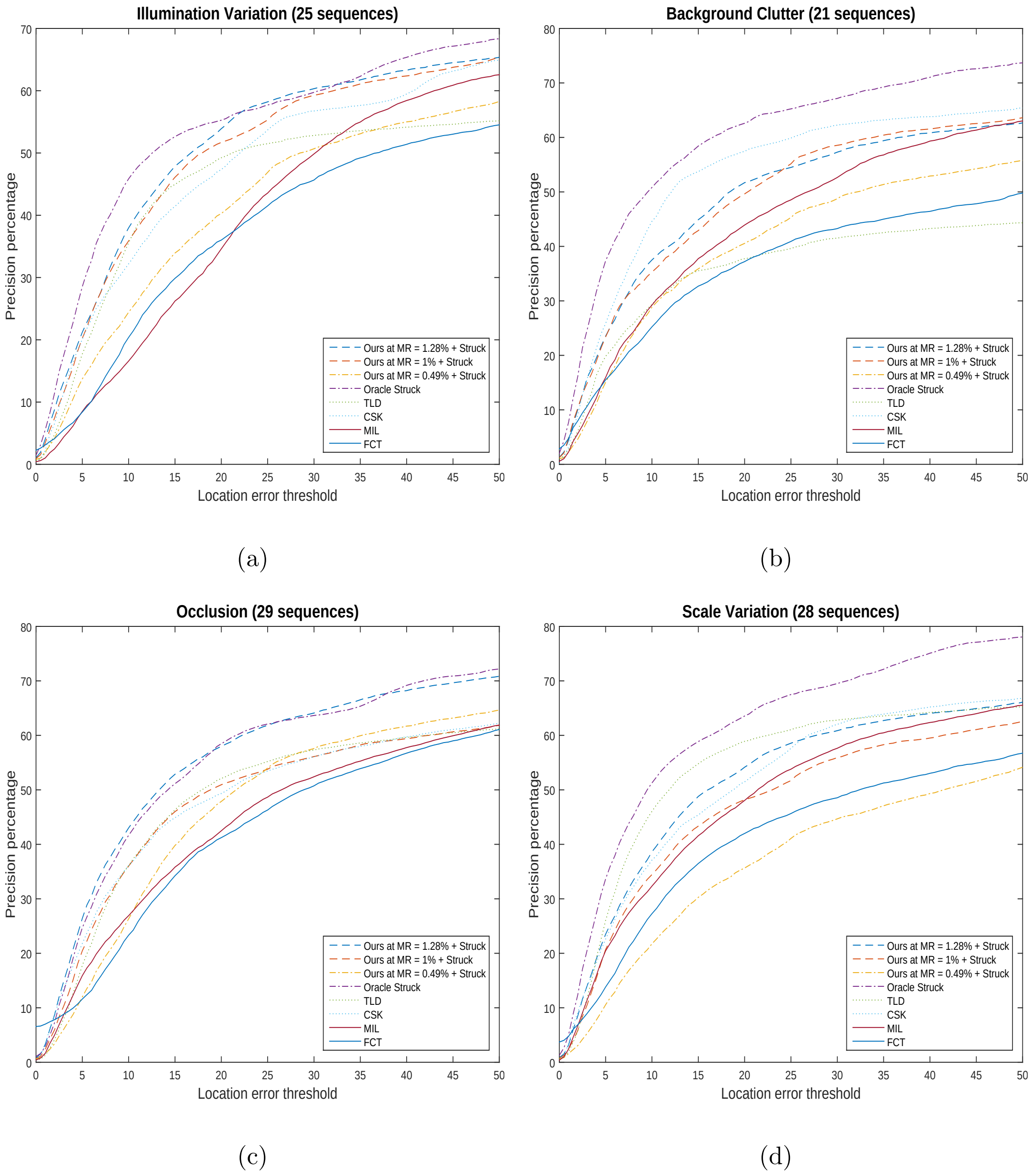
Tracker	Mean Precision	Mean FPS
ReFInE at MR = 1.28% + Struck	59.26	19.61
ReFInE at MR = 1% + Struck	52.47	19.61
ReFInE at MR = 0.49% + Struck	45.78	19.62
Oracle Struck [35]	65.5	20
TLD [46]	60.8	28
CSK [37]	54.11	362
MIL [4]	47.5	38
FCT [101]	42.37	34.92

Table 4.2: Mean precision percentage for 20 pixels error and mean frames per second for various state-of-the-art trackers are compared with our framework at different measurement rates. The precision percentages for our framework are stable even at extremely low measurement rates, and compare favorably with other trackers which operate at 100% measurement rate, i.e utilize all the pixels in the frames.

better than TLD, CSK, MIL and FCT, whereas in the case of 'Low Resolution', TLD performs better than 'ReFInE+Struck'.

Figure 4.8 shows the corresponding plots for attributes, 'In the Plane rotation', 'Out of View', and 'Out of Plane rotation'.

Tracking with high resolution videos: Tracking using high-resolution videos can potentially lead to improvement in performance due to availability of fine-grained information regarding the scene. However, in many applications, the deployment of high-resolution sensors is severely limited by the lack of storage capacity. In such scenarios, it will be interesting to see if the small number of ReFInE measurements of high-resolution videos can yield better tracking performance than the full-blown low-resolution videos. To conduct tracking experiments on high resolution videos, we first employ a deep convolutional network based image super resolution (SR) algorithm, SRCNN, [21] to obtain high resolution frames of the 50 videos considered earlier in the section. The aspect ratio for all frames is maintained, and the upscaling factor for



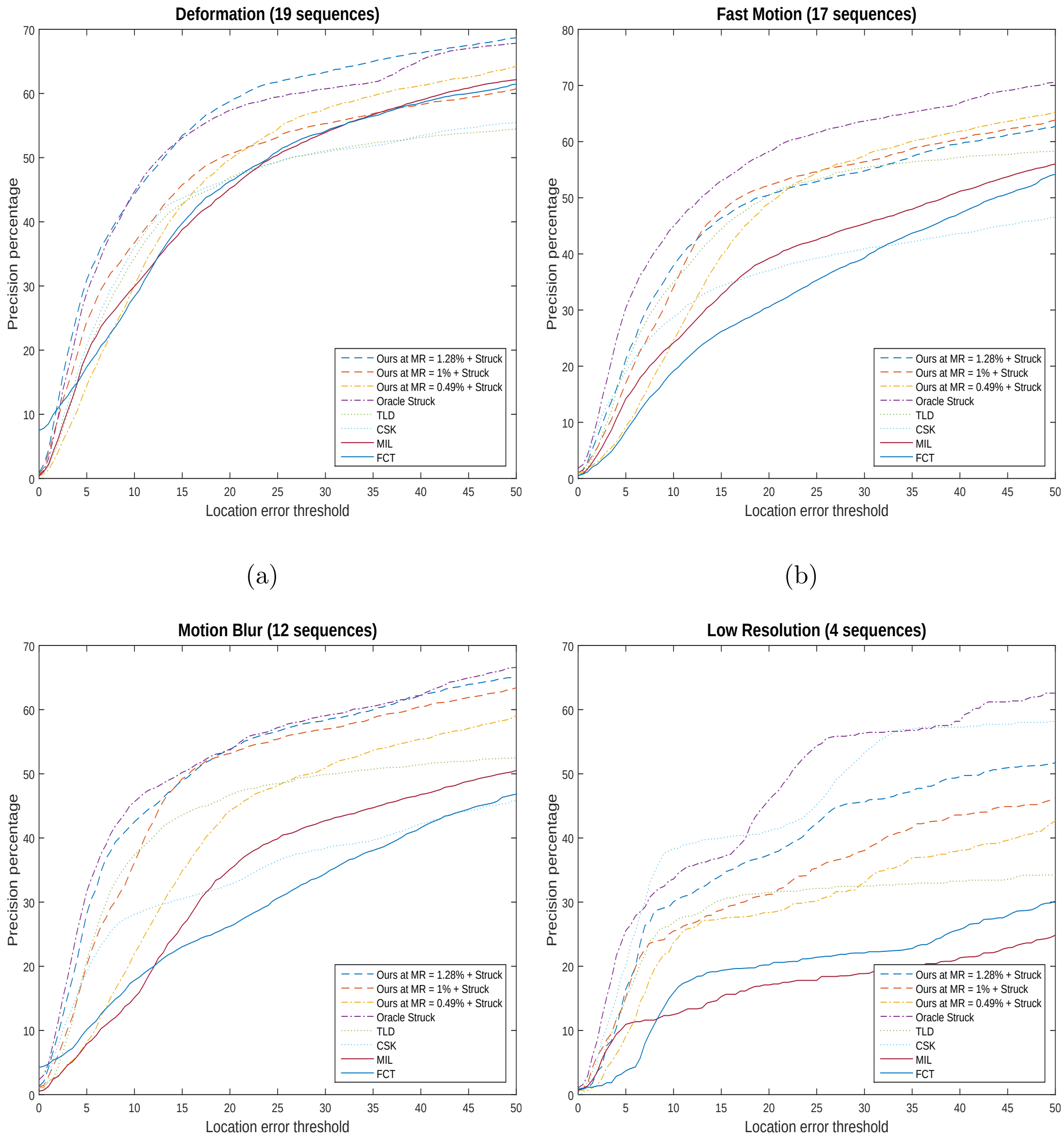
(a)

(b)

(c)

(d)

Figure 4.6: Precision plots for four different attributes. In the case of 'Illumination Variation' and 'Occlusion' 'ReFInE+Struck' at measurement rate of 1.28% performs better than TLD, CSK, MIL and FCT, whereas in the case of the 'Background Clutter' and 'Scale Variation' attributes, TLD performs slightly better than 'ReFInE+Struck' at measurement rate of 1.28%.



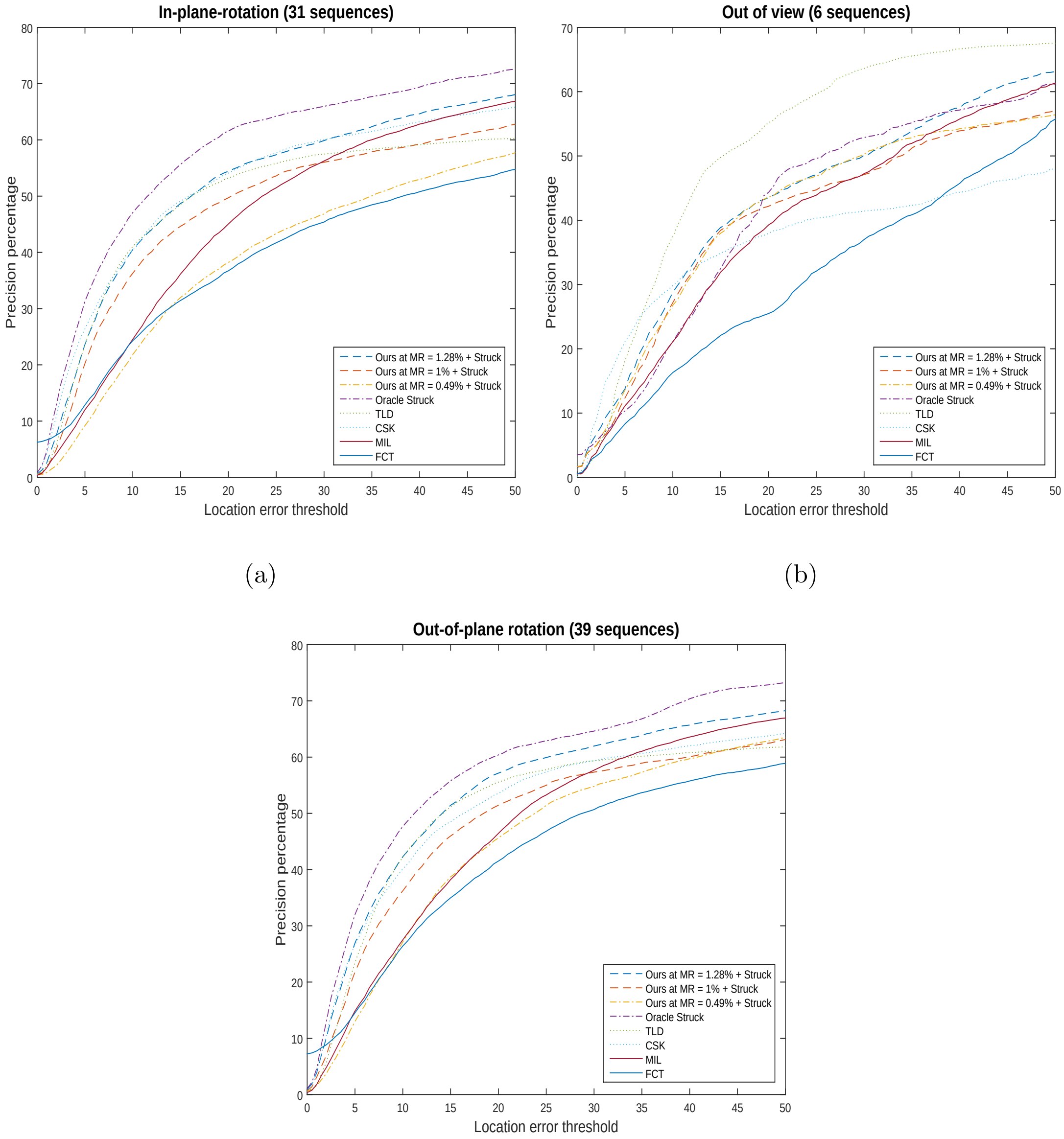
(a)

(b)

(c)

(d)

Figure 4.7: Precision plots for four different attributes. In the cases of 'Deformation', 'Fast Motion' and 'Motion Blur', 'ReFInE+Struck' at measurement rate of 1.28% performs better than TLD, CSK, MIL and FCT, whereas in the case of 'Low Resolution', TLD performs better than 'ReFInE+Struck'.



(a)

(b)

(c)

Figure 4.8: Precision plots for three different attributes. In the cases of 'In the plane rotation', and 'Out of plane rotation', 'ReFInE+Struck' at measurement rate of 1.28% performs better than TLD, CSK, MIL and FCT, whereas in the case of 'Out of View', TLD performs better than 'ReFInE+Struck'.

Tracker	Mean Precision	Mean FPS
SR + ReFInE at EMR = 8.16% + FCT	54.83	19.61
SR + ReFInE at EMR = 4% + FCT	53.03	19.61
SR + ReFInE at EMR = 2.37% + FCT	50.9	19.62
SR + ReFInE at EMR = 1.63% + FCT	45.79	19.62
Oracle FCT [101]	42.37	34.92

Table 4.3: Mean precision percentage for 20 pixels error and mean frames per second for 'SR + ReFIne + FCT' at various measurement rates are compared with 'Oracle FCT'. Even at extremely low measurement rates, the precision percentages for 'SR + ReFIne + FCT' are better that for 'Oracle FCT' which operates on full-blown original resolution images.

image super resolution is calculated such that the resolution of the longer dimension in the higher resolution frame is at least 1000 pixels. We found that upscale factors varies between 2 and 8 for various videos in the dataset. Once the high resolution videos are obtained, we proceed to obtain ReFInE measurements as before. We conduct tracking experiments at four different measurement rate (1%, 0.49%, 0.29%, 0.2%). Note that these different measurement rates are with respect to (wrt) the high-resolution frames, and the measurement rate wrt original resolution, which we call effective measurement rate (EMR), is given by the ratio of the number of ReFInE measurements per frame to the number of pixels in a frame of original resolution video. Here, the tracking algorithm, Struck which we used for original resolution videos does not scale well in terms of computational complexity. For higher resolution videos, where the search space is much larger, we found that Struck is too slow for real-time application. Instead, we use a faster Haar feature based tracking algorithm, FCT [101] algorithm. Henceforth, we dub this tracking pipeline as **SR+ReFInE+FCT**. Once tracking results are obtained for the high resolution videos are obtained, we normalize the coordinates so as to obtain the tracking outputs with respect to original resolution videos. The precision score is calculated as before. The mean precision percentage for 20 pixels error and mean frames per second for 'SR + ReFIne + FCT' at various measurement rates are given in table 4.3 and are compared for the same for 'Oracle FCT' which operates for full-blown original resolution videos. It is clear that we obtain a significant boost in tracking accuracy for high resolution videos. At measurement rate of 1% (EMR of 8.16%), we obtain a mean precision percentage of 54.83, which is 12.46 percentage points more than that for 'Oracle FCT'. Even at a measurement rate of 0.2% ((EMR of 1.63%)), the precision percentage of 45.79, which is about 3.42 percentage points more than that for 'Oracle FCT'. However, the more accurate precision comes at the cost of frame rate. Since the search space is much larger for high resolution videos, the speed of tracking for high resolution videos, is only about 20 FPS, while 'Oracle FCT' operates at 34.92 FPS. But 20 FPS suffices for near real-time implementations.

Chapter 5

DISCUSSIONS AND FUTURE WORK

In this dissertation, we proposed principled ways to extract information from spatial-multiplexing measurements for computer vision applications at low measurement rates.

First, we proposed a correlation based framework to recognize actions from compressive cameras without reconstructing the sequences. It is worth emphasizing that the goal of the work is not to outperform a state-of-the-art action recognition system but is to build a action recognition system which can perform with an acceptable level of accuracy in heavily resource-constrained environments, both in terms of storage and computation. The fact that we are able to achieve a recognition rate of 54.55% at a compression ratio of 100 on a difficult and large dataset like UCF50 and also localize the actions reasonably well clearly buttresses the applicability and the scalability of reconstruction-free recognition in resource constrained environments. Further, we reiterate that at compression ratios of 100 and above, when reconstruction is generally of low quality, action recognition results using our approach, while working in compressed domain, were shown to be far better than reconstructing the images, and then applying a state-of-the-art method. In our future research, we wish to extend this approach to more generalizable filter-based approaches. One possible extension is to use motion sensitive filters like Gabor or Gaussian derivative filters which have proven to be successful in capturing motion. Furthermore, by theoretically proving that a single filter is sufficient to encode an action over the space of all affine transformed views of the action, we showed that more robust filters can be designed by transforming all training examples to a canonical viewpoint.

Next, we have presented a CNN-based non-iterative solution to the problem of CS image reconstruction. We showed that our algorithm provides high quality reconstructions on both simulated and real data for a wide range of measurement rates. Through a proof of concept real-time tracking application at the very low measurement rate of 0.01, we demonstrated the possibility of CS imaging becoming a resource-efficient solution in applications where the final goal is high-level image understanding rather than exact reconstruction. However, the existing CS imagers are not capable of delivering real-time video. We hope that this work will give the much needed impetus to building of more practical and faster video CS imagers.

Next, we qualitatively and quantitatively showed that it is possible obtain high quality estimates of integral images and box-filtered outputs directly from a small number of specially designed spatially multiplexed measurements called **ReFInE** measurements. To show the practical applicability of the integral image estimates, we presented impressive reconstruction-free tracking results on challenging videos at an extremely low measurement rate of 1%. We also showed that with only a small number of **ReFInE** measurements on high-resolution videos, which is only a fraction (2-8%) of the size of the original resolution, one can obtain significantly better object tracking results than using full blown original resolution videos. From a philosophical point of view, this points to the possibility of attaining greater performance on other computer vision inference tasks from a small number of carefully tailored spatially multiplexed measurements of high-resolution imagery rather than full-blown low resolution imagery.

BIBLIOGRAPHY

[1] Jake K Aggarwal and Michael S Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011.

[2] Saad Ali and Simon Lucey. Are correlation filters useful for human action recognition? In *Intl. Conf. Pattern Recog*, 2010.

[3] David F Andrews and Colin L Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 99–102, 1974.

[4] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Robust object tracking with online multiple instance learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(8):1619–1632, 2011.

[5] Richard G Baraniuk, Volkan Cevher, Marco F Duarte, and Chinmay Hegde. Model-based compressive sensing. *IEEE Trans. Inf. Theory*, 56(4):1982–2001, 2010.

[6] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *IEEE Intl. Conf. Comp. Vision.*, 2005.

[7] RN Bracewell, K-Y Chang, AK Jha, and Y-H Wang. Affine theorem for two-dimensional Fourier transform. *Electronics Letters*, 29(3):304, 1993.

[8] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[9] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory*, 52(2):489–509, 2006.

[10] Emmanuel J Candes, Thomas Strohmer, and Vladislav Voroninski. Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming. *Communications on Pure and Applied Mathematics*, 66(8):1241–1274, 2013.

[11] Emmanuel J Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Inf. Theory*, 52(12):5406–5425, 2006.

[12] Emmanuel J. Candes and Michael B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, pages 21 – 30, 2008.

[13] V. Cevher, A. C. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa. Compressive sensing for background subtraction. In *Euro. Conf. Comp. Vision*, 2008.

[14] Hyun Sung Chang, Yair Weiss, and William T Freeman. Informative sensing. *arXiv preprint arXiv:0901.4275*, 2009.

[15] Rizwan Chaudhry, Avinash Ravichandran, Gregory Hager, and René Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *IEEE Conf. Comp. Vision and Pattern Recog*, 2009.

[16] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Process.*, 16(8):2080–2095, 2007.

[17] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Conf. Comp. Vision and Pattern Recog*, pages 886–893. IEEE, 2005.

[18] Konstantinos G. Derpanis, Mikhail Sizintsev, Kevin J. Cannons, and Richard P. Wildes. Efficient action spotting based on a spacetime oriented structure representation. In *IEEE Conf. Comp. Vision and Pattern Recog*, 2010.

[19] Achlioptas. Dimitris. Database-friendly random projections. *Proc. ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems*, pages 274–281, 2001.

[20] Piotr Dollár, Serge Belongie, and Pietro Perona. The fastest pedestrian detector in the west. In *British Machine Vision Conf.*, volume 2, page 7, 2010.

[21] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Euro. Conf. Comp. Vision*, pages 184–199. Springer, 2014.

[22] Weisheng Dong, Guangming Shi, Xin Li, Yi Ma, and Feng Huang. Compressive sensing via nonlocal low-rank regularization. *Image Processing, IEEE Transactions on*, 23(8):3618–3632, 2014.

[23] David L Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, 2006.

[24] David L Donoho, Arian Maleki, and Andrea Montanari. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, 2009.

[25] Marco F Duarte, Michael B Wakin, and Richard G Baraniuk. Wavelet-domain compressive signal reconstruction using a hidden markov tree model. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 5137–5140. IEEE, 2008.

[26] Julio Martin Duarte-Carvajalino and Guillermo Sapiro. Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization. *IEEE Transactions on Image Processing*, 18(7):1395–1408, 2009.

[27] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Adv. Neural Inf. Proc. Sys.*, pages 2366–2374, 2014.

[28] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[29] Gabriel Frahm. *Generalized elliptical distributions: theory and applications.* PhD thesis, Universität zu Köln, 2004.

[30] Masao Fukushima, Zhi-Quan Luo, and Paul Tseng. Smoothing functions for second-order-cone complementarity problems. *SIAM Journal on optimization*, 12(2):436–460, 2002.

[31] Lu Gan. Block compressed sensing of natural images. In *Digital Signal Processing, 2007 15th International Conference on*, pages 403–406. IEEE, 2007.

[32] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conf. Comp. Vision and Pattern Recog*, pages 580–587. IEEE, 2014.

[33] Tom Goldstein, Lina Xu, Kevin F Kelly, and Richard Baraniuk. The stone transform: Multi-resolution image enhancement and real-time compressive video. *arXiv preprint arXiv:1311.3405*, 2013.

[34] Helmut Grabner, Michael Grabner, and Horst Bischof. Real-time tracking via on-line boosting. In *British Machine Vision Conf.*, 2006.

[35] Sam Hare, Amir Saffari, and Philip HS Torr. Struck: Structured output tracking with kernels. In *IEEE Intl. Conf. Comp. Vision.*, pages 263–270. IEEE, 2011.

[36] Chinmay Hegde, Aswin C Sankaranarayanan, Wotao Yin, and Richard G Baraniuk. Numax: A convex approach for learning near-isometric linear embeddings. *IEEE Transactions on Signal Processing*, 63(22):6109–6121, 2015.

[37] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *Euro. Conf. Comp. Vision*, pages 702–715, 2012.

[38] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):583–596, 2015.

[39] Arpit Jain, Abhinav Gupta, Mikel Rodriguez, and Larry S Davis. Representing videos using mid-level discriminative patches. In *IEEE Conf. Comp. Vision and Pattern Recog.* IEEE, 2013.

[40] Mihir Jain, Hervé Jégou, and Patrick Bouthemy. Better exploiting motion for better action recognition. In *IEEE Conf. Comp. Vision and Pattern Recog.* IEEE, 2013.

[41] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[42] Yu-Gang Jiang, Qi Dai, Xiangyang Xue, Wei Liu, and Chong-Wah Ngo. Trajectory-based modeling of human actions with motion reference points. In *Euro. Conf. Comp. Vision*. Springer, 2012.

[43] W. B. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Conference in Modern Analysis and Probability (New Haven, Conn.)*, 1982.

[44] I.N. Junejo, E. Dexter, I. Laptev, and P. Perez. View-independent action recognition from temporal self-similarities. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):172 –185, 2011.

[45] K. Kulkarni and P. Turaga. Recurrence textures for activity recognition using compressive cameras. In *IEEE Conf. Image Process.*, 2012.

[46] Zdenek Kalal, Jiri Matas, and Krystian Mikolajczyk. Pn learning: Bootstrapping binary classifiers by structural constraints. In *IEEE Conf. Comp. Vision and Pattern Recog*, pages 49–56, 2010.

[47] Ronan Kerviche, Nan Zhu, and Amit Ashok. Information optimal scalable compressive imager demonstrator. In *IEEE Conf. Image Process.*, 2014.

[48] Ronan Kerviche, Nan Zhu, and Amit Ashok. Information-optimal scalable compressive imaging system. In *Classical Optics 2014*. Optical Society of America, 2014.

[49] Tae-Kyun Kim and Roberto Cipolla. Canonical correlation analysis of video volume tensors for action categorization and detection. 31(8):1415–1428, 2009.

[50] Yookyung Kim, Mariappan S Nadar, and Ali Bilgin. Compressed sensing using a gaussian scale mixtures model in wavelet domain. In *IEEE Conf. Image Process.*, pages 3365–3368. IEEE, 2010.

[51] Alexander Klaser and Marcin Marszalek. A spatio-temporal descriptor based on 3d-gradients. In *British Machine Vision Conf.*, 2008.

[52] Orit Kliper-Gross, Yaron Gurovich, Tal Hassner, and Lior Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *Euro. Conf. Comp. Vision*. Springer, 2012.

[53] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Adv. Neural Inf. Proc. Sys.*, pages 1097–1105, 2012.

[54] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *IEEE Intl. Conf. Comp. Vision.*, pages 2556–2563, 2011.

[55] K. Kulkarni and P. Turaga. Reconstruction-free action inference from compressive imagers. *IEEE Trans. Pattern Anal. Mach. Intell.*, PP(99), 2015.

[56] Tian Lan, Yang Wang, and Greg Mori. Discriminative figure-centric models for joint action localization and recognition. In *IEEE Intl. Conf. Comp. Vision.*, 2011.

[57] Chengbo Li, Wotao Yin, Hong Jiang, and Yin Zhang. An efficient augmented lagrangian method with applications to total variation minimization. *Computational Optimization and Applications*, 56(3):507–530, 2013.

[58] Yong-Jin Liu, Defeng Sun, and Kim-Chuan Toh. An implementable proximal point algorithmic framework for nuclear norm minimization. *Mathematical programming*, 133(1-2):399–436, 2012.

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

[60] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conf. Comp. Vision and Pattern Recog*, June 2015.

[61] Michael Lustig, David Donoho, and John M Pauly. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic resonance in medicine*, 58(6):1182–1195, 2007.

[62] Siwei Lyu and Eero P Simoncelli. Modeling multiscale subbands of photographic images with fields of gaussian scale mixtures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(4):693–706, 2009.

[63] M. A. Davenport, M. F. Duarte, M. B. Wakin, J. N. Laska, D. Takhar, K. F. Kelly and R. G. Baraniuk. The smashed filter for compressive classification and target recognition. pages 6498–6499, 2007.

[64] Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(7):674–693, 1989.

[65] M.B. Wakin, J.N. Laska, M.F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K.F. Kelly and R.G. Baraniuk. An architecture for compressive imaging. In *IEEE Conf. Image Process.*, 2006.

[66] Christopher A Metzler, Arian Maleki, and Richard G Baraniuk. From denoising to compressed sensing. *arXiv preprint arXiv:1406.4175*, 2014.

[67] Ali Mousavi, Ankit B Patel, and Richard G Baraniuk. A deep learning approach to structured signal recovery. In *Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on*, pages 1336–1343. IEEE, 2015.

[68] Shree K Nayar, Vlad Branzoi, and Terry E Boult. Programmable imaging: Towards a flexible camera. *Intl. J. Comp. Vision*, 70(1):7–22, 2006.

[69] Deanna Needell and Joel A Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

[70] Deanna Needell and Joel A Tropp. Cosamp: iterative signal recovery from incomplete and inaccurate samples. *Communications of the ACM*, 53(12):93–100, 2010.

[71] B. Ozer, W. Wolf, and A. N. Akansu. Human activity detection in MPEG sequences. In *Proceedings of the Workshop on Human Motion (HUMO'00)*, HUMO '00, pages 61–66. IEEE Computer Society, 2000.

[72] Gabriel Peyré, Sébastien Bougleux, and Laurent Cohen. Non-local regularization of inverse problems. In *Euro. Conf. Comp. Vision*, pages 57–68. Springer, 2008.

[73] P.H Hennings-Yeoman, B.V.K.V Kumar and M. Savvides. Palmprint classification using multiple advanced correlation filters and palm-specific segmentation. *IEEE Trans. on Information Forensics and Security*, 2(3):613–622, 2007.

[74] R. Calderbank, S. Jafarpour and R. Schapire. Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain. In *Preprint*, 2009.

[75] S Ramakanth and R Babu. Seamseg: Video object segmentation using patch seams. In *IEEE Conf. Comp. Vision and Pattern Recog*, pages 376–383, 2013.

[76] Kishore K Reddy and Mubarak Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24(5):971–981, 2013.

[77] Mikel D. Rodriguez, Javed Ahmed, and Mubarak Shah. Action MACH: a spatio-temporal maximum average correlation height filter for action recognition. In *IEEE Conf. Comp. Vision and Pattern Recog*, 2008.

[78] S. Sadanand, J. J. Corso. Action bank: A high-level representation of activity in video. In *IEEE Conf. Comp. Vision and Pattern Recog*, 2012.

[79] A. C. Sankaranarayanan, P. Turaga, R. Baraniuk, and R. Chellappa. Compressive acquisition of dynamic scenes. In *Euro. Conf. Comp. Vision*, 2010.

[80] Aswin C Sankaranarayanan, Christoph Studer, and Richard G Baraniuk. Cs-muvi: Video compressive sensing for spatial-multiplexing cameras. In *Computational Photography (ICCP), 2012 IEEE International Conference on*, pages 1–10. IEEE, 2012.

[81] Hae Jong Seo and Peyman Milanfar. Action recognition from one example. 33(5):867–882, 2011.

[82] Eli Shechtman and Michal Irani. Space-time behavior based correlation. In *IEEE Conf. Comp. Vision and Pattern Recog.* IEEE, 2005.

[83] Feng Shi, Emil Petriu, and Robert Laganiere. Sampling strategies for real-time action recognition. In *IEEE Conf. Comp. Vision and Pattern Recog.* IEEE, 2013.

[84] S Sims and Abhijit Mahalanobis. Performance evaluation of quadratic correlation filters for target detection and discrimination in infrared imagery. *Optical Engineering*, 43(8):1705–1711, 2004.

[85] Subhojit Som and Philip Schniter. Compressive imaging using approximate message passing and a markov-tree prior. *Signal Processing, IEEE Transactions on*, 60(7):3439–3448, 2012.

[86] Dharmpal Takhar, Jason N Laska, Michael B Wakin, Marco F Duarte, Dror Baron, Shriram Sarvotham, Kevin F Kelly, and Richard G Baraniuk. A new compressive imaging camera architecture using optical-domain compression. In *Electronic Imaging 2006.* International Society for Optics and Photonics, 2006.

[87] Jin Tan, Yanting Ma, and Dror Baron. Compressive imaging via approximate message passing with image denoising. *Signal Processing, IEEE Transactions on*, 63(8):2085–2092, 2015.

[88] Vijayaraghavan Thirumalai and Pascal Frossard. Correlation estimation from compressed images. *J. Visual Communication and Image Representation*, 24(6):649–660, 2013.

[89] Yicong Tian, Rahul Sukthankar, and Mubarak Shah. Spatiotemporal deformable part models for action detection. In *IEEE Conf. Comp. Vision and Pattern Recog*, 2013.

[90] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inf. Theory*, 53(12):4655–4666, 2007.

[91] Paul Viola and Michael J Jones. Robust real-time face detection. *Intl. J. Comp. Vision*, 57(2):137–154, 2004.

[92] Martin J Wainwright and Eero P Simoncelli. Scale mixtures of gaussians and the statistics of natural images. In *Adv. Neural Inf. Proc. Sys.*, pages 855–861, 1999.

[93] Jacob Walker, Abhinav Gupta, and Martial Hebert. Dense optical flow prediction from a static image. *arXiv preprint arXiv:1505.00295*, 2015.

[94] Heng Wang, Alexander Klaser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *IEEE Conf. Comp. Vision and Pattern Recog.* IEEE, 2011.

[95] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *IEEE Intl. Conf. Comp. Vision.*, pages 3551–3558. IEEE, 2013.

[96] X. Wang, David F. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *IEEE Conf. Comp. Vision and Pattern Recog*, 2015.

[97] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *IEEE Conf. Comp. Vision and Pattern Recog*, pages 2411–2418. IEEE, 2013.

[98] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(9):1834–1848, 2015.

[99] Chuohao Yeo, Parvez Ahammad, Kannan Ramchandran, and S Shankar Sastry. High speed action recognition and localization in compressed domain videos. *IEEE Trans. Cir. and Sys. for Video Technol.*, 18(8):1006–1015, 2008.

[100] Jian Zhang, Shaohui Liu, Ruiqin Xiong, Siwei Ma, and Debin Zhao. Improved total variation based image compressive sensing recovery by nonlocal regularization. In *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, pages 2836–2839. IEEE, 2013.

[101] Kaihua Zhang, Lei Zhang, and M Yang. Fast compressive tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(10):2002–2015, 2014.

[102] Kaihua Zhang, Lei Zhang, and Ming-Hsuan Yang. Real-time compressive tracking. In *Euro. Conf. Comp. Vision*, pages 864–877, 2012.

[103] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In *Euro. Conf. Comp. Vision*, pages 834–849. Springer, 2014.

[104] Ning Zhang, Manohar Paluri, Marc'Aurelio Ranzato, Trevor Darrell, and Lubomir Bourdev. Panda: Pose aligned networks for deep attribute modeling. In *IEEE Conf. Comp. Vision and Pattern Recog*, pages 1637–1644. IEEE, 2014.

[105] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Adv. Neural Inf. Proc. Sys.*, pages 487–495, 2014.

[106] Xiangxin Zhu, Shengcai Liao, Zhen Lei, Rong Liu, and Stan Z Li. Feature correlation filter for face recognition. In *Advances in Biometrics*, volume 4642, pages 77–86. Springer, 2007.