ABSTRACT

Title of dissertation: SPARSE ACQUISITION AND RECONSTRUCTION FOR SOME COMPUTER VISION PROBLEMS

Nagilla Dikpal Reddy, Doctor of Philosophy, 2011

Dissertation directed by: Professor Rama Chellappa
Department of Electrical and Computer Engineering

Sparse representation, acquisition and reconstruction of signals guided by theory of Compressive Sensing (CS) has become an active research research topic over the last few years. Sparse representations effectively capture the idea of parsimony enabling novel acquisition schemes including sub-Nyquist sampling. Ideas from CS have had significant impact on well established fields such as signal acquisition, machine learning and statistics and have also inspired new areas of research such as low rank matrix completion. In this dissertation we apply CS ideas to low-level computer vision problems. The contribution of this dissertation is to show that CS theory is an important addition to the existing computational toolbox in computer vision and pattern recognition, particularly in data representation and processing. Additionally, in each of the problems we show how sparse representation helps in improved modeling of the underlying data leading to novel applications and better understanding of existing problems.

In our work, the impact of CS is most felt in the acquisition of videos with novel camera designs. We build prototype cameras with slow sensors capable of

capturing at an order of magnitude higher temporal resolution. First, we propose sub-Nyquist acquisition of periodic events and then generalize the idea to capturing regular events. Both the cameras operate by first acquiring the video at a slower rate and then computationally recovering the desired higher temporal resolution frames. In our camera, we sense the light with a slow sensor after modulating it with a fluttering shutter and then reconstruct the high speed video by enforcing its sparsity. Our cameras offer a significant advantage in light efficiency and cost by obviating the need to sense, transfer and store data at a higher frame rate.

Next, we explore the applicability of compressive cameras for computer vision applications in bandwidth constrained scenarios. We design a compressive camera capable of capturing video using fewer measurements and also separate the foreground from the background. We model surveillance type videos with two processes, a slower background and a faster but spatially sparse foreground such that we can recover both of them separately and accurately. By formulating the problem in a distributed CS framework we achieve state-of-the-art video reconstruction and background subtraction. Subsequently we show that if the camera geometry is provided in a multi-camera setting, the background subtracted CS images can be used for localizing the object and tracking it by formulating its occupancy in a grid as a sparse reconstruction problem.

Finally, we apply CS to robust estimation of gradients obtained through photometric stereo and other gradient-based techniques. Since gradient fields are often not integrable, the errors in them need to be estimated and removed. By assuming the errors, particularly the outliers, as sparse in number we accurately estimate

and remove them. Using conditions on sparse recovery in CS we characterize the distribution of errors which can be corrected completely and those that can be only partially corrected. We show that our approach has the important property of localizing the effect of error during integration where other parts of the surface are not affected by errors in gradients at a particular location.

This dissertation is one of the earliest to investigate the implications of compressive sensing theory to some computer vision problems. We hope that this effort will spur more interest in researchers drawn from computer vision, computer graphics, computational photography, statistics and mathematics.

SPARSE ACQUISITION AND RECONSTRUCTION FOR SOME COMPUTER VISION PROBLEMS

by

Nagilla Dikpal Reddy

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2011

Advisory Committee:
Professor Rama Chellappa, Chair/Advisor
Professor Larry Davis
Professor Min Wu
Professor K. J. Ray Liu
Professor David Jacobs
Professor Ashok Veeraraghavan

Dedication

Dedicated to my *parents*.

Acknowledgments

This dissertation would not have been possible without the encouragement and support of many people, to whom I express my deepest gratitude.

Foremost I would like to thank my advisor Prof. Rama Chellappa who provided me the opportunity, freedom, encouragement and resources to pursue my dissertation. I have learnt much from Prof. Chellappa about research and life. The cheerfulness and energy with which Prof. Chellappa approaches all the tasks is infectious. I always left his office with more energy than what I entered with. I am always amazed by the balanced and broad perspective which Prof. Chellappa brings to research, teaching and interaction with people. For example, I was fortunate to have experienced it as a teaching assistant for the Pattern Recognition course in Fall 2009. As a TA with no burden of learning the course, I appreciated the way Prof. Chellappa seamlessly connected the old topics with the new, reached out to the class with his humor and made it an enjoyable experience for everyone. I am also grateful to him for his magnanimity. Prof. Chellappa encouraged and provided me the freedom to pursue a big portion of my dissertation at Mitsubishi Electric Research Labs (MERL). He also ensured that I always have the necessary intellectual room for conducting research without financial worries.

I am grateful to Prof. Ashok Veeraraghavan for his mentorship and for providing me the opportunity to pursue exciting problems at MERL. Working with Ashok I learnt about various aspects of conducting research, from picking problem areas to designing experiments to validate one's intuition. Importantly Ashok and

MERL taught me how ideas can be translated into prototypes, something which I never imagined doing as a part of my dissertation. Thanks are due to MERL for providing an excellent academic environment and for the financial support during my visits. I thank Dr. Chris Wren, Dr. Amit Agrawal, Dr Aswin Sankaranarayanan and Dr. Zhanfeng Yue for mentoring me at various times during the course of my PhD.

I would also like to thank Prof. Min Wu, Prof. Larry Davis, Prof. K.J. Ray Liu and Prof. David Jacobs for serving on my dissertation committee and providing valuable feedback. The thesis wouldn't have taken the shape without the feedback, encouragement and support of my fellow group members. I particularly thank Aswin Sankaranarayanan, Pavan Turaga, Kaushik Mitra, Raghuraman Gopalan, Nitesh Shroff and Ming-Yu Liu. Things would come to a standstill at UMD without the crucial support provided by Janice Perrone, Arlene Schenk, members of ECE staff, UMIACS computing staff and OIS staff. A big thanks to them.

During my stay at UMD I seldom felt I was away from home. Anyone would feel so with the support of wonderful friends I was fortunate to have. I thank Avinash Varna, Bhargav Kanagal, Suriyanarayanan Vaikuntanathan, Baladitya Suri, Varada Shevade, Aditya Ramanathan, Nitesh Shroff and Ming-Yu Liu. I thank my roommates through the years Anand Ramanathan, Arvind Ananthanarayanan, Praveen Narayanan, Vishnu Arcot, Kedar Dimble, Ashwin Murali and Amar Setty for maintaining a comfortable academic environment at home. I also thank the student organizations ECEGSA and DESI for providing me with an enriching and balanced perspective outside my research life.

Words will not be enough to express my gratitude to my mother, father and brother. They have been the foundation of my life and have sacrificed much for me. This is as much their dissertation as it is mine.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Modeling data and representing signals as a linear combination of basis elements is a simple and popular way of describing a signal using its constituent parts. For example, the Fourier transform of a signal gives us the weights with which the sinusoidal basis functions should be combined. Similarly, given data samples the singular value decomposition (SVD) provides us with orthogonal basis elements and their weights. Nevertheless, most of the energy in the signal is often concentrated at few basis elements.

The simplest way to characterize this redundancy is to approximate the data with a subspace of the space spanned by all the basis elements. For instance, natural signals have a Fourier transform which rapidly falls off with frequency leading to a baseband bandwidth which is a fraction of the entire spectrum. Similarly, the principal component analysis (PCA) of data samples reveals a pattern where most of the energy is concentrated at principal components. Thus subspace representation of signals captures the energy distribution but it doesn't always explain the underlying redundancy. For example, a signal may have a large baseband bandwidth but only a small bandwidth. Also, to explain the redundancy in a signal, a different set of basis elements might be required such as Wavelets for image decomposition. In such

a decomposition the energy maybe distributed sparse and non-contiguously.

Often over-complete representations (such as the union of Fourier and canonical basis) are used to capture the underlying redundancy leading to a non-unique representation. Unlike subspace representation, the ensemble of signals here are represented not by a single subspace but a union-of-subspaces. To represent a signal with a subspace from the union requires solving computationally intense optimization problems with sparsity constraints. The guarantees for unique sparse decomposition of signals given an over-complete representation have been found only recently. This resulted in great excitement and fervent research in the development of theory and applications of sparse representations.

Under the umbrella term 'Compressive Sensing' (CS), the sparse representation/acquisition theory has had significant impact on many problems in signal processing, statistics and machine learning with applications in many other. For instance, CS questions the traditional Nyquist sampling theory and proposes novel sub-Nyquist sampling schemes. In the appendix, we briefly describe the CS theory.

In this dissertation we apply CS theory to low-level computer vision problems. Particularly, we focus on novel acquisition of videos leading to prototype compressive cameras which can capture at higher temporal resolution even at lower frame rate. We also investigate the potential application of compressive cameras in low-level vision tasks in bandwidth constrained scenarios. Further, we apply it to traditional gradient based surface reconstruction algorithms. The contribution of this dissertation is to show that sparse representation and reconstruction algorithms are important tools which can improve the state-of-the-art in many applications in

computer vision with potential to define new research areas. Next, we provide an overview of the dissertation and briefly describe its component chapters.

1.1 Coded strobing photography

We propose a camera that can capture high speed periodic phenomenon with a slow sensor at sub-Nyquist rates. We do so by strobing the scene during the exposure duration of the sensor according to a pre-determined pseudo random code, thereby preserving the high frequency information from being lost due to blur. Later, we computationally recover the fast phenomenon by enforcing its sparsity in Fourier transform. We have built a prototype 25fps camera capable of capturing at 2000fps

Strobing is a traditional passive/active way of visualizing/capturing high speed periodic visual signals using a low frame rate camera. It relies on flashing a bright source of light (strobe) periodically for a short duration to illuminate the scene. By keeping the period of the strobe to be near the signal's period, beat frequencies are generated which allows visualization of the signal. Although this approach is widely used in medical imaging and industrial settings, it is very light inefficient. Moreover, the period of the visual signal needs to be known a priori. Instead, coded strobing is light efficient and doesn't need a priori knowledge of the period. Also, it allows us to capture multiple periodic signals of with different periods.

Each frame of our camera captures the modulation of coded strobe and the visual signal. Since the modulation matrix is known, we can invert the observed coded frames by enforcing sparsity of the periodic signal using techniques from CS

literature. Our approach is also applicable to sparse band-limited signals including quasi-periodic signals.

1.2 Programmable pixel compressive camera

We extend the idea of coded strobing to handle regular motion by designing and building programmable pixel compressive camera (P2C2). Our camera captures videos at a higher temporal resolution without blur, loss in spatial resolution and any additional cameras. We generalize the model of videos to handle a broad class of motions and at the same time generalize the acquisition by building a per-pixel shutter. While sparse representations model images accurately, they need to be augmented in videos by treating the spatial and temporal correlations separately. To handle a broad class of motion patterns and occlusions, we enforce sparsity in each high-speed frame but enforce brightness constancy constraints temporally.

To prevent the temporal information from being lost through integration of light during exposure of each frame, we code each pixel independently. This can be interpreted as 'time-stamping' the incoming high speed frames with a unique mask which can be inverted during reconstruction. The per-pixel coding offers improved conditioning of the measurement matrix compared to the fluttered global shutter. We have built a prototype camera (25fps) with a liquid crystal on silicon (LCOS) modulating device for testing and show an order of magnitude improvement in temporal resolution (200fps). Per-pixel control of shutter offers significant control over the sensor and can be used for other applications as well. For instance, per-

pixel coding can be used to capture high dynamic range (HDR) images and also for implementing programmable sensors.

1.3 Compressive background substraction and tracking

We propose to use compressive cameras for surveillance applications involving computer vision tasks such as detection and tracking. Compressive cameras by virtue of sensing less data, are particularly useful in bandwidth constrained scenarios. Compressive cameras also offer the advantage of using fewer/cheaper sensors (e.g. Single Pixel Camera [45]) allowing significant savings in hyper-spectral and fast imaging. For example, in a camera (say hyper-spectral) network with bandwidth limitations it would be prudent to sense compressively and then transmit to a central location for subsequent processing.

We propose a background subtraction algorithm on compressive frames without the need for reconstructing the individual frames. Since, the compressed measurements are a linear projection of the scene, the statistics of the background model extend in a straightforward way from the typical background subtraction algorithms. Typically, the foreground pixel statistics are different from that of background and this holds in compressed measurements as well. Since the foreground image is typically sparse, we show that we can perform background subtraction with even fewer measurements. Our algorithm is robust to both slow and fast variations in the background.

Next, we show that tracking of objects can be performed in a multi-camera

setting on compressed measurements. The problem of tracking is formulated as that of sparse estimation where the object being tracked occupies sparse support on a localization grid on the ground plane. Given the homography between the ground plane and the cameras, the appearance of the scene corresponding to the grid points on the ground plane can be mapped to pixels on the cameras. By collecting random projections of these pixels at multiple cameras, the position of the object is estimated centrally by first estimating the background subtracted image at each pixel and then solving the position using homography from all cameras. We show that our approach is scalable in number of cameras and the computation depends only on the size of the grid.

1.4 Joint compressive video sensing and background subtraction

In the background subtraction approach mentioned above, the background information is completely lost. This leads to the loss of background context in which the object is moving. We propose to reconstruct both foreground and background without any additional measurements by exploiting the fact that the background changes slowly compared to the foreground and by modifying the sensing by changing the measurement matrix in each frame. We overcome the drawback of strict sparsity necessary for good quality compressive reconstruction by avoiding the need to enforce sparsity in background. By formulating the problem in a distributed compressive framework, we not only separate the background but also sense it.

1.5 Enforcing integrability

In many applications such as shape from shading, photometric stereo and gradient domain processing, the estimated gradient field is noisy. This renders the gradient field to be non-integrable (non-zero curl) leading to poor surface reconstruction. We treat the problem of enforcing integrability as that of error correction where we correct the errors (noise and outliers). Using sufficient conditions for sparse recovery, we analyze the scenarios under which all the errors can be corrected. We propose minimizing the ℓ_1-norm of gradient error subject to the observed curl values resulting in robust estimation of the gradients. We show through experiments that our approach is significantly better than existing techniques in correcting errors and that even when it fails to correct the errors it doesn't corrupt the reconstructed surface elsewhere.

Chapter 2

Coded Strobing Photography

2.1 Introduction

Periodic signals are all around us. Several human and animal biological processes such as heart-beat, breathing, several cellular processes, industrial automation processes and everyday objects such as hand-mixer and blender all generate periodic processes. Nevertheless, we are mostly unaware of the inner workings of some of these high-speed processes because they occur at a far greater speed than can be perceived by the human eye. Here, we show a simple but effective technique that can turn an off-the-shelf video camera into a powerful high-speed video camera for observing periodic events.

Strobing is often used in entertainment, medical imaging and industrial applications to visualize and capture high-speed visual phenomena. Active strobing involves illuminating the scene with a rapid sequence of flashes within a frame time. The classic example is Edgerton's Rapatron to capture a golf swing [46]. In modern sensors, it is achieved passively by multiple-exposures within a frame time [157][121] or fluttering [124]. We use the term 'strobing' to indicate both active illumination and passive sensor methods.

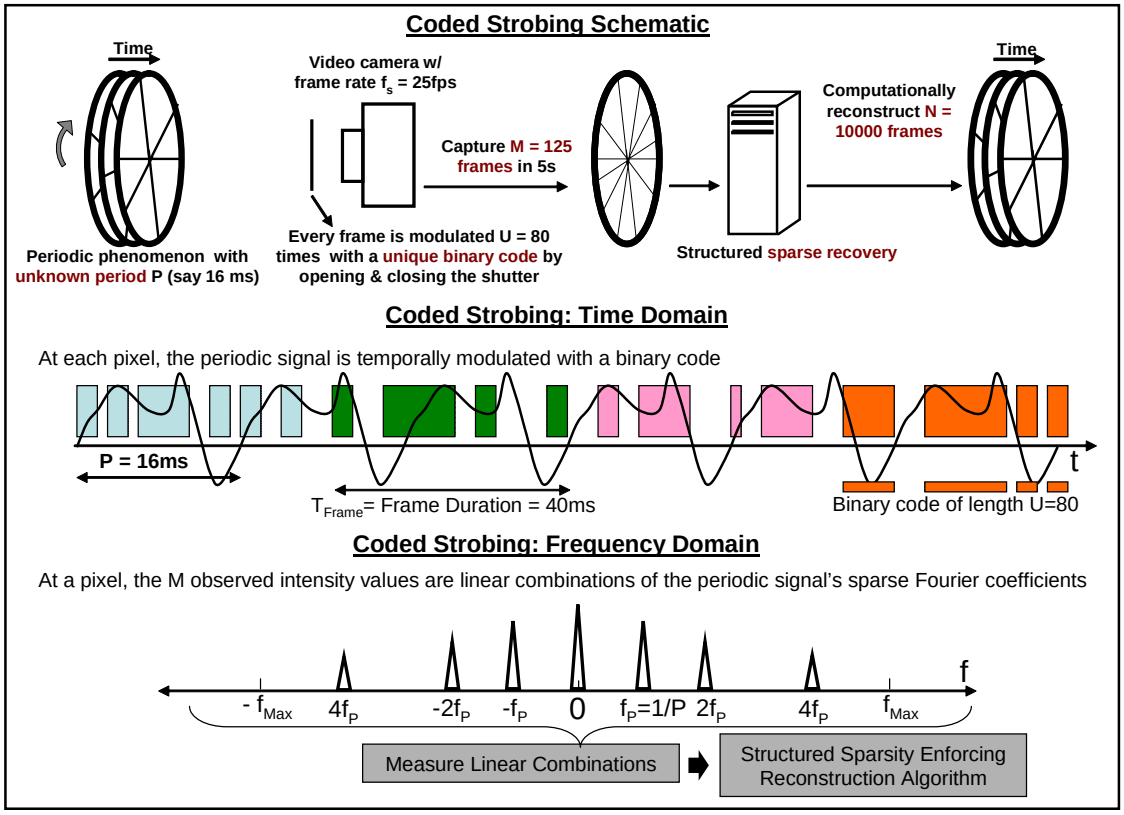

Figure 2.1: CSC: A fast periodic visual phenomenon is recorded by a normal video camera (25 fps) by randomly opening and closing the shutter at high speed (2000 Hz). The phenomenon is accurately reconstructed from the captured frames at the high-speed shutter rate (2000 fps).

In the case of periodic phenomenon, strobing is commonly used to achieve aliasing and generate lower beat frequencies. While strobing performs effectively when the scene consists of a single frequency with a narrow sideband, it is difficult to visualize multiple or a wider band of frequencies simultaneously. Instead of direct observation of beat frequencies, we exploit a computational camera approach based on different sampling sequences. The key idea is to measure appropriate linear combinations of the periodic signal and then decode the signal by exploiting the sparsity of the signal in Fourier domain. We observe that by coding during the exposure duration of a low-frame-rate (e.g., 25 fps) video camera, we can take

appropriate projections of the signal needed to reconstruct a high-frame-rate (e.g., 2000 fps) video. During each frame, we strobe and capture a coded projection of the dynamic event and store the integrated frame. After capturing several frames, we computationally recover the signal independently at each pixel by exploiting the Fourier sparsity of periodic signals. Our method of coded exposure for sampling periodic signals is termed 'coded strobing' and we call our camera the 'coded strobing camera' (CSC). Figure 2.1 illustrates the operation of CSC.

2.1.1 Contributions

- We show that sub-Nyquist sampling of periodic visual signals is possible and that such signals can be captured and recovered using a coded strobing computational camera.
- We develop a sparsity-exploiting reconstruction algorithm and expose connections to compressive sensing.
- We show that the primary benefit of our approach over traditional strobing is, increased light-throughput and the ability to simultaneously tackle multiple frequencies post-capture.

2.1.2 Benefits and limitations

The main constraint for recording a high-speed event is **light throughput**. We overcome this constraint for periodic signals via sufficient exposure duration (in each frame) and extended observation window (multiple frames). For well-lit

non-periodic events, high-speed cameras are ideal. For a static snapshot, a short exposure photo (or single frame of the high-speed camera) is sufficient. In both cases, light throughput is limited but unavoidable. Periodic signals can also be captured with a high-speed camera. But one will need a well-lit scene or must illuminate it with unrealistic bright lights. For example, if we use a 2000 fps camera for vocal cord analysis instead of strobing using a laryngoscope, we would need a significantly brighter illumination source and this creates the risk of burn injuries to the throat. A safer option would be 25 fps camera with strobed light source and then exploit the periodicity of vocal fold movement. Here, we show that an even better option in terms of light-throughput is a computational camera approach. Further, the need to know frequency of the signal at capture-time is also avoided. Moreover, the computational recovery algorithm can tackle the presence of multiple fundamental frequencies in a scene, which poses a challenge to traditional strobing.

2.1.3 Related work

High-speed imaging hardware: Capturing high-speed events with fast, high-frame rate cameras require imagers with high photoresponsivity at short integration times, synchronous exposure and high-speed parallel readout due to the necessary bandwidth. In addition, they suffer from challenging storage problems. A high-speed camera also fails to exploit the inter-frame coherence, while our technique takes advantage of a simplified model of motion. Edgerton [46] and others have shown visually stunning results for high-speed objects using extremely narrow-

duration flash . These snapshots capture an instant of the action but fail to indicate the general movement in the scene. Multiple low-frame rate cameras can be combined to create high-speed sensing. Using a staggered exposure approach, Shechtman et al. [137] used frames captured by multiple co-located cameras with overlapped exposure time. This staggered exposure approach also assisted a novel reconfigurable multi-camera array [159]. Although numerous super-resolution techniques have been proposed to increase the spatial resolution of images, only few methods are available for temporally super-resolving a video [57]. In [65], a super-resolution technique to reconstruct a high-resolution image from a sequence of low-resolution images was proposed using the backprojection method. A method to do super-resolution on a low quality image of a moving object was proposed in [12] by first tracking it, estimating the motion and deblurring the motion blur and creating a high quality image. Freeman et al. [55] proposed a learning-based technique for superresolution from one image where the high frequency components like edges of an image are filled by patches obtained from examples with similar low resolution properties. Finally, fundamental limits on super-resolution for reconstruction based algorithms have been explored in [7][85].

Stroboscopy and periodic motion: Stroboscopes (from the Greek word $\sigma\tau\rho\omega\beta\omega\sigma$ for 'whirling') play an important role in scientific research, to study machinery in motion, in entertainment and medical imaging. Muybridge in his pioneering work used multiple triggered cameras to capture high-speed motion of animals [105] and proved that all four of a horse's hooves left the ground at the same time during a gallop. Edgerton also used flashing lamp to study machine parts in motion

[46]. The most common approaches for "freezing" or "slowing down" the movement are based on temporal aliasing. In medicine, stroboscopes are used to view the vocal cords for diagnosis. The patient hums or speaks into a microphone which in turn activates the stroboscope at either the same or a slightly lower frequency [81],[131]. However, in all healthy humans, vocal-fold vibrations are aperiodic to a greater or lesser degree. Therefore, strobolaryngoscopy does not capture the fine detail of each individual vibratory cycle; rather, it shows a pattern averaged over many successive nonidentical cycles [100][135]. Modern strobocopes for machine inspection [36] are designed for observing fast repeated motions and for determining RPM. The idea can also be used to improve spatial resolution by introducing high-frequency illumination [60].

Processing: In computer vision, periodic motion of humans has received significant attention. Seitz et al. [132] introduced a novel motion representation, called the period trace, that provides a complete description of temporal variations in a cyclic motion, which can be used to detect motion trends and irregularities. A technique to repair videos with large static background or cyclic motion was presented in [69]. Laptev et al. [80] presented a method to detect and segment periodic motion based on sequence alignment without the need for camera stabilization and tracking. [14] exploited periodicity of moving objects to perform 3D reconstruction by treating frames with same phase to be of same pose observed from different views. In [141], the authors showed a strobe based approach for capturing high-speed motion using multiexposure images obtained within a single frame of a camera. The images of a baseball appear as distinct non-overlapping positions in the image . High temporal

and spatial resolution can be obtained via a hybrid imaging device which consists of a high spatial resolution digital camera in conjunction with a high frame-rate but low resolution video camera [16]. In cases where the motion can be modeled as linear, there have been several interesting methods to engineer the motion blur point spread function so that the blur induced by the imaging device is invertible. These include coding the exposure [124] and moving the sensor during the exposure duration [83]. The method presented in this dissertation tackles a somewhat related problem of reconstructing periodic signals from very low-speed images acquired via a conventional video camera (albeit enhanced with coded exposure).

Comparison with flutter shutter: In [124], the authors showed that by opening and closing the shutter according to an optimized coded pattern during the exposure duration of a photograph, one can preserve high-frequency spatial details in the blurred captured image. The image can be then de-blurred using a manually specified point-spread function. Similarly, we open and close the shutter according to a coded pattern and this code is optimized for capture. Nevertheless, there are significant differences in motion models and reconstruction procedures of both these methods. In flutter shutter (FS), a constant velocity linear motion model was assumed and deblurring was done in blurred pixels along the motion direction. On the other hand, CSC works even on very complicated motion models as long as the motion is periodic. In CSC each of the captured frames is the result of modulation with a different binary sequence whereas in FS a single frame is modulated with a 'all-pass' code. Further, our method contrasts fundamentally with FS in reconstruction of the frames. In FS the system of equations is not

under-determined whereas in CSC we have a severely under-determined system. We overcome this problem by ℓ_1-norm regularization, appropriate for enforcing sparsity of periodic motion in time. In FS a single system of equations is solved for entire image whereas in CSC at each pixel we temporally reconstruct the periodic signal by solving an under-determined system.

2.1.4 Capture and reconstruction procedure

The sequence of steps involved in the capture and reconstruction of a high-speed periodic phenomenon with typical physical values are listed below with references to appropriate sections for detailed discussion.

- Goal: Using a 25 fps camera and a shutter which can open and close at 2000 Hz, capture a high-speed periodic phenomenon of unknown period by observing for 5s.

- The length of the binary code needed is $N = 2000 \times 5 = 10000$. For an upsampling factor of $U = 2000/25 = 80$, find the optimal pseudo random code of length N (Section 2.3.1).

- Capture $M = 25 \times 5 = 125$ frames by fluttering the shutter according to the optimal code. Each captured frame is an integration of the incoming visual signal modulated with a corresponding subsequence of binary values of length $U = 80$ (Section 2.2.3).

- Estimate the fundamental frequency of the periodic signal (Section 2.2.4.3).

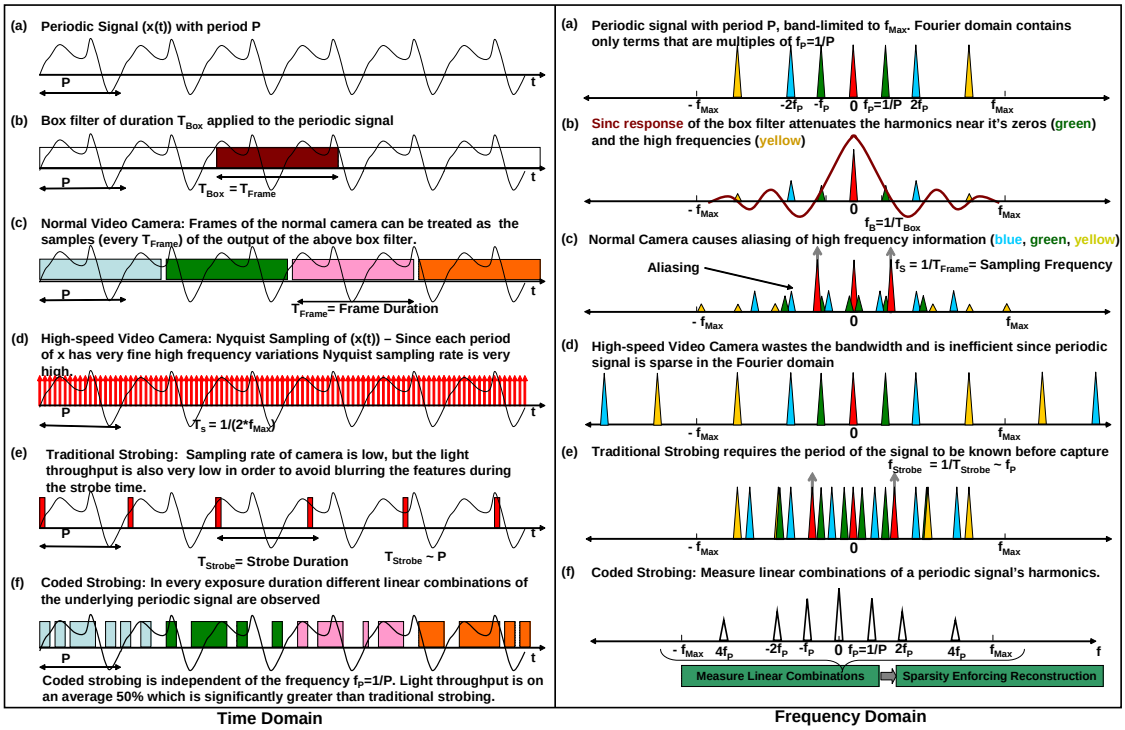

Figure 2.2: Time domain (Left) and the corresponding frequency domain (Right) characteristics of various sampling techniques as applicable to periodic signals. Note that capturing high-speed visual signals using normal camera can result in attenuation of high frequencies ((b) and (c)) whereas a high-speed camera demands large bandwidth (d) and traditional strobing is light-inefficient (e). Coded strobing is shown in (f). To illsutrate sampling only two replicas have been shown and note that colors used in time domain and frequency domain are unrelated.

- Using the estimated fundamental frequency, at each pixel reconstruct the periodic signal of length $N = 10000$ from $M = 125$ values by recovering the signal's sparse Fourier coefficients (Section 2.2.4).

2.2 Strobing and Light Modulation

2.2.1 Traditional sampling techniques

Sampling is the process of converting a continuous domain signal into a set of discrete samples in a manner that allows approximate or exact reconstruction of

the continuous domain signal from just the discrete samples. The most fundamental result in sampling is that of Nyquist-Shannon sampling theorem. Figure 2.2 provides a graphical illustration of traditional sampling techniques applied to periodic signals.

Nyquist sampling: Nyquist-Shannon sampling states that when a continuous domain signal is band-limited to $[0\,,\,f_0]$ Hz, one can exactly reconstruct the band-limited signal, by just observing discrete samples of the signal at a sampling rate f_s greater than $2f_0$ [111]. When the signal has frequency components that are higher than the prescribed band-limit, then during reconstruction, the higher frequencies get aliased as lower frequencies contributing to erroneous reconstruction (see Figure 2.2(Right)(c)). If the goal is to capture a signal whose maximum frequency f_{Max} is 1000 Hz, then one needs a high-speed camera capable of 2000 fps in order to acquire the signal. Such high-speed video cameras are light limited and expensive.

Band-pass sampling (strobing): If the signal is periodic as shown in Figure 2.2(Left)(a), then we can intentionally alias the periodic signal by sampling at a frequency very close to the fundamental frequency of the signal as shown in Figure 2.2(Left)(e). This intentional aliasing allows us to measure the periodic signal. This technique is commonly used for vocal fold visualization [100][135]. However, traditional strobing suffers from the following limitations. The frequency of the original signal must be known at capture-time so that one may perform strobing at the right frequency. Secondly, the strobe signal must be 'ON' for a very short duration so that the observed high-speed signal is not smoothed out and this makes traditional strobing light-inefficient. Despite this handicap, traditional strobing is an extremely

interesting and useful visualization tool (and has found several applications in varying fields).

Non-uniform sampling: With periodic sampling, aliasing occurs when the sampling rate is not adequate because, all frequencies of the form $f_1 + k \cdot f_s$ (k an integer) lead to identical samples. One method to counter this problem is to employ non-uniform or random sampling [19][98]. The key idea in non-uniform sampling [19][98] is to ensure a set of sampling instants such that the observation sequence for any two frequencies are different at least in one sampling instant. This scheme has not found widespread practical applicability because of its noise sensitivity and light inefficiency.

2.2.2 Periodic signals

Since, the focus of this chapter is high-speed video capture of periodic signals, we first study the properties of such signals.

2.2.2.1 Fourier domain properties of periodic signals

Consider a signal $x(t)$, which has a period $P = 1/f_P$ and a bandlimit f_{Max}. Since the signal is periodic, we can express it as,

$$x(t) = x_{DC} + \sum_{j=1}^{j=Q} a_j \cos(2\pi j f_P t) + b_j \sin(2\pi j f_P t) \tag{2.1}$$

Therefore, the Fourier transform of the signal $x(t)$ contains energy only in the frequencies corresponding to jf_P, where $j \in \{-Q, -(Q-1), ...0, 1, ..., Q\}$. Thus, a periodic signal has a maximum of $(K = 2Q+1)$ non-zero Fourier coefficients. There-

fore, periodic signals by definition, have a very sparse representation in the Fourier domain. Recent advances in the field of compressed sensing (CS) [43][23][9][22][144] have developed reliable recovery algorithms for inferring sparse representations if one can measure arbitrary linear combinations of the signals. Here, we propose and describe a method for measuring such linear combinations and use the reconstruction algorithms inspired by CS to recover the underlying periodic signal from its low-frame-rate observations.

2.2.2.2 Effect of visual texture on periodic motion

Visual texture on surfaces exhibiting periodic motion introduces high frequency variations in the observed signal (Figure 2.3(d)). As a very simple instructive example consider the fan shown in Figure 2.3(a). The fan rotates at a relatively slow rate of 8.33 Hz. This would seem to indicate that in order to capture the spinning fan one only needs a 16.66 fps camera. During exposure time of 60 ms of a 16.66 Hz camera, the figure '1' written on the fan blade completes about half a revolution blurring it out (Figure 2.3(b)). Shown in Figure 2.3(c) is the time profile of the intensity of a single pixel using a high-speed video camera. Note that the sudden drop in intensity due to the dark number '1' appearing on the blades persists only for about 1 millisecond. Therefore, we need a 1000 fps high-speed camera to observe the '1' without any blur. In short, the highest temporal frequency observed at a pixel is a product of the highest frequency of the periodic event in time and the highest frequency of the spatial pattern on the objects across the direction of

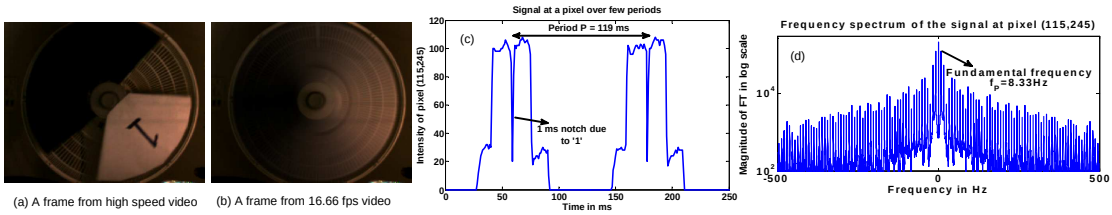

Figure 2.3: (a) Video of a fan from a high-speed camera (b) A 16.66 Hz camera blurs out the '1' in the image (c) Few periods of the signal at a pixel where the figure '1' passes. Note the notch of duration 1 ms in the intensity profile. (d) Fourier transform of the signal in (c). Notice the higher frequency components in a signal with low fundamental frequency f_P.

motion. This makes the capture of high-speed periodic signals with texture more challenging.

2.2.2.3 Quasi-periodic signals

Most real world "periodic signals" are not exactly so, but almost; there are small changes in the period of the signal over time. We refer to such broader class of signals as quasi-periodic. For example, the Crest toothbrush we use in our experiments exhibits a quasi-periodic motion with fundamental frequency that varies between $63 - 64$ Hz. Figure 2.4(a) shows few periods of a quasi-periodic signal at a pixel of a vibrating tooth brush. Variation in fundamental frequency f_P, between 63 and 64 Hz, over time can be seen in (b). Variation in f_P of a quasi-periodic signal is reflected in its Fourier transform which contains energy not just at multiples jf_P but in small band around jf_P. Nevertheless, like periodic signals, the Fourier coefficients are concentrated at jf_P (Figure 2.4(c)) and are sparse in the frequency domain. The coefficients are distributed in a band $[jf_P - j\Delta f_P\,,\, jf_P + j\Delta f_P]$. For example, $\Delta f_P = 0.75$ Hz in Figure 2.4(d).

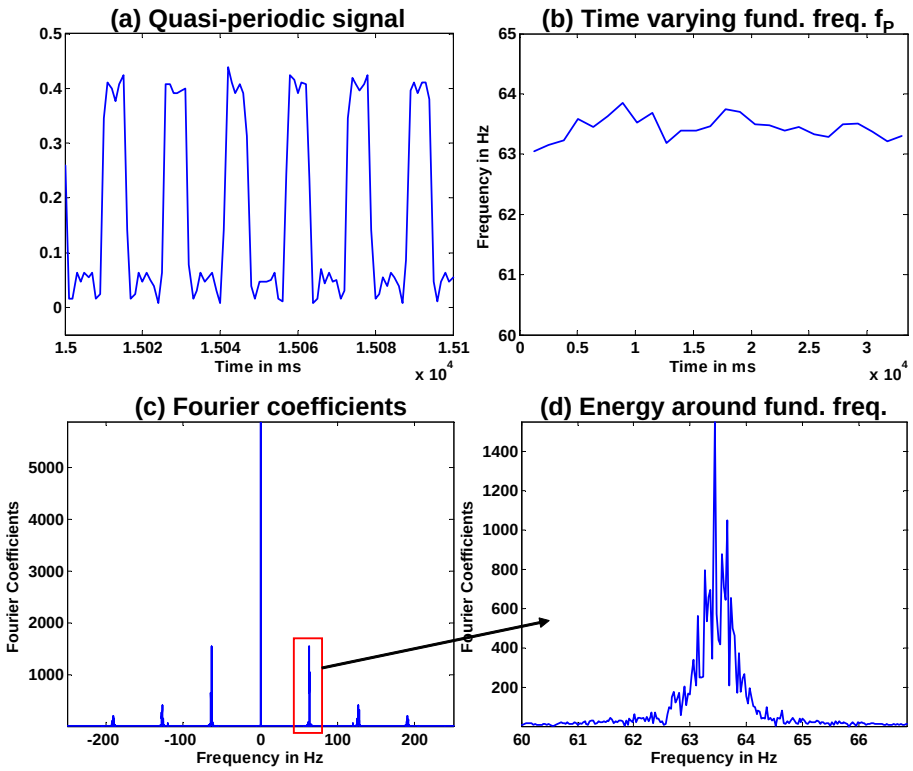

Figure 2.4: (a) Six periods of a $N = 32768$ ms long quasi-periodic signal at a pixel of a scene captured by 1000 fps high-speed camera. (b) Fundamental frequency f_P varying with time. (c) Fourier coefficients of the quasi-periodic signal shown in (a). (d) On zoom we notice that the signal energy is concentrated in a band around the fundamental frequency f_P and its harmonics.

2.2.3 Coded exposure sampling (or Coded strobing)

The key idea is to measure appropriate linear combinations of the periodic signal and then recover the signal by exploiting the sparsity of the signal in Fourier domain (Figure 2.5). Observe that by coding the incoming signal during the exposure duration, we take appropriate projections of the desired signal.

2.2.3.1 Camera observation model

Consider a luminance signal $x(t)$. If the signal is band-limited to $[-f_{Max}\ ,\ f_{Max}]$, then in order to accurately represent and recover the signal, we only need to measure

samples of the signal that are $\delta t = 1/(2f_{Max})$ apart where δt represents the temporal resolution with which we wish to reconstruct the signal. If the total time of observing the signal is $N\delta t$, then the N samples can be represented in a N dimensional vector x.

In a normal camera, the radiance at a single pixel is integrated during the exposure time, and the sum is recorded as the observed intensity at a pixel. Instead of integrating during the entire frame duration, we perform amplitude modulation of the incoming radiance values, before integration. Then the observed intensity values y at a given pixel can be represented as

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \eta, \tag{2.2}$$

where the $M \times N$ matrix C performs both the modulation and integration for frame duration, and η represents the observation noise. Figure 2.5 shows the structure of matrix C. If the camera observes a frame every T_s seconds, the total number of frames/observations would be $M = N\delta t/T_s$ and so y is a $M \times 1$ vector. The camera sampling time T_s is far larger than the time resolution we would like to achieve (δt), therefore $M << N$. The upsampling factor (or decimation ratio) of CSC can be defined as,

$$\text{Upsampling factor} = U = \frac{N}{M} = \frac{2f_{Max}}{f_s}. \tag{2.3}$$

For example, in the experiment shown in Figure 2.15, $f_{Max} = 1000$ Hz, and $f_s = 25$ fps. Therefore, the upsampling factor achieved is 80, i.e., the frame-rate of CSC is eighty times smaller than that of an equivalent high-speed video camera. Even though, the modulation function can be arbitrary, in practice it is usually restricted

to be binary (open or close shutter). Effective modulation can be achieved with codes that have a 50% transmission, i.e., the shutter is open for 50% of the total time, thereby limiting light-loss at capture-time to just 50%.

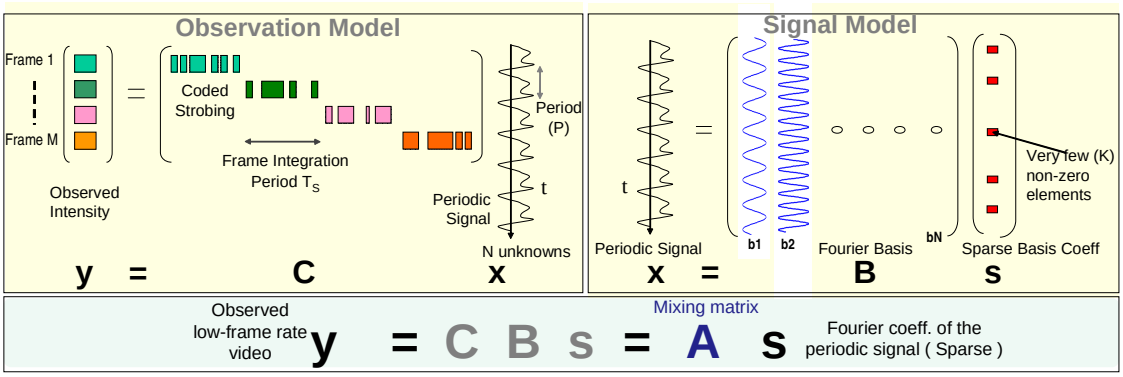

Figure 2.5: Observation model shows the capture process of the CSC where different colors correspond to different frames and the binary shutter sequence is depicted using the presence or absence of color. Note that each frame uses a different binary sub-sequence. The signal model illustrates the sparsity in the frequency spectrum of a periodic signal.

2.2.3.2 Signal model

If x, the luminance at a pixel is bandlimited it can be represented as,

$$\mathbf{x} = \mathbf{Bs}, \tag{2.4}$$

where, the columns of B contain Fourier basis elements. Moreover, since the signal $x(t)$ is assumed to be periodic, we know that the basis coefficient vector s is sparse as shown in Figure 2.5. Putting together the signal and observation model, the intensities in the observed frames are related to the basis coefficients as,

$$y = Cx + \eta = CBs + \eta = As + \eta, \tag{2.5}$$

where A is the effective mixing matrix of the forward process. Recovery of the high-speed periodic motion x amounts to solving the linear system of equations (2.5).

2.2.4 Reconstruction algorithms

To reconstruct the high-speed periodic signal x, it suffices to reconstruct its Fourier coefficients s from modulated intensity observations y of the scene.

Unknowns, measurements and sparsity: In (2.5), the number of unknowns exceeds the number of known variables by a factor U (typically 80) and hence the system of equations (2.5) is severely under-determined ($M << N$). To obtain robust solutions, further knowledge about the signal must be used. Since the Fourier coefficients s, of a periodic signal x, are sparse, a reconstruction technique enforcing sparsity of s could still hope to recover the periodic signal x.

We present two reconstruction algorithms, one which enforces the sparsity of the Fourier coefficients and is inspired by compressive sensing and other which additionally enforces the structure of the sparse Fourier coefficients.

2.2.4.1 Sparsity enforcing reconstruction

Estimating a sparse vector s (with K non-zero entries) that satisfies $y = As+\eta$, can be formulated as an ℓ_0 optimization problem:

$$(P0): \qquad min||s||_0 \;\; s.t \;\; ||y - As||_2 \leq \epsilon. \tag{2.6}$$

Although for general s this is a NP-hard problem, for K sufficiently small the equivalence between ℓ_0 and ℓ_1-norm [22] allows us to reformulate the problem as one

of ℓ_1-norm minimization, which is a convex program with very efficient algorithms [43][22][9].

$$(P1): \qquad min||s||_1 \;\; s.t \;\; ||y - As||_2 \leq \epsilon \tag{2.7}$$

The parameter ϵ allows for the variation in the modeling of signal's sparsity and/or noise in the observed frames. In practice, it is set to a fraction of captured signal energy (e.g., $\epsilon = 0.03||y||_2$) and is dictated by the prior knowledge about camera noise in general and the extent of periodicity of the captured phenomenon. An interior point implementation basis pursuit de-noising (BPDN) of (P1) is used to accurately solve for s. Instead, in most experiments in this chapter, at the cost of minor degradation in performance we use CoSaMP [107], a faster greedy algorithm to solve (P0). Both (P0) and (P1) do not take into account the structure in the sparse coefficients of the periodic signal. By additionally enforcing the structure of the sparse coefficients s, we achieve robustness in the recovery of the periodic signal.

2.2.4.2 Structured sparse reconstruction

We recall that periodic/quasi-periodic signals are (a) sparse in the Fourier basis and (b) if the period is $P = 1/f_P$, the only frequency content the signal has is in the small bands at the harmonics jf_P, j an integer. Often, the period P is not known a priori. If the period is known or can be estimated from the data y, then for a hypothesized fundamental frequency f_H, we can construct a set S_{fH} with basis elements $[jf_H - \Delta f_H \;,\; jf_H + \Delta f_H]$, for $j \in \{-Q, ...0, 1, ..., Q\}$ such that all the sparse Fourier coefficients will lie in this smaller set. Now the problem (P0) can

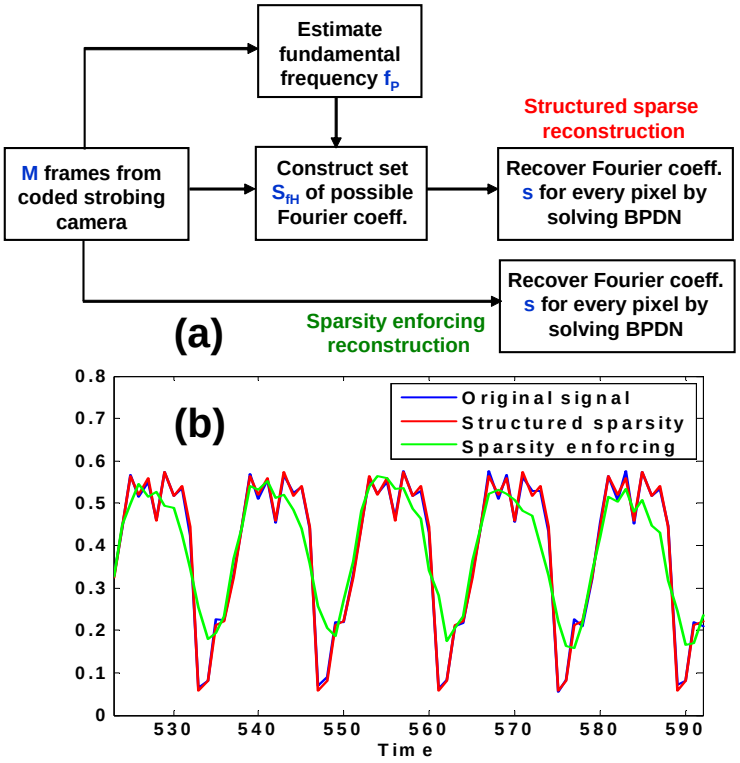

Figure 2.6: (a) Overview of *structured sparse* and *sparsity enforcing* reconstruction algorithms (b) Five periods of a noisy (SNR=35 dB) periodic signal x ($P = 14$ units). Signal recovered by *structured* and *normal* sparsity enforcing reconstruction are also shown.

instead be reformulated as

$$(P_{Structured}): \quad min||s||_0 \qquad \text{s.t} \tag{2.8}$$

$$||y - As||_2 \leq \epsilon \qquad \text{and}$$

$$nonZero(s) \in S_{fH} \quad \text{for some } f_H \in [0\,,\, f_{Max}].$$

where $nonZero(s)$ is a set containing all the non-zero elements in the reconstructed s. Since the extent of quasi-periodicity is not known a priori, the band Δf_H is chosen safely large and the non-zero coefficients continue to remain sparse in the set S_{fH}. Intuitively, problem $P_{Structured}$ gives a better sparse solution compared to

(P0) since the non-zero coefficients are searched over a smaller set S_{fH}. An example of a periodic signal and its recovery using *sparsity enforcing* (P1) and *structured sparsity* are shown in Figure 2.6(b). The recovery using $P_{Structured}$ is exact whereas (P0) fails to recover the high-frequency components.

The restatement of the problem provides two significant advantages. Firstly, it reduces the problem search space of the original ℓ_0 formulation. To solve the original ℓ_0 formulation, one has to search over ${}^{N}C_{K}$ sets. For example, if we observe a signal for 5 seconds at 1 ms resolution, then N is 5000 and ${}^{N}C_{K}$ is prohibitively large (10^{212} for $K = P = 100$). Secondly, this formulation implicitly enforces the quasi-periodicity of the recovered signal and this extra constraint allows us to solve for the unknown quasi-periodic signal with far fewer measurements than would otherwise be possible. The type of algorithms which exploit further statistical structure in the support of the sparse coefficients come under model-based compressive sensing [10].

2.2.4.3 Knowledge of fundamental frequency

Structured sparse reconstruction performs better over a larger range of up-sampling factors and since the structure of non-zero coefficients is dependent on fundamental frequency f_P, we estimate it first.

Identification of fundamental frequency: For both periodic and quasi-periodic signals we solve a sequence of least-square problems to identify the fundamental frequency f_P. For a hypothesized fundamental frequency f_H, we build a set S_{fH} with only the frequencies jf_H (for both periodic and quasi-periodic sig-

nals). Truncated matrix A_{fH} is constructed by retaining only the columns with indices in S_{fH}. Non-zero coefficients $\hat{s}_{fH}$ are then estimated by solving the equation $y = A_{fH}s_{fH}$ in a least-squares sense. We are interested in f_H which has a small reconstruction error $\|y - \hat{y}_{fH}\|$ (or largest output SNR) where $\hat{y}_{fH} = A_{fH}\hat{s}_{fH}$. If f_P is the fundamental frequency, then all the sets S_{fH}, where f_H is a factor of f_P, will provide a good fit to the observed signal y. Hence, the plot of output SNR has multiple peaks corresponding to the good fits. From these peaks we pick the one with largest f_H. In Figure 2.7, we show the results of experiments on synthetic datasets, under two scenarios: noisy signal and quasi-periodicity. We note that even when (a) the signal is noisy and (b) when the quasi-periodicity of the signal increases, the last peak in the SNR plot occurs at fundamental frequency f_P. We generate quasi-periodic signals from periodic signals by warping the time variable. Note that, solving a least squares problem for a hypothesized fundamental frequency f_H is equivalent to solving $P_{structured}$ with $\Delta f_H = 0$. Setting $\Delta f_H = 0$ eases the process of finding the fundamental frequency by avoiding the need to set the parameter Δf_H appropriate for both the captured signal and f_H. This is especially useful for quasi-periodic signals where a priori knowledge of quasi-periodicity is not available.

2.3 Design Analysis

In this section, we analyze important design issues and gain a better understanding of the performance of the coded strobing method through experiments on synthetic examples.

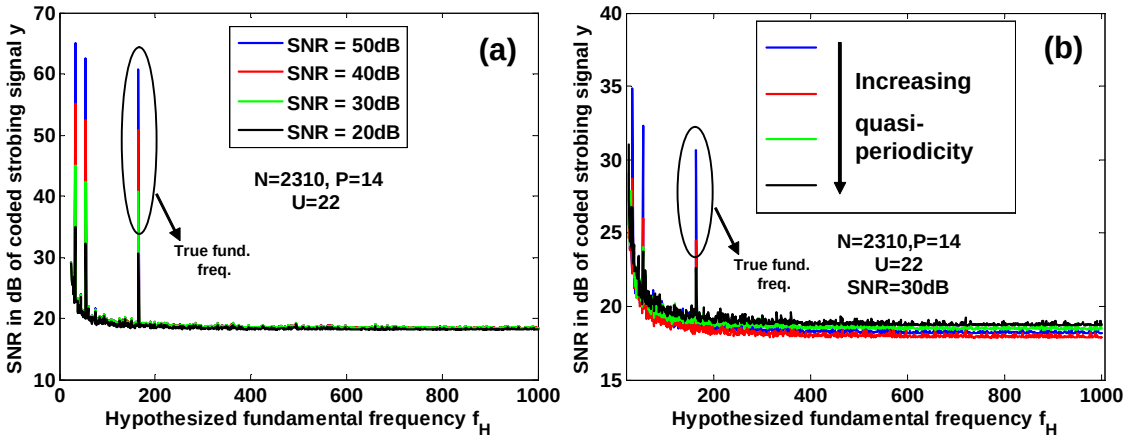

Figure 2.7: Identifying the fundamental frequency f_P. Output SNR $||y||/||y - \hat{y}_{fH}||$ in dB is plotted against hypothesized fundamental frequency f_H. (a) Plot of SNR as the noise in y is varied. Note that the last peak occurs at $f_H = 165$ $(= \frac{N}{P})$. (b) Plot of SNR with varying level of quasi-periodicity.

2.3.1 Optimal code for coded strobing

Theoretically optimal code: The optimization problems (2.6) and (2.7) give unique and exact solutions provided the under-determined matrix A satisfies the *restricted isometry property* (RIP) [30]. Since the location of the K non-zeros of the sparse vector s which generates the observation y is not known a priori, RIP demands that all sub-matrices of A with $2K$ columns have a low condition number. In other words, every possible restriction of $2K$ columns are nearly orthonormal and hence isometric. Evaluating RIP for a matrix is a combinatorial problem since it involves checking the condition number of all ${}^{N}C_{2K}$ submatrices.

Alternately, the matrix A satisfies RIP if every row of C is incoherent with every column of B. In other words, no row of C can be sparsely represented by columns of B. Tropp et al. [144] showed in a general setting that if the code matrix C is drawn from a IID Rademacher distribution, the resulting mixing matrix A

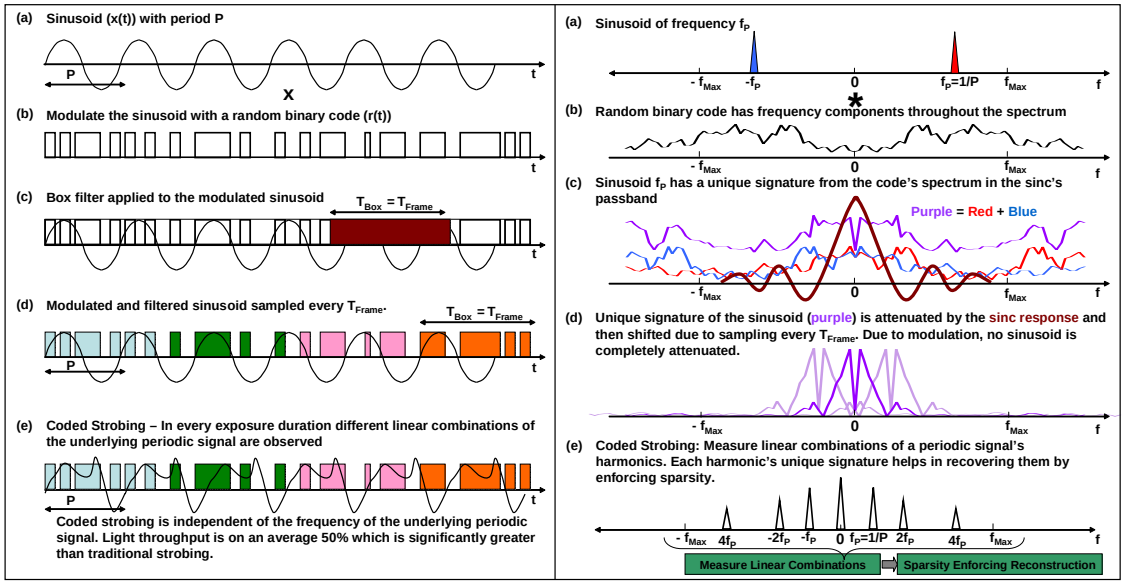

Figure 2.8: Time domain (Left) and corresponding frequency domain (Right) understanding of CSC. Shown in (a) is a single sinusoid. (b),(c) & (d) show the effect of coded strobing capture on the sinusoid. (e) coded strobing capture of multiple sinusoids is simply a linear combination of the sinusoids.

satisfies RIP with a high probability. It must be noted that a modulation matrix C with entries '+1', '-1' is implementable but would involve using a beam splitter and two cameras in place of one. Due to ease of implementation (details in section 2.4), for modulation we use a binary '1', '0' code matrix C as described in section 2.2.3.1. For a given signal length N and an upsampling factor U we would like to pick a binary '1', '0' code which results in mixing matrix A, optimal in the sense of RIP.

Note that the sparsity of quasi-periodic signals is structured and the non-zero elements occur at regular intervals. Hence, unlike the general setting, RIP should be satisfied and evaluated over only a select subset of columns. Since the fundamental frequency f_P of the signal is not known a priori, it suffices if the isometry is evaluated over a sequence of matrices $\bar{A}$ corresponding to the hypothesized funda-

mental frequency f_H. Hence, for a given N and U, a code matrix C which results in a smallest condition number over all the sequence of matrices $\bar{A}$ is desired. In practice, such a C is sub-optimally found by randomly generating the binary codes tens of thousand times and picking the best one.

Compared to a normal camera, CSC blocks half the light but captures all the frequency content of the periodic signal. The sinc response of the box filter of a normal camera attenuates the harmonics near its zeros as well as the higher frequencies as shown in Figure 2.2(b). To avoid the attenuation of harmonics, the frame duration of the camera has to be changed appropriately. But, this is undesirable since most cameras come with a discrete set of frame rates. Moreover, it is hard to have a priori knowledge of the signal's period. This problem is entirely avoided by modulating the incoming signal with a pseudo-random binary sequence. Shown in Figure 2.8 is the temporal and frequency domain visualization of the effect of CSC on a single harmonic. Modulation with a pseudo-random binary code spreads the harmonic across the spectrum. Thus, every harmonic irrespective of its position avoids the attenuation, the sinc response causes.

We performed numerical experiments to show the effectiveness of CSC (binary code) over the normal camera (all '1' code). Shown in Table 2.1 are the comparison of the largest and smallest condition numbers of the matrix $\bar{A}$ arising in CSC and the normal camera. For a given signal length $N = 5000$ and upsampling factor $U = 25$ (the second column in Table 1), we vary the period P and generate different matrices $\bar{A}$ for both CSC and normal camera. The largest condition number (1.8×10^{19}) of mixing matrix $\bar{A}$ of a normal camera occurs for signal of period $P = 75$. Similarly,

Condition Number κ (Period P)	**U = 25**	**U = 40**	**U = 47**	**U = 55**	**U = 63**	**U = 91**
NC: largest	1.8×10^{19} (75)	8.6×10^{33} (5)	6.1×10^{32} (47)	4.5×10^{65} (95)	3.4×10^{64} (90)	6.5×10^{48} (70)
CSC: largest	1.3×10^{3} (9)	1.4×10^{4} (7)	6.0×10^{3} (8)	2.1×10^{4} (19)	8.1×10^{2} (27)	2.4×10^{3} (7)
NC: smallest	5.9×10^{2} (67)	8.4×10^{2} (63)	1.5×10^{3} (54)	2.7×10^{2} (92)	1.5×10^{3} (80)	1.6×10^{3} (55)
CSC: smallest	16.5 (67)	11.5 (94)	10.1 (98)	9.7 (90)	10.9 (77)	13.2 (53)

Table 2.1: Table comparing the largest and smallest condition numbers of mixing matrix $\bar{A}$ corresponding to normal (NC) and coded strobing exposure (CSC).

the smallest condition number occurs for $P = 67$. On the other hand, the mixing matrix $\bar{A}$ of CSC has significantly lower maximum (at $P = 9$) and minimum (at $P = 67$) condition numbers. Note that the largest and smallest condition number of CSC matrices $\bar{A}$ across different upsampling factors U are significantly smaller compared to those of normal camera matrices. This indicates that when the period of the signal is not known a priori, it is prudent to use the CSC over normal camera.

Performance evaluation: We perform simulations on periodic signals to compare the performance of *sparsity enforcing* and *structured sparse* reconstruction algorithms on CSC frames, *structured sparse* reconstruction on normal camera frames and traditional strobing. SNR plots of the reconstructed signal using the four approaches for varying period P, upsampling factor U and noise level in y are shown in Figure 2.9. The signal length is fixed at $N = 2000$ units. The advantage of *structured sparse* reconstruction is apparent from comparing blue and red plots. The advantage of CSC over a normal camera can be seen by comparing blue and black plots. Note that the normal camera performs poorly when the upsampling factor U is a multiple of the period P.

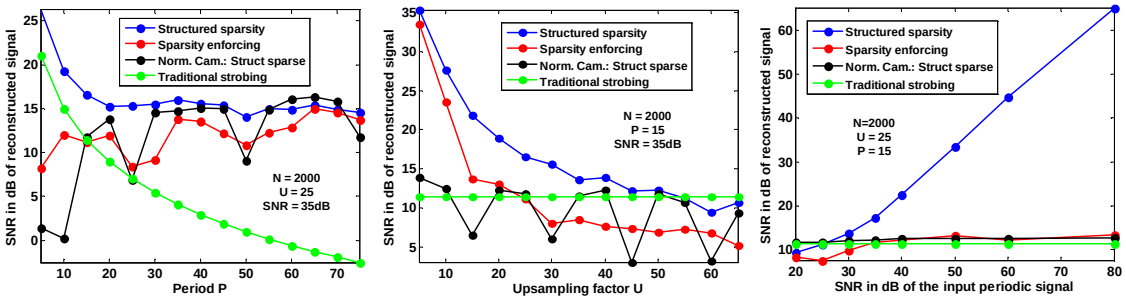

Figure 2.9: Performance analysis of *structured* and *normal* sparsity enforcing reconstruction for CSC and *structured* sparsity enforcing reconstruction for normal camera: (a) Reconstruction SNR as the period P increases. (b) Reconstruction SNR as upsampling factor U increases. (c) Reconstruction SNR as the noise in y is varied.

2.3.2 Experiments on a synthetic animation

We performed experiments on a synthetic animation of a fractal to show the efficacy of our approach. We also analyzed the performance of the algorithm under various noisy scenarios. It was assumed that at every $\delta t = 1$ ms, a frame of the animation is being observed and that the animation is repetitive with $P = 25$ ms (25 distinct images in the fractal). Two such frames are shown in Figure 2.10(a). A normal camera running at $f_s = 25$ fps will integrate 40 frames of the animation into a single frame, resulting in blurred images. Two images from a 25 fps video are shown in (b). By performing amplitude modulation at the shutter, as described in 2.2.3.1, the CSC obtains frames at the same rate as that of the normal camera (25 fps) but with the images encoding the temporal movement occurring during the integration process of the camera sensor. Two frames from the CSC are shown in (c). Note that in images (b) & (c) and also images in other experiments we rescaled the intensities appropriately for better display. We observed the animation for 5 seconds ($N = 5000$) resulting in $M = 125$ frames. From these 125 frames we recover frequency

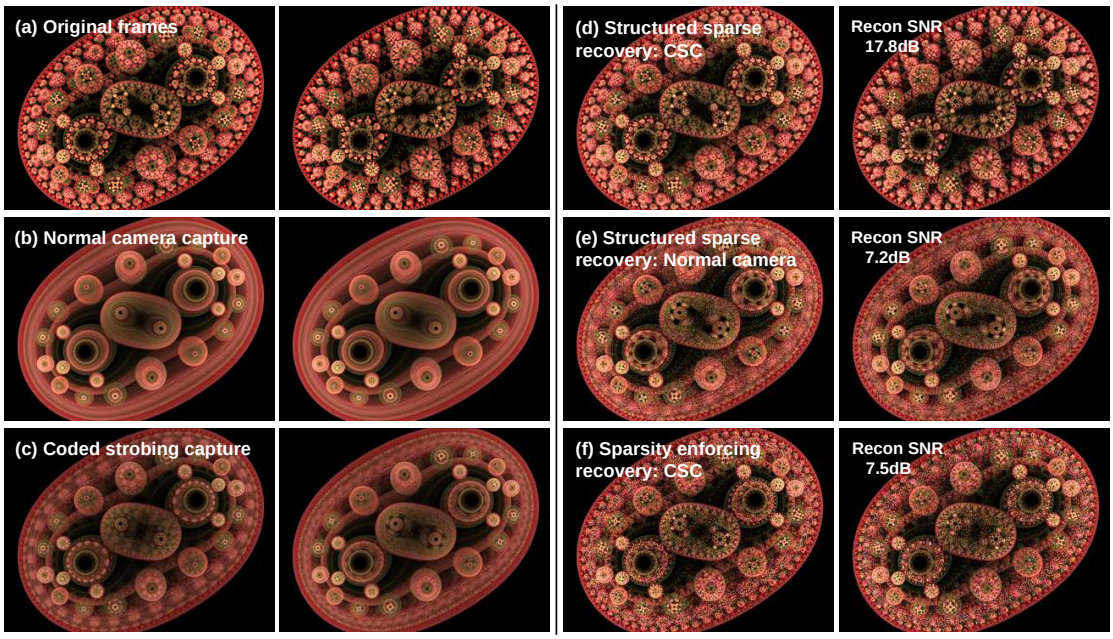

Figure 2.10: (a) Original frames of the fractal sequence which repeat every $P = 25$ ms. (b) Frames captured by a normal 25 fps camera. (c) Frames captured by a CSC running at 25 fps. (d) Frames reconstructed by enforcing structured sparsity on CSC frames. (e) Frames reconstructed by enforcing structured sparsity on normal camera frames. (f) Frames reconstructed by enforcing simple sparsity on CSC frames. Overall 5 seconds ($N = 5000$) of the sequence was observed to reconstruct it back fully. Upsampling factor was set at $U = 40$ ($M = 125$) corresponding to $\delta t = 1$ ms. Note that image intensities in (b) and (c) have been rescaled appropriately for better display.

content of the periodic signal being observed by enforcing sparsity in reconstruction as described in 2.2.4. We compared the *structured sparse* reconstruction on normal camera frames, *normal sparse* and *structured sparse* reconstruction on CSC frames and the results are shown in (d),(e) and (f) respectively. It is important to modulate the scene with a code to capture all frequencies and enforcing both sparsity and structure in reconstruction ensures that the periodic signal is accurately recovered.

Noise analysis and influence of upsampling factor: We perform statistical analysis on the impact of two most common sources of noise in CSC and also analyze the influence of upsampling factor on reconstruction. We recover the signal using *structured sparsity* enforcing reconstruction. First, we study the impact of sensor noise. Figure 2.11(a) shows the performance of our reconstruction with increasing noise level η. We fixed the upsampling factor at $U = 40$ in these simulations. The reconstruction SNR varies linearly with the SNR of the input signal in accordance with compressive sensing theory. The second most significant source of errors in a CSC are errors in the implementation of the code due to lack of synchronization between the shutter and the camera. These errors are modeled as bit-flips in the code. Figure 2.11(b) shows the resilience of the coded strobing method to such bit-flip errors. The upsampling factor is again fixed at 40. Finally, we are interested in understanding how far the upsampling factor can be pushed without compromising the reconstruction quality. Figure 2.11(c) shows the reconstruction SNR as the upsampling factor increases. This indicates that by using structured sparsity enforcing reconstruction algorithm, we can achieve large upsampling factors with a reasonable fidelity of reconstruction. Using the procedure described in previous

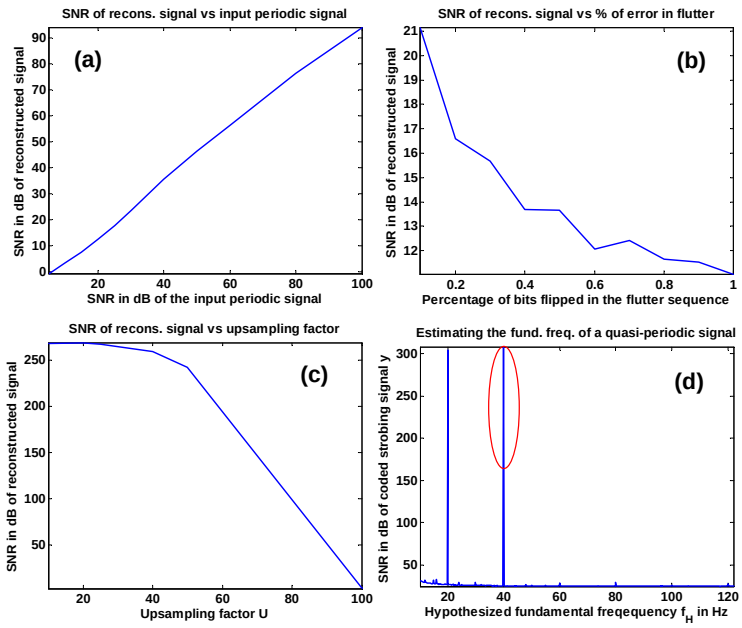

Figure 2.11: Performance analysis of CSC: (a) Reconstruction SNR as the observation noise increases. (b) Impact of bit-flips in binary exposure sequence. (c) Coded strobing camera captures the scene accurately upto an upsampling factor $U = 50$. (d) $||y||/||y - \hat{y}||$ against varying hypothesized fundamental frequency f_H.

section we estimate the fundamental frequency as $f_p = 40$ Hz (Figure 2.11(d)).

2.4 Experimental Prototypes

2.4.1 Hi-speed video camera

In order to study the feasibility and robustness of the proposed camera, we first tested the approach using a high-speed video camera. We used an expensive 1000 fps video camera, and captured high-speed video. We had to use strong illumination sources to light the scene and capture reasonably noise-free high-speed frames. We then added several of these frames (according to the strobe code) in software to simulate low speed coded strobing camera frames. The simulated CSC frames were used to reconstruct the high-speed video. Some results of such experiments are

reported in Figure 2.12.

2.4.2 Sensor integration mechanism

We implemented the CSC for our experiments using an off-the-shelf Dragonfly2 camera from PointGrey Research [121], without modifications. The camera allows a triggering mode (Multiple Exposure Pulse Width Mode- Mode 5) in which the sensor integrates the incoming light when the trigger is '1' and is inactive when the trigger is '0'. The trigger allows us exposure control at a temporal resolution of $\delta t = 1$ ms. For every frame we use a unique triggering sequence corresponding to a unique code. The camera outputs the integrated sensor readings as a frame after a specified number of integration periods. Also, each integration period includes at its end a period of about 30 ms during which the camera processes the integrated sensor readings into a frame. The huge benefit of this setup is that it allows us to use an off-the-shelf camera to slow down high-speed events around us. On the other hand, the hardware bottleneck in the camera restricts us to operate at an effective frame rate of 10 fps (100 ms) and a strobe rate of 1000 strobes/second ($\delta t = 1$ ms).

2.4.3 Ferro-electric shutter

The PointGrey Dragonfly2 provides exposure control with a time resolution of 1 ms. Hence, it allows us a temporal resolution of $\delta t = 1$ ms at recovery time. However, when the maximum linear velocity of the object is greater than 1 pixel per ms, the reconstructed frames have motion blur. One can avoid this problem

with finer control over the exposure time. For example, a DisplayTech ferro-electric liquid crystal shutter provides an ON/OFF contrast ratio of about 1000 : 1, while simultaneously providing very fast switching time of about 250μs. We built a prototype where the Dragonfly2 captures the frames at usual 25 fps and also triggers a PIC controller after every frame which in turn flutters the ferro-electric shutter with a new code at a specified temporal frequency. In our experiment we set the temporal resolution at 500μs i.e. 2000 strobes/second.

2.4.4 Retrofitting commercial stroboscopes

Another exciting alternative to implement CSC is to retrofit commercial stroboscopes. Commercial stroboscopes used in laryngoscopy usually allow the strobe light to be triggered via a trigger input. Stroboscopes that allow such an external trigger for the strobe can be easily retrofitted to be used as a CSC. The PIC controller used to trigger the ferro-electric shutter can instead be used to synchronously trigger the strobe light of the stroboscope, thus converting a traditional stroboscope to a coded stroboscope.

2.5 Experimental Results

To validate our design we conducted two kinds of experiments. In the first experiment, we captured high-speed videos and then generate CSC frames by appropriately adding frames of the high-speed video. In the second set of experiments we captured videos of fast moving objects with a low-frame-rate CSC implemented us-

ing a Dragonfly2 video camera. Details about the project and implementation can be found at the webpage http://www.umiacs.umd.edu/ dikpal/Projects/codedstrobing.html .

2.5.1 High-speed video of toothbrush

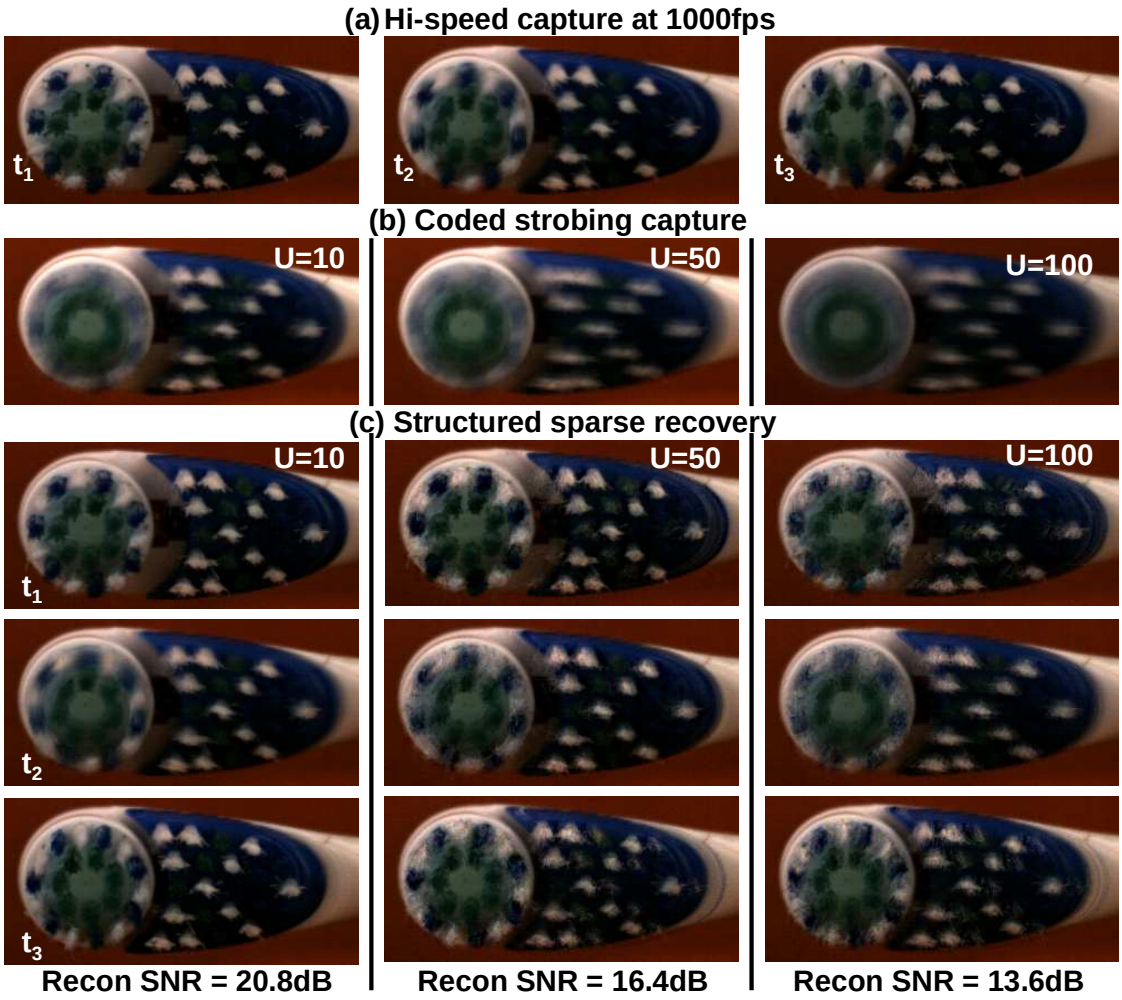

Figure 2.12: Reconstruction results of an oscillating toothbrush under three different capture parameters (U): Images for simulation captured by a 1000 fps high-speed camera at time instances t_1, t_2 and t_3 are shown in (a). The second row (b) shows a frame each from the coded strobing capture (simulated from frames in (a)) at up-sampling factors $U = 10, 50,$ and 100 respectively. Reconstruction at time instances t_1, t_2 and t_3 from the frames captured at $U = 10$ are shown in first column of (c).

We captured a high-speed (1000 fps) video of a pulsating Crest toothbrush with quasi-periodic linear and oscillatory motions at about 63 Hz. Figure 2.4(b) shows the frequency of the toothbrush as a function of time. Notice that even within a short window of 30 seconds, there are significant changes in frequency. We render a 100 fps, 20 fps, 10 fps CSC (i.e., a frame duration of 10 ms, 50 ms, 100 ms respectively) by adding appropriate high-speed video frames, but reconstruct the moving toothbrush images at a resolution of 1 ms as shown in Fig 2.12c. Frames of the CSC operating at 100, 20 and 10 fps (U = 10, 50 and 100 respectively) are shown in Figure 2.12(b). The fine bristles of the toothbrush add high frequency components because of texture variations. The bristles on the circular head moved almost 6 pixels within 1 ms. Thus the captured images from the high-speed camera themselves exhibited blur of about 6 pixels which can be seen in the recovered images. Notice that contrary to what it seems to the naked eye, the circular head of the toothbrush does not actually complete a rotation. It just exhibits oscillatory motion of 45 degrees and we are able to see it from the high-speed reconstruction.

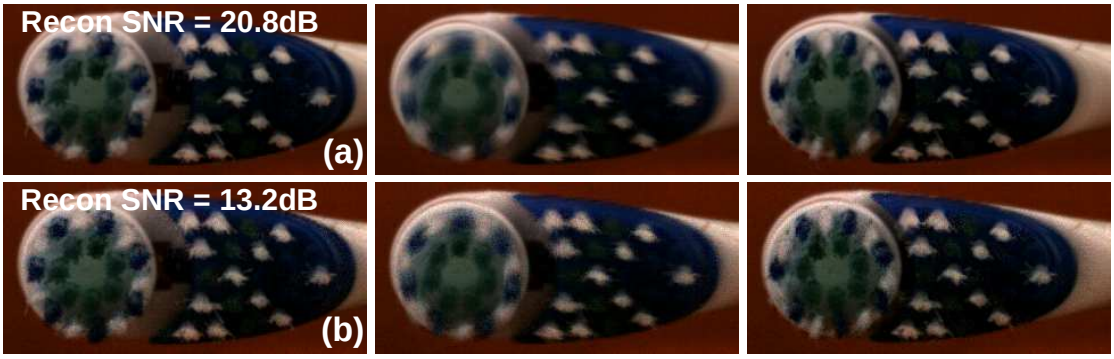

Figure 2.13: Reconstruction results of toothbrush with upsampling factor U = 10 without and with 15 dB noise in (a) and (b) respectively.

To test the robustness of coded strobing capture and recovery on the visual

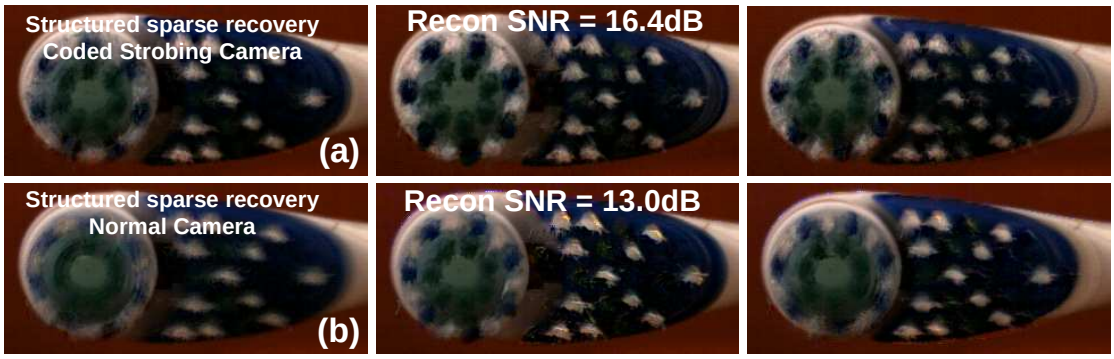

Figure 2.14: Reconstruction results of toothbrush with upsampling factor $U = 50$. Note that using CSC to capture periodic scenes allows us better reconstruction over using a normal camera.

quality of images, we corrupted the observed images y with white noise having $SNR = 15$ dB. The results of the recovery without and with noise are shown in Figure 2.13.

We compare frames recovered from CSC to those recovered from a normal camera (by enforcing structured sparsity) to illustrate the effectiveness of modulating the frames. Normal camera doesn't capture the motion in the bristles as well (Figure 2.14) and is saturated.

2.5.2 Mill-tool results using ferro-electric shutter

We used a Dragonfly2 camera with a ferro-electric shutter and captured images of a tool rotating in a mill. Since the tool can rotate at speeds as high as 12000 rpm (200 Hz), to prevent blur in reconstructed images we use the ferro-electric shutter for modulation with a temporal resolution of 0.5 ms. The CSC runs at 25 fps (40 ms frame length) with the ferro-electric shutter fluttering at 2000 strobes/second. Shown in Figure 2.15 are the reconstructions at 2000 fps ($\delta t = 0.5$ ms) of a tool

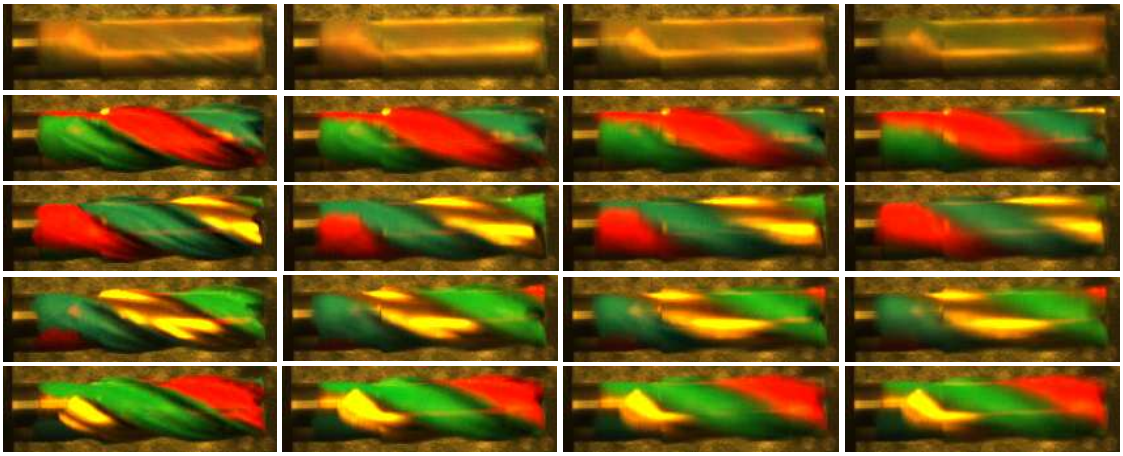

Figure 2.15: Tool bit rotating at different rpm captured using coded strobing: Top row shows the coded images acquired by a PGR Dragonfly2 at 25 fps, with an external FLC shutter fluttering at 2000 Hz. (a)-(d) Reconstruction results, at 2000 fps (temporal resolution $\delta t = 500\mu$s), of a tool bit rotating at $3000, 6000, 9000$ and 12000 rpm respectively. For better visualization, the tool was painted with color prior to the capture.

rotating at 3000, 6000, 9000 and 12000 rpm. Without any prior knowledge of scene frequencies, we use the same strobed coding and the same software decoding procedure for the mill tool rotating at different rpm. This shows that we can capture any sequence of periodic motion with unknown period with a single pre-determined code. In contrast, traditional strobing methods need prior knowledge of the period to strobe at the appropriate frequency. Note that the reconstructed image of the tool rotating at 3000 rpm is crisp (Figure 2.15(a)) and the images blur progressively as the rpm increases. Since the temporal resolution of Dragonfly2 strobe is 0.5 ms, the features on the tool begin to blur at speeds as fast as 12000 rpm (Figure 2.15(d)). In fact, the linear velocity of the tool across the image plane is about 33 pixels per ms (for 12000 rpm), while the width of the tool is about 45 pixels. Therefore, the recovered tool is blurred to about one-third its width in 0.5 ms.

2.5.3 Toothbrush using Dragonfly2 camera

We used a Dragonfly2 camera operating in Trigger Mode 5 to capture a coded sequence of the Crest toothbrush oscillating. The camera operated at 10 fps, but we reconstructed a video of the toothbrush at 1000 fps ($U = 100$) as shown in Figure 2.16. Even though the camera acquires a frame every 100 ms, the reconstruction is at a temporal resolution of 1 ms. If we assume that there are L photons per ms, then each frame of the camera would acquire around $0.5 * 100 * L$ photons. In comparison, each frame of a high-speed camera would accumulate L photons, while traditional strobing camera would accumulate $L * f_P/f_s = 6.3L$ photons per frame.

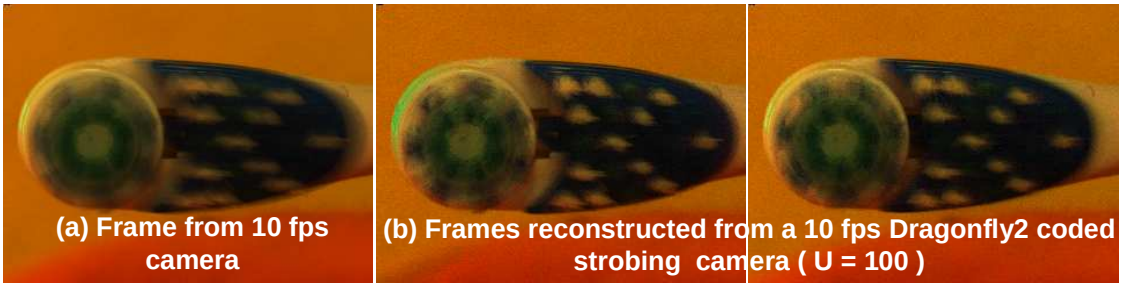

Figure 2.16: Demonstration of CSC at upsampling factor $U = 100$ using Dragonfly2. (a) Captured image from a 10 fps CSC (Dragonfly2). (b)-(c) Two reconstructed frames. While the CSC captured an image frame every 100 ms, we obtain reconstructions with a temporal resolution of 1 ms.

2.5.4 High-speed video of a jog

Using frames from a high-speed (250 fps) video of a person jogging-in-place we simulate in computer the capture of the scene using a normal camera and the CSC at upsampling factors of $U = 25, 50$ and 75. The coded frames from CSC are used to reconstruct back the original high-speed frames by enforcing *structured*

sparsity. The result of the reconstruction using frames from the CSC is contrasted with frames captured using a normal camera in Figure 2.17(a). At any given pixel, the signal is highly quasi-periodic since it is not a mechanically driven motion but our algorithm performs reasonably well in capturing the scene. In Figure 2.17(b) we contrast the reconstruction at a pixel for $U = 25$, 50, and 75.

2.6 Benefits and Limitations

2.6.1 Benefits and advantages

Coded strobing allows three key advantages over traditional strobing: (i) signal to noise ratio (SNR) improvements due to light-efficiency, (ii) does not require prior knowledge of the dominant frequency, and (iii) the ability to capture scenes with multiple periodic phenomena with different fundamental frequencies.

Light throughput: Light efficiency plays an important role if one cannot increase the brightness of external light sources. Let us consider the linear noise model (scene independent) where the SNR of the captured image is given by $LT_{Exposure}/\sigma_{gray}$, where L is the average light intensity at a pixel and σ_{gray} is a signal independent noise level which includes effects of dark current, amplifier noise and A/D converter noise. For both traditional and coded strobing cameras, the duration of the shortest exposure time should at most be $t_\delta = 1/(2f_{Max})$. In traditional strobing, this short exposure t_δ is repeated once every period of the signal, and therefore the total exposure time in every frame is given by $T_{Strobing} = (1/2f_{Max})(f_P/f_s)$. Since the total exposure time within a frame can be as large as 50% of the total frame

duration for CSC, $T_{Coded} = 1/2f_s$. The decoding process in coded strobing introduces additional noise, and this decoding noise factor is $d = \sqrt{trace((A^TA)^{-1})/M}$. Therefore, the SNR gain of CSC as compared to traditional strobing is given by

$$SNR_{Gain} = \frac{SNR_{Coded}}{SNR_{Strobing}} = \frac{(LT_{Coded})/(d\sigma)}{(LT_{Strobing})/(\sigma)} = \frac{f_{Max}}{df_P} \tag{2.9}$$

For example, in the case of the tool spinning at 3000 rpm (or 50 Hz), this gain is $20\log(1000/(2 \cdot 50)) = 20dB$ since $f_{Max} = 1000$ Hz for strobe rate 2000 strobes/second. So coded strobing is a great alternative for light-limited scenarios such as medical inspection in laryngoscopy (where patient tissue burn is a concern) and long range imaging.

Knowledge of fundamental frequency: Unlike traditional strobing, coded strobing can determine signal frequency in post-capture, software only process. This allows for interesting applications such as simultaneous capture of multiple signals with very different fundamental frequencies. Since the processing is independent for each pixel, we can support scenes with several independently periodic signals and capture them without a-priori knowledge of the frequency bands as shown in Figure 2.18(a). Shown, in Figure 2.15 are the reconstructions obtained for the tool which was rotating at 3000,4500,6000 and 12000 rpm. In all these cases, the same coded shutter sequence was used at capture-time. Also, the reconstruction algorithm can easily handle both periodic and quasi-periodic signals using the same framework.

Multiple periodic signals: Unlike traditional strobing, coded strobing allows us to capture and recover scenes with multiple periodic motions with different

fundamental frequencies. The capture in coded strobing does not rely on the frequency of the periodic motion being observed and the recovery of the signal at each pixel is independent of the other. This makes it possible to capture a scene with periodic motions with different fundamental frequency all at the same time using the same hardware settings. The different motions are independently reconstructed by first estimating the respective fundamental frequencies and then reconstructing by enforcing structured sparsity.

We performed experiments on synthetic data with two periodic motions with different fundamental frequencies. Shown in Figure 2.18(a) are few frames of the animation with a rotating globe on the left and a horse galloping on the right. The animation was created using frames of a rotating globe which repeats every 24 frames and frames of the classic galloping horse which repeats every 15 frames. For simulation, we assumed that a new frame of the animation is being observed at a resolution of $\delta t = 1$ ms and observed the animation for a total time of 4.8 seconds ($N = 4800$). This makes the period of the globe 24 ms ($f_P = 41.667$ Hz) and that of horse 15 ms ($f_P = 66.667$ Hz). The scene is captured using a 25 fps ($U = 40$) camera and few of the captured CSC frames are shown in (b). The reconstructed frames obtained by enforcing structured sparsity are shown in (c). Prior to the reconstruction of the scene at each pixel, fundamental frequencies of the different motions were estimated. For one pixel on horse (marked blue in Figure 2.18(a)) and one pixel on the globe (marked red), the output SNR $||y||/||y - \hat{y}||$ is shown as a function of hypothesized fundamental frequency f_H in Figure 2.18(d). The fundamental frequency are accurately estimated as 66.667 Hz for the horse and

41.667 Hz for the globe.

Ease of implementation: The previous benefits assume significance because modern cameras, such as PointGrey DragonFly2, allow coded strobing exposure and hence there is no need for expensive hardware modifications. We transform this off-the-shelf camera instantly into a 2000 fps high-speed camera using our sampling scheme. On the other hand, traditional strobing has been extremely popular and successful because of its direct-view capability. Since our reconstruction algorithm is not yet real-time, we can only provide a delayed viewing of the signal. Table 2.2 lists the most important characteristics of the various sampling methodologies presented.

Method	**Sampling Rate**	**Best Scenario**	**Benefits**	**Limitations**
High-speed (Nyquist)	2 f_0	Scene within f_0	Robust	Costly
Strobing (band-pass)	Lower than f_0	Periodic and Brightly lit	Direct-view	Linear search
Non-uniform	Lower than f_0	Brightly lit	No aliasing	Not robust to noise
Coded Strobing	Lower than f_0	Periodic	Light-efficient	No direct-view

Table 2.2: Table showing relative benefits and appropriate sampling for presented methods.

2.6.2 Artifacts and limitations

In this section, we address the three most dominant artifacts in our reconstructions: (a) blur in the reconstructed images due to time resolution, (b) temporal ringing introduced during de-convolution process, and (c) saturation due to specularity.

Blur: As shown in Fig 2.19, we observe blur in the reconstructed images when the higher spatio-temporal frequency of the motion is not captured by the shortest exposure time of 0.5 ms. Notice that the blur when $\delta t = 0.5$ ms is less compared to when $\delta t = 1$ ms. The width of the tool is about 45 pixels and the linear velocity of the tool across the image plane is 33 pixels per millisecond. Hence, there is a blur of about 16 pixels in the reconstructed image when $\delta t = 0.5$ ms and 33 pixels when $\delta t = 1$ ms. Note that this blur is not a result of the reconstruction process and is dependent on the smallest temporal resolution. It must also be noted here that while 12000 rpm (corresponding to 200 Hz) is significantly less compared to the 2000 Hz temporal resolution offered by coded strobing, the blur is a result of visual texture on the tool.

Temporal ringing: Temporal ringing is introduced in the reconstructed images during the reconstruction (deconvolution) process. For simplicity, we presented results without any regularization in the reconstruction process (Figure 2.12(c)). Note that in our algorithm reconstruction is per pixel and the ringing is over time. Figure 2.20(a) shows temporal ringing at two spatially close pixels. Since the waveforms at these two pixels are related (typically phase shifted), the temporal ringing appears as spatial ringing in the reconstructed images (Figure 2.16(b)). Either data independent Tikhonov regularization or data dependent regularization (like priors) can be used to improve the visual quality of the reconstructed videos.

Saturation: Saturation in the captured signal y results in sharp edges which in turn leads to ringing artifacts in the reconstructed signal. In Figure 2.20(b) we can see that the periodic signal recovered from saturated y has temporal ringing.

Since reconstruction is independent for each pixel, the effect of saturation is local and does not affect the rest of the pixels in the image. A typical cause of saturation in the captured image is due to specularities in the observed scene. Specularities, that are not saturated, do not pose a problem and are reconstructed as well as other regions.

2.7 Discussion and Conclusion

2.7.1 Spatial redundancy

In this chapter, we discussed a method called coded strobing that exploits the temporal redundancy of periodic signals and in particular, their sparsity in the Fourier domain in order to capture high-speed periodic and quasi-periodic signals. The analysis and reconstruction algorithms presented considered the data at every pixel as independent. In reality, adjacent pixels have temporal profiles that are very similar. In particular (see Figure 2.21), the temporal profiles of adjacent pixels are related to each other via a phase shift which depends upon the local speed and direction of motion of scene features. This redundancy is currently not being exploited in our current framework. We are currently exploring extensions of the CSC, that explicitly model this relationship and use these constraints during the recovery process.

2.7.2 Spatio-temporal resolution trade-off

The focus of this chapter, was on the class of periodic and quasi-periodic signals. One interesting and exciting avenue for future work is to extend the application of the CSC to a wider class of high-speed videos such as high-speed videos of statistically regular dynamical events (e.g., waterfall, fluid dynamics etc) and finally to arbitrary high-speed events such as bursting balloons etc. One alternative we are pursuing in this regard is considering a scenario which allows for spatio-temporal resolution trade-offs, i.e., use a higher resolution CSC in order to reconstruct lower resolution high-speed videos of arbitrary scenes. The spatio-temporal regularity and redundancy available in such videos needs to be efficiently exploited in order to achieve this end.

2.7.3 Conclusions

In this chapter, we present a simple, yet powerful sampling scheme and reconstruction algorithm that turns a normal video camera into a high-speed video camera for periodic signals. We show that the current design has many benefits over traditional approaches and show a working prototype that is able to turn an off-the-shelf 25 fps PointGrey Dragonfly2 camera into a 2000 fps high-speed camera.

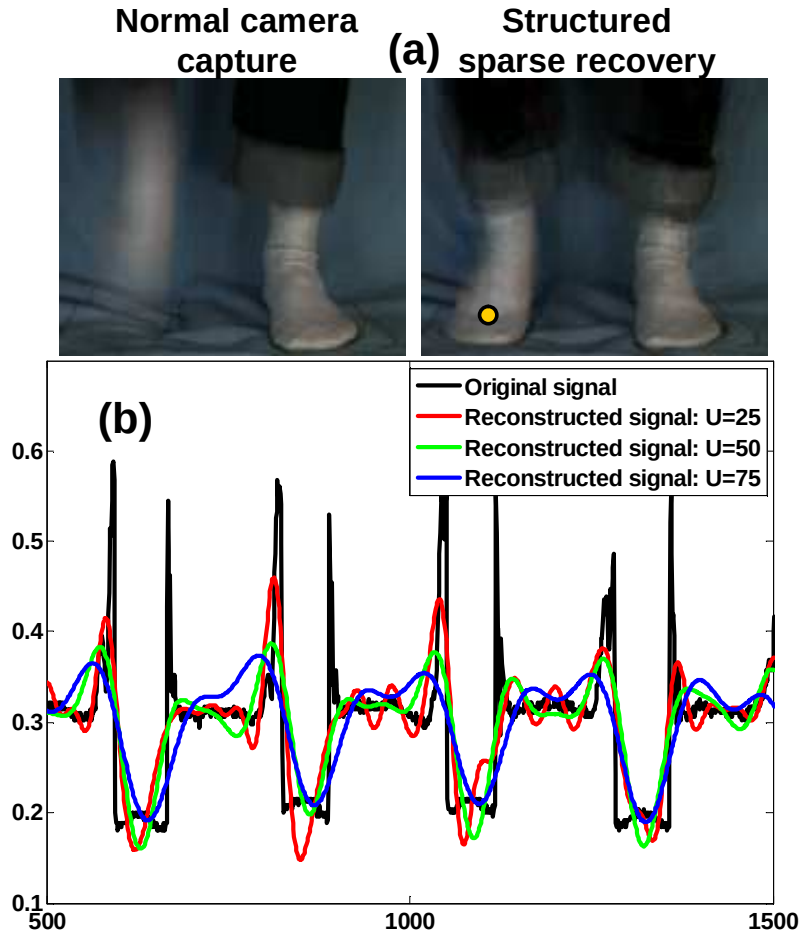

Figure 2.17: Frontal scene of a person jogging-in-place. (a) A frame captured by a normal camera (left) and one of the frames recovered from coded strobing capture at $U = 25$ (right). (b) Plot in time of the pixel (yellow) of the original signal and signal reconstructed from coded strobing capture at $U = 25, 50$ and 75. Note that the low frequency parts of the signal are recovered well compared to the high-frequency spikes.

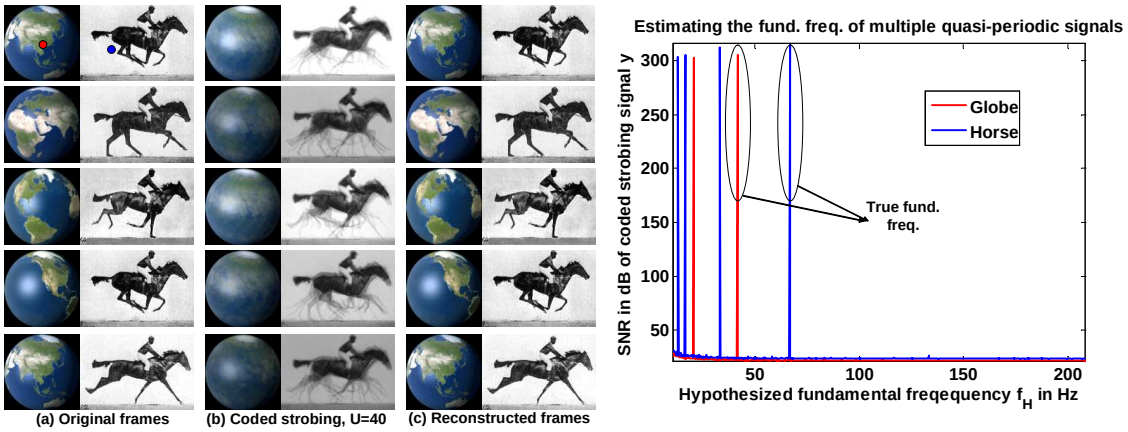

Figure 2.18: Recovery of multiple periodic motion in a scene. (a) Periodic events with different periods in the same scene. The scene as captured by CSC with $U = 40$ is shown in (b). The recovered frames are shown in (c). Shown in (d) is the estimated fundamental frequency of globe and horse at points marked red and blue. Note that the last peak in both globe and horse corresponds to the respective fundamental frequency of 41.667 Hz and 66.667 Hz.

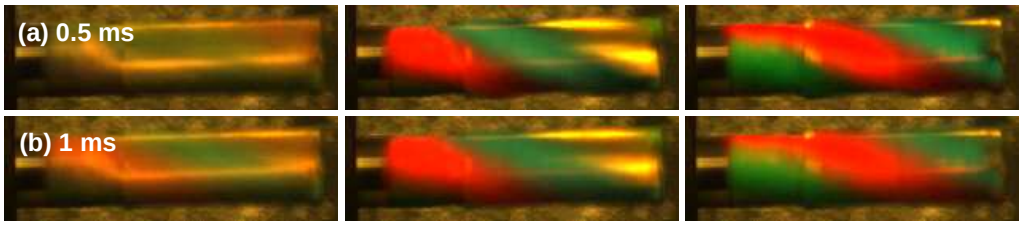

Figure 2.19: Coded strobing reconstructions exhibit blur when the temporal resolution δt is not small enough. Shown in (a) and (b) are the same mill tool rotating at 12000 rpm and captured by a strobe with $\delta t = 0.5$ ms and $\delta t = 1$ ms respectively. The reconstructions shown in the second and third column show that $\delta t = 1$ ms strobe rate is insufficient and leads to blur in the reconstructions.

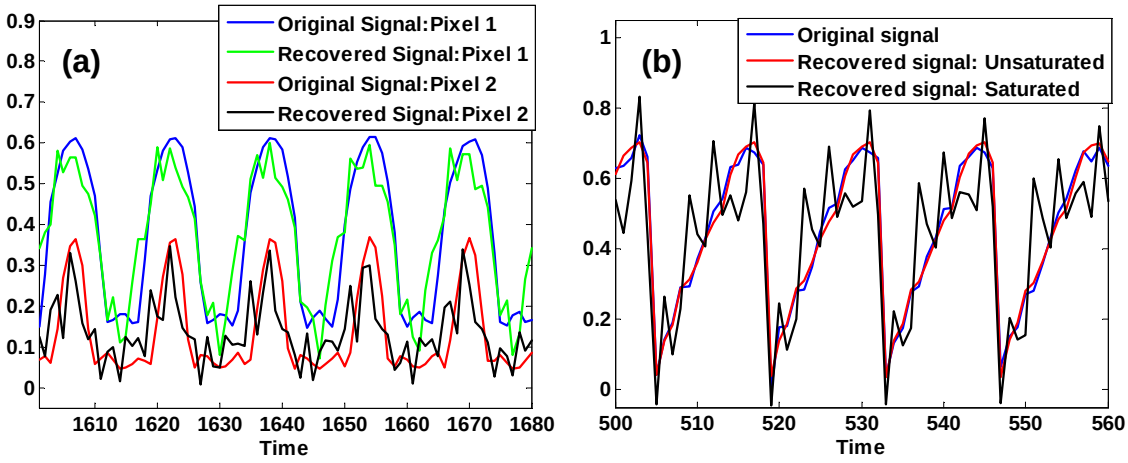

Figure 2.20: (a) Ringing artifacts (in time) in the reconstructed signal at two pixels separated by 8 units in Fig 2.12(c). Also shown are the input signals. Note that the artifacts in reconstruction (in time) manifests as artifacts in space in the reconstructed image. (b) Artifacts in the reconstructed signal due to saturation in the observed signal y.

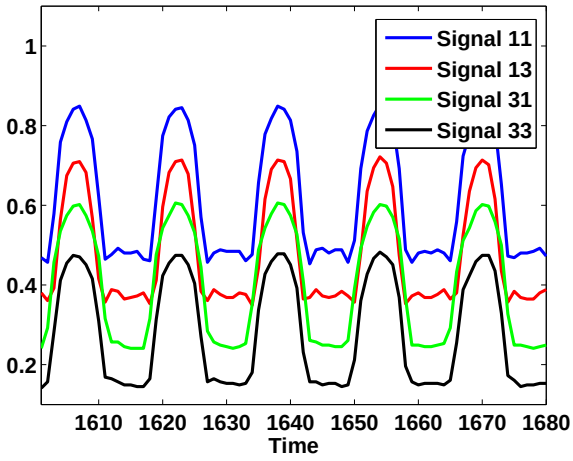

Figure 2.21: The waveforms in a neighborhood are highly similar and hence the information is redundant. Shown are the waveforms of 4 pixels at the corners of a 3×3 neighborhood. The waveforms are displaced vertically for better visualization.

Chapter 3

Programmable Pixel Compressive Camera

3.1 Introduction

Spatial resolution of imaging devices is steadily increasing; mobile phone cameras have $5-10$ megapixels while point-and-shoot cameras have $12-18$ megapixels. But the temporal resolution of video cameras has increased slowly; today's video cameras mostly operate at $30-60$ fps. High-speed video cameras are technically challenging due to high bandwidth and high light efficiency requirements. In this chapter, we present an alternative architecture for acquiring high-speed videos that overcomes both these limitations.

The imaging architecture we present (Figure 3.1), is termed Programmable Pixel Compressive Camera (P2C2). Our camera consists of a normal 25 fps, low resolution video camera, with a high resolution, high frame-rate modulating device such as a Liquid Crystal on Silicon (LCOS) or a Digital Micromirror Device (DMD) array. The modulating device modulates each pixel independently in a predetermined random fashion at a rate higher than the acquisition frame rate of the

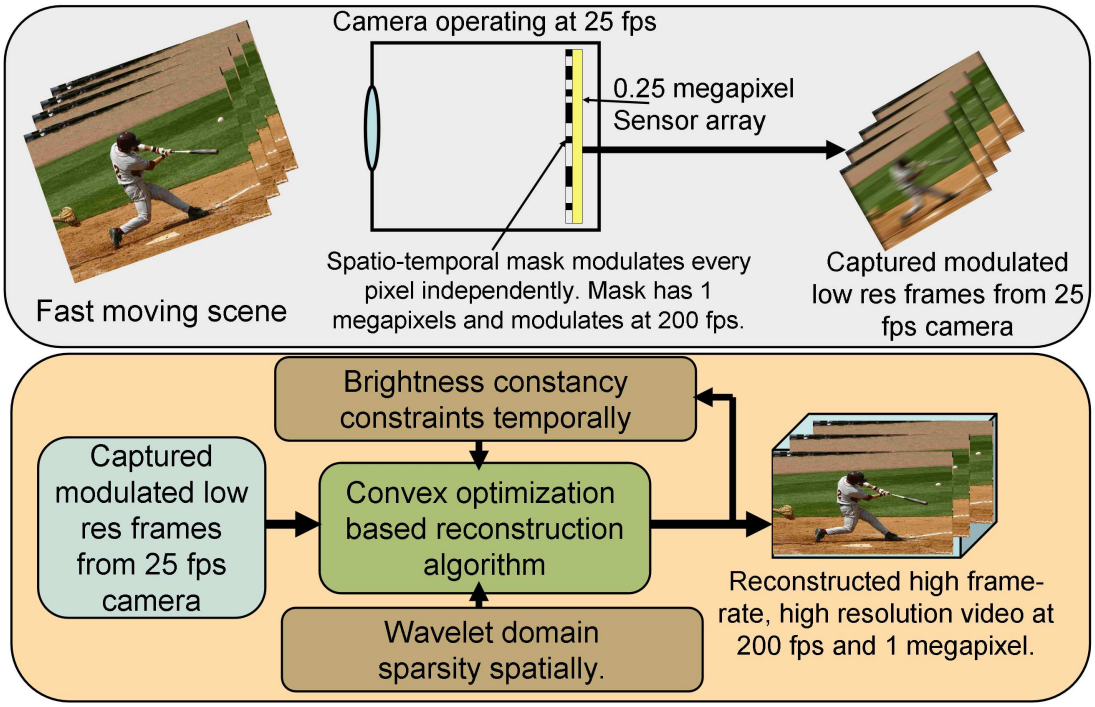

Figure 3.1: Programmable Pixel Compressive Camera (P2C2): Each pixel of a low frame rate, low resolution camera is modulated independently with a fast, high resolution modulator (LCOS or DMD). The captured modulated low resolution frames are used with accompanying brightness constancy constraints and a wavelet domain sparsity model in a convex optimization framework to recover high resolution, high-speed video.

camera. Thus, each observed frame at the camera is a coded linear combination of the voxels of the underlying high-speed video frames. Both low frame-rate video cameras and high frame-rate amplitude modulators (DMD/LCOS) are inexpensive and this results in significant cost reduction. Further, the capture bandwidth is significantly reduced due to P2C2's compressive imaging architecture. The underlying high resolution, high-speed frames are recovered from the captured low resolution frames by exploiting temporal redundancy in the form of brightness constancy and spatial redundancy through transform domain sparsity in a convex optimization framework.

3.1.1 Contributions:

- We present a new imaging architecture 'P2C2' for compressive acquisition of high-speed videos. P2C2 allows temporal super-resolution of videos with no appreciable loss in spatial resolution.

- We show that the brightness constancy constraint significantly improves video reconstruction. Our algorithm reconstructs high-speed videos from low frame rate observations over a broad range of scene motions.

- We characterize the benefits and limitations of P2C2 through experiments on high-speed videos. We implement a prototype P2C2 and acquire 200 fps videos of challenging scenes using a 25 fps video camera.

3.2 Related Work

High speed sensors: Traditional high-speed cameras are expensive due to requirement of high light sensitivity and large bandwidth. Usually these cameras [1] have limited on-board memory with a dedicated bus connecting the sensor. The acquisition time is limited by the on-board memory. For example, FastCam SA5 (a $\$300K$ high-speed camera) can capture atmost 3 seconds of video at 7500 fps and 1 megapixel. Though most videos have significant spatio-temporal redundancy, current high-speed cameras do not exploit them. Our camera allows us to exploit this, thereby reducing the capture bandwidth significantly. Further, existing cameras use specialized sensors with high light sensitivity and image intensifiers to ensure

each frame is above the noise bed. In contrast, P2C2 captures a linear combination of video voxels, thereby naturally increasing the signal-to-noise ratio and partially mitigating the need for image intensifiers.

Temporal super-resolution: Shechtman et al. [136] perform spatio-temporal super-resolution by using multiple cameras with staggered exposures. Similarly, Wilburn et al. [160] use a dense 30 fps camera array to generate a 1000 fps video. Recently Agrawal et al. [4] showed that combining this idea with per camera flutter shutter (FS) [123] significantly improves the performance of such staggered multi-camera high-speed acquisition systems. While these systems acquire high-speed videos, they require multiple cameras with accurate synchronization and their frame-rate scales linearly with number of cameras. In contrast, we increase temporal resolution without the need for multiple cameras and also our camera is not restricted to planar scene motion. Ben-Ezra [15] built a hybrid camera where motion is measured using an additional higher frame rate sensor and then used to estimate the point spread function for deblurring. We estimate both motion and appearance from the same sensor measurements.

Video interpolation: Several techniques exist for frame-rate conversion [134]. Recently, [89] showed that explicit modeling of occlusions and optical flow in the interpolation process allows us to extract 'plausible' interpretations of intermediate frames.

Motion deblurring: When a fast phenomenon is acquired via a low frame-rate camera one can either obtain noisy and aliased sharp images using short exposure, or blurred images using long exposures. Motion deblurring has made great

progress by incorporating spatial regularization terms within the deconvolution framework [133][51]. Novel hardware architectures [123][82] have also been designed to improve deconvolution. These techniques require the knowledge of motion magnitude/direction and cannot handle general scenes exhibiting complex motion. In contrast, P2C2 can handle complex motion without the need for any prior knowledge.

Compressive sensing (CS) of videos: Existing methods for video CS assume multiple random linear measurements are available at each time instant either using a coded aperture [94] or a single pixel camera (SPC) [44]. [150] shows that videos with slowly changing dynamics need far fewer measurements for subsequent frames once the first frame is recovered using standard number of measurements. [113] presents an algorithm for compressive video reconstruction by using a motion compensated wavelet basis to sparsely represent the spatio-temporal volume. Such methods have achieved only moderate success since (a) the temporal redundancy of videos is not explicitly modeled and (b) the hardware architectures need to be highly engineered and/or are expensive.

In [153], the authors extend FS to videos and build a high-speed camera for periodic scenes. For the class of video that can be adequately modeled as a linear dynamical system [129] provides a method for compressively acquiring videos using the SPC architecture. Both approaches can handle only periodic/dynamic texture scenes while P2C2 can capture arbitrary videos.

Spatio-temporal trade-off: Gupta et al. [59] show how per-pixel temporal modulation allows flexible post-capture spatio-temporal resolution trade-off. The

method loses spatial resolution for moving elements of the scene, whereas our method preserves spatial resolution while achieving higher temporal resolution. Similarly, Bub et al. [21] propose spatio-temporal trade-off of captured videos but has limited light throughput and unlike [59] lacks flexible resolution trade-off. Gu et al. [58] proposed a coded rolling shutter architecture for spatio-temporal trade-off.

Per-pixel control: Nayar et al. [106] propose a DMD array based programmable imager for HDR imaging, feature detection and object recognition. [59, 21] use DMD array based per-pixel control for spatio-temporal resolution trade-off. Similarly, DMD arrays were used in [127, 126] for phase analysis and shape measurement. While the idea of per-pixel modulation is not new, we propose a sophisticated spatio-temporal modulation using P2C2 for high-speed imaging. Such modulation allows us to achieve higher temporal resolution without loss in spatial resolution.

3.3 Imaging Architecture

Let the intensity of desired high frame rate video be $x(s,t)$ where $s = (r,c) \in [1\ N] \times [1\ N]$ are the row and column coordinates respectively and $t \in [1\ T]$ the temporal coordinates. We term the higher rate frames x_t as 'sub-frames' since the acquired frames are formed by integrating them. Our camera captures the modulated intensities $y(s_l, t_l)$ where $s_l = (r_l, c_l) \in [1\ N/L_s] \times [1\ N/L_s]$ and $t_l \in [1\ T/L_t]$ are its spatial and temporal coordinates. L_s and L_t are the spatial and temporal sub-sampling factors respectively. The captured frame y_{t_l} is related to

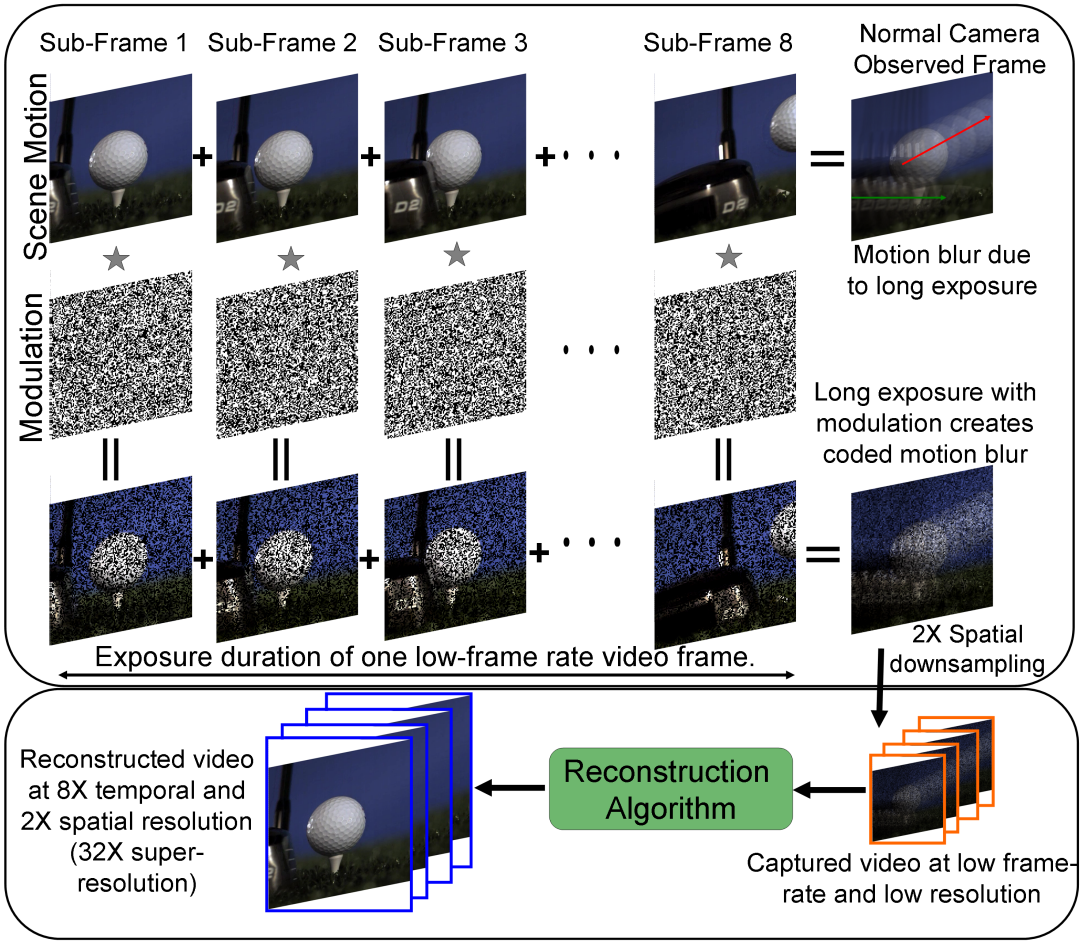

Figure 3.2: Camera architecture: At every pixel the camera independently modulates the incoming light at sub-frame durations and then integrates it. For example, the 3-D spatio temporal volume of a golf ball is modulated with a random mask at sub-frame durations and then integrated into a frame. A frame captured by our camera has the code embedded in the blur.

sub-frames x_t as

$$y_{t_l} = D\left(\sum_{t=(t_l-1)L_t+1}^{t_l L_t} x_t \phi_t\right) \tag{3.1}$$

where ϕ is the spatio-temporal modulation function (achieved by LCOS as shown in Figure 3.3) and $x_t\phi_t$ is modulation of sub-frame at t with mask at t. $D(\cdot)$ denotes a spatial subsampling operation to account for the possibility that camera could

also be spatially modulated at sub-pixel resolution. Notice that L_t sub-frames are modulated with L_t independent high resolution random masks and then integrated to produce one spatio-temporally subsampled frame of captured video (as shown in Figure 3.2). We limit our discussion mostly to temporal downsampling. Nevertheless, the architecture and recovery algorithm presented later easily extend to spatial subsampling as well and we illustrate it through results in experimental section.

Since the observed pixel intensities y are linear combinations of the desired voxels x, with the weights given by modulation function ϕ, the equation (3.1) can be written in matrix-vector form as,

$$\mathbf{y} = \mathbf{\Phi x} \tag{3.2}$$

where $\mathbf{\Phi}$ is the matrix representing per pixel modulation followed by integration in time and spatial sub-sampling. $\mathbf{x}$ and $\mathbf{y}$ are the vectorized form of desired high-speed voxels x (eg., $256 \times 256 \times 32$ voxels) and the captured video y ($128 \times 128 \times 4$ video) respectively. The optimization term enforcing fidelity of the recovered sub-frames to the captured frames is given by $E_{data} = \|\mathbf{y} - \mathbf{\Phi x}\|_2^2$.

3.3.1 Prototype P2C2

We realize P2C2 with an LCOS mirror SXGA-3DM from Forth Dimension Displays as a spatio-temporal modulator. The mirror has 1280×1024 pixels and each pixel can be fluttered (binary) independently at maximum rate of 3.2 kHz. This imposes an upper limit of 3200 fps on frame-rate of the recovered video. LCOS works by flipping the polarization state of incoming light and therefore needs to be

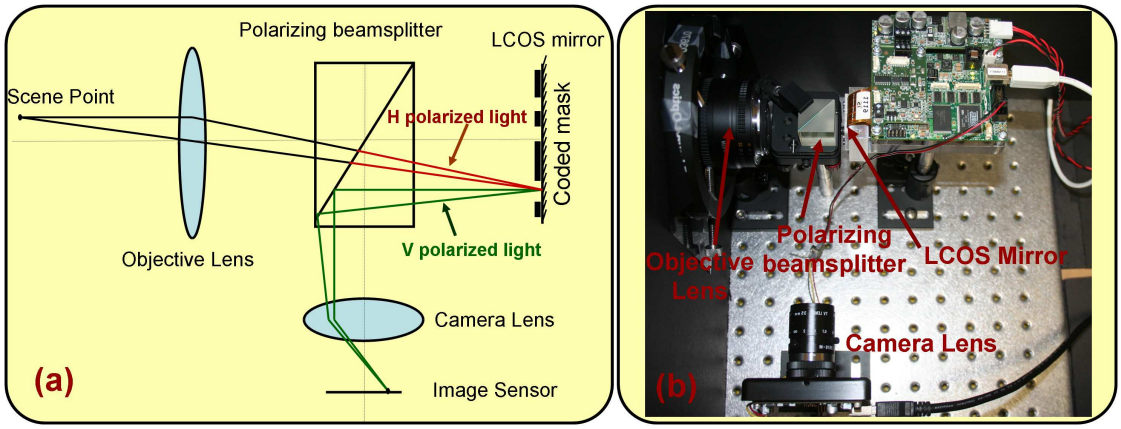

Figure 3.3: Prototype: Illustration of the optical setup.

used with a polarizing beam-splitter and necessary relay optics as shown in Figure 3.3. The scene is focused on LCOS device which modulates this incoming light. The Pointgrey Dragonfly2 sensor (1024×768 pixels at 25 fps) is in turn focused on LCOS mirror. An LCOS modulator offers a significantly higher contrast ratio (> 100) compared to off-the-shelf graphic LCD attenuators. Further, the primary advantage of LCOS mirror over LCD arrays is the higher fill factor of pixels. LCOS based per-pixel control was used by Mannami et al. [93] for recovering high dynamic range images.

Related architectures: The architecture of P2C2 is a generalization of previous imaging architectures proposed for high-speed imaging and motion deblurring. For example, flutter shutter (FS) camera [123] is a special case where all the pixels have the same shutter. P2C2 adopts a random spatio-temporal modulation and is a generalized version of architectures for spatio-temporal resolution trade-off [8, 4, 7].

P2C2 is a compressive imaging system and is related to SPC [44]. In P2C2 the

mixing of voxel intensities is localized in space and time as opposed to SPC which aims for global mixing of underlying voxel intensities. Our architecture also exploits the cost benefit of current sensors (especially in visible wavelength) by using a pixel array in place of single pixel detector.

3.4 High speed video recovery

Since the number of unknown pixel intensities is much larger than available equations, (3.2) is severely under-determined. To solve for sub-frames $\mathbf{x}$, a prior on spatio-temporal video volume should be incorporated.

Most natural video sequences are spatio-temporally redundant. Spatially, images are compressible in transform basis such as wavelets and this fact is used in image compression techniques such as JPEG2000. Temporally, object and/or camera motion preserves the appearance of objects in consecutive frames and this fact is used in video compression schemes such as MPEG. We exploit both forms of redundancy to solve the system of under-determined equations (3.2) and recover the high-speed sub-frames.

3.4.1 Transform domain sparsity

Each sub-frame is sparse in appropriate transform domain and we enforce this property in our recovery through ℓ_1 regularization of its transform coefficients. The regularization term enforcing spatial-sparsity of sub-frames is $E_{spatial} = \sum_{t=1}^{T} \beta \|\mathbf{\Psi}^{-1}\mathbf{x}_t\|_1$, where $\mathbf{x}_t$ is the vectorized sub-frame x_t and $\mathbf{\Psi}$ the transform basis.

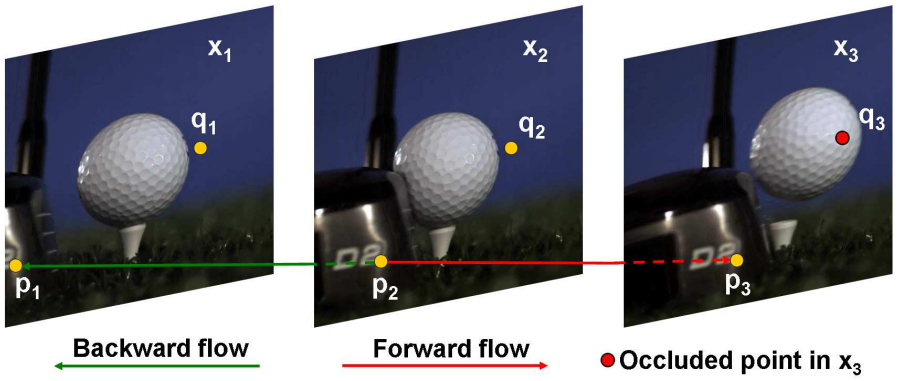

Figure 3.4: Brightness constancy constraints at p_1, p_2 and p_3 and OF consistency check at q_2 and q_3.

3.4.2 Brightness constancy as temporal redundancy

Unlike spatial redundancy, temporal redundancy in videos is not easily amenable to sparse representation in a transform basis. Hence, regularization of 3-D transform basis coefficients to solve the under-determined system in (3.2) results in poor reconstruction quality. To overcome this challenge, we propose to keep the temporal regularization term distinct from spatial regularization. We exploit the brightness constancy(BC) constraint in temporal direction. This constraint is distinct from and in addition to the spatial transform domain sparsity regularization.

Consider three consecutive frames of a club hitting the ball in Figure 3.4. The points p_1, p_2 and p_3 correspond to the same point on the golf club in frames x_1, x_2 and x_3 respectively. If the relative displacement of these points is estimated, then their pixel intensities in (3.2) can be constrained to be equal i.e. brightness at these pixels is constant $x(p_2, 2) - x(p_1, 1) = 0$ (backward flow) and $x(p_2, 2) - x(p_3, 3) = 0$ (forward flow). This effectively decreases the number of unknowns by 2. The system becomes significantly less under-determined if BC constraints at other points are

known as well. The sub-frame BC constraints over entire video volume are then given by

$$\mathbf{\Omega x} = 0 \tag{3.3}$$

where every row of matrix $\mathbf{\Omega}$ is the relevant BC equation of a spatio-temporal point (s,t). We incorporate these constraints in the optimization by adding a BC regularization term $E_{BC} = \lambda \|\mathbf{\Omega x}\|_2^2$.

To enforce the BC constraint at any spatio-temporal point (s,t), we first estimate the optical flow (OF) at sub-frame x_t in forward direction (u_t^f, v_t^f). Then we perform a consistency check by estimating the backward flow (u_{t+1}^b, v_{t+1}^b) at sub-frame x_{t+1}. Such a consistency check not only detects points of x_t occluded in sub-frame x_{t+1}, but also prunes the untrustworthy flow in (u_t^f, v_t^f). For example, consider points q_1 and q_2 on a blue background in Figure 3.4. Both points have same spatial coordinates and have the same intensity $x(q_2, 2) - x(q_1, 1) = 0$. The fact that both q_1 and q_2 are same points in the scene (here background) is established solely from OF by performing the following consistency check: q_1 goes to q_2 according to forward OF and q_2 comes back to q_1 in the backward OF. On the other hand $x(q_2, 2) \neq x(q_3, 3)$ even though $q_2 = q_3$. This is because forward OF suggests q_2 is same as q_3 since q_2 belongs to background and has 0 flow. But the backward OF at q_3 is non-zero and hence $q_3 + (u^b(q_3, 3), v^b(q_3, 3)) \neq q_2$. This implies that point q_2 is occluded and/or has unreliable forward OF $(u^f(q_2, 2), v^f(q_2, 2))$. This means the consistency doesn't check at q_2 i.e. $o^f(q_2, 2) = 0$ whereas it checks at q_1 i.e. $o^f(q_1, 1) = 1$. The BC constraint is enforced only when consistency checks. We perform the consistency check in backward direction as well by checking the con-

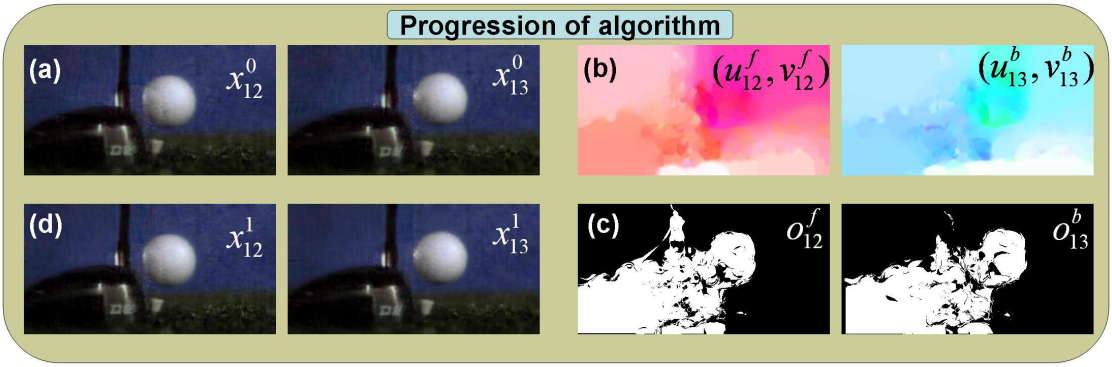

Figure 3.5: In clockwise direction (a) two sub-frames from the initialization (b) forward and backward OF at respective sub-frames (c) corresponding forward and backward consistency map (d) sub-frames from next iteration incorporate BC constraints only at white pixels from (c).

sistency between (u_t^b, v_t^b) and (u_{t-1}^f, v_{t-1}^f). The process of pruning OF is illustrated in Figure 3.5. The sub-frames estimated in first iteration of our algorithm (Figure 3.5a) are used to determine E_{BC} for the next iteration. The results of next iteration are shown in Figure 3.5d.

The importance of brightness constancy in video recovery is illustrated in Figure 3.6. The third column shows reconstruction fidelity obtained by assuming only spatial sparsity. The fourth column shows our reconstruction which incorporates explicit brightness constancy (BC) constraints. This significantly improves reconstruction since the algorithm adapts to the complexity of motion in a particular video.

3.4.2.1 Recovery Algorithm

Initialization: Given optical flow, the BC constraints are incorporated through E_{BC}. But OF can be determined only when the sub-frames are available. Hence,

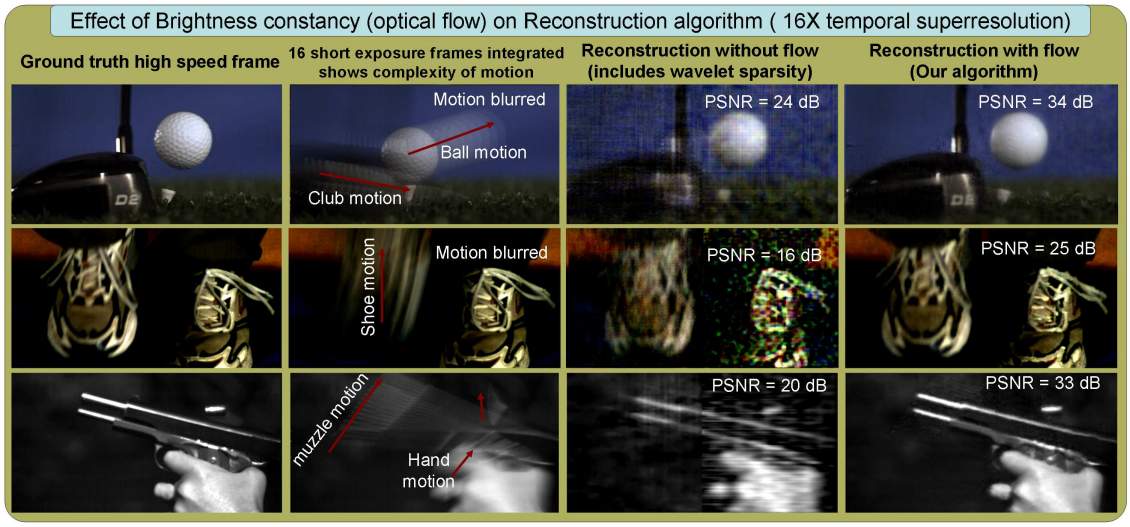

Figure 3.6: Importance of Brightness Constancy: All results are 16X temporal super-resolution. Shown are the original high-speed frames, motion blurred frames and reconstructions with and without BC. Notice the huge improvement in reconstruction SNR due to BC. The results in column 4 and its necessary OF were computed in an alternating fashion using an iterative procedure on the observations. OF was not assumed to be available. 'Golf' and 'Recoil' high-speed video credit TECH IMAGING.

we iteratively determine the sub-frames and the optical flow in an alternating fashion. We begin by estimating the sub-frames without any BC constrains. In the first iteration we trade-off spatial resolution to recover frames at desired temporal resolution. We assume that each sub-frame x_t is an upsampled version of a lower spatial resolution frame: $\mathbf{x}_t = U(\mathbf{z}_t)$ where $\mathbf{z}_t$ is a vectorized $[\frac{N}{L_s\sqrt{L_t}} \times \frac{N}{L_s\sqrt{L_t}}]$ image and $U(\cdot)$ is a linear upsampling operation such as bilinear interpolation. The initial estimate is given by solving

$$\mathbf{z}^0 = \arg\min \sum_{t=1}^{T} \beta \|\mathbf{\Psi}^{-1} U(\mathbf{z}_t)\|_1 + \|\mathbf{y} - \mathbf{\Phi} U(\mathbf{z})\|_2^2. \tag{3.4}$$

The estimate $\mathbf{x}^0 = U(\mathbf{z}^0)$ doesn't capture all the spatial detail and is noisy but it preserves the motion information accurately as shown in Figure 3.5a. We estimate OF [86] on initial estimate (Figure 3.5b) and perform consistency check to

prune the flow (Figure 3.5c) as described in section 3.4.2. Only consistent flow is used to enforce the BC constraint for the next iteration.

Optimization: We minimize the total energy function which also includes the term E_{BC} built using the matrix $\mathbf{\Omega}^{k-1}$ from the previous iteration.

$$\mathbf{x}^k = \arg\min \sum_{t=1}^{T} \beta \|\mathbf{\Psi}^{-1}\mathbf{x}_t\|_1 + \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2^2 + \lambda \|\mathbf{\Omega}^{k-1}\mathbf{x}\|_2^2 \tag{3.5}$$

The above optimization problem is convex but even for a moderate sized video of $256X256$ pixels and 32 frames, the variable $\mathbf{x}$ is 2 million large. To solve the optimization problem we use a fast algorithm designed for large systems, based on fixed point continuation [61]. In all our experiments we fix the parameters at $\beta = 10^{-5}$ and $\lambda = 10^{-1}$. In practice, our algorithm converges in 5 iterations.

3.5 Experimental Results

We rigorously evaluate the performance and reconstruction fidelity on several challenging datasets. First, we simulate P2C2 in software by capturing fast events with a standard high-speed camera.

3.5.1 Simulation on high speed videos

Figure 3.6 shows example reconstructions of high-speed sub-frames at 16X temporal super-resolution. Notice that while normal camera frames are highly blurred, the reconstruction retains sharpness and high frequency texture detail is maintained. Several of our examples contain complex and non-linear motion. Most examples also contain several objects moving independently causing occlusion

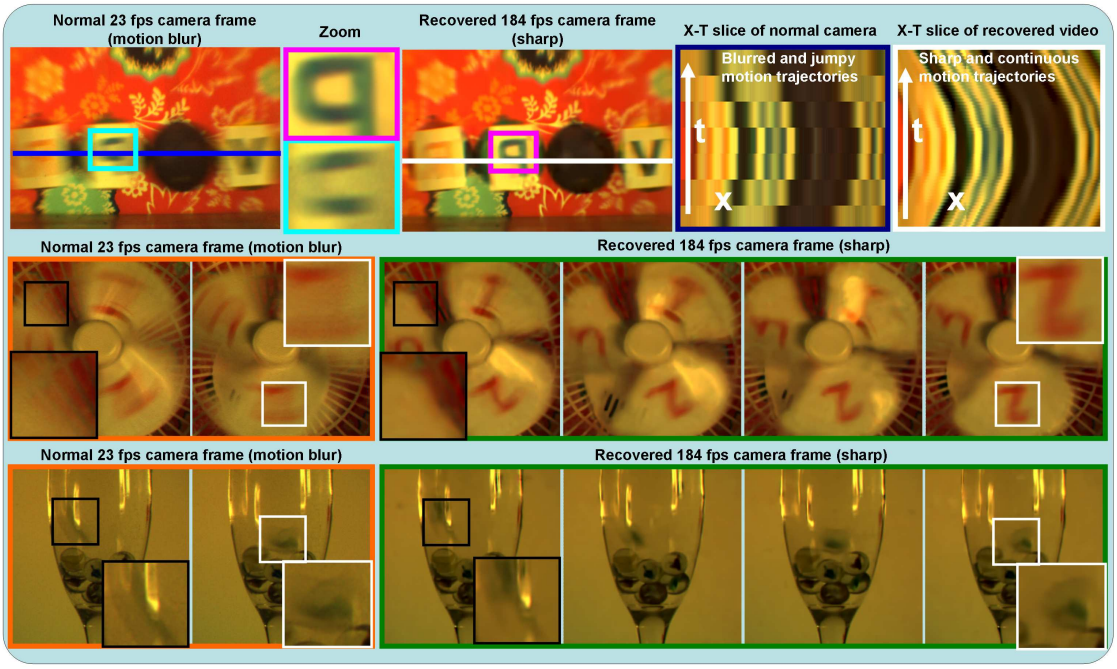

Figure 3.7: Results on LCOS prototype: For the dataset in the top row, one frame from a normal 23 fps camera and our recovered video with zoom-in insets are shown. The fourth and fifth column shows the X-T slices of the original and recovered videos. For the middle and bottom datasets, two images from normal 23 fps camera and four recovered images are shown.

and disocclusions. To better understand the quality of reconstruction with varying spatio-temporal compression factors, examine Figure 3.8. This video has highly complex motion, where different dancers are performing different motions. There is a significant non-rigidity in motion and challenging occlusion effects. Notice that our reconstruction retains high fidelity even at high compression factors. Even a compression factor of $4 \times 4 \times 4 = 64$ produces acceptable visual quality and $24dB$ PSNR.

Figure 3.8: Effect of spatio-temporal upsampling factors on Dancers video. Notice that our reconstructions retain visual fidelity even in the presence of complex non-rigid multi-body motions and occlusions. High-speed video credit TECH IMAGING.

3.5.2 Results on P2C2 prototype datasets

We captured several datasets using our prototype device. The camera was operated at 23 fps and 8 different masks were flipped during the integration time of sensor. This allows us to reconstruct the sub-frames at a frame rate of 184 fps (23×8). We note that in our experimental setup we were limited by the field of view since the beamsplitter size forced us to use a lens with a large focal length .

In Figure 3.7, we show three different datasets. In the pendulum dataset, four letters 'CVPR' were affixed to four out of five balls and the pendulum was swung. As shown in X-T slices the balls and the letters had significant acceleration and also change in the direction of motion. The recovered frames are much sharper than the original 23 fps frames as shown in inset. Note that the characters are much clearer in the reconstructed frame despite a 40 pixel blur in each captured frame. Also, the X-T slice clearly illustrates the reconstruction quality. On the other hand, a fine feature such as the thread is blurred out since the flow corresponding to it is hard to recover.

Next, we rotate a fan and capture it at 23 fps and reconstruct sharper and clearer frames at 184 fps as shown in the white inset. During recovery, we do not assume that motion is rotational. Note that the normal camera frame has intensities integrated from both fan blade and the background mesh as shown in the black inset. We can handle this sub-frame occlusion in our recovery as indicated by the clear background and foregrounds in the recovered frame.

Finally, we drop a marble in water and capture it at 23 fps and reconstruct at 184 fps. Again, we do not assume any motion path but still recover the curved path of the marble. Note that, despite specularities in the scene our algorithm is robust.

3.6 Analysis

Choice of modulation masks: There are two requirements on modulation masks to obtain high fidelity reconstruction. Firstly, the temporal code at a given

pixel should have a broadband frequency response [123], such that none of the scene features are blurred irrevocably. Secondly, the temporal code at a local neighborhood of pixels should be different. This along with spatial smoothness assumption provides sufficient constraints in a neighborhood to recover the low resolution sub-frames during the initialization process (3.4). This initialization is important to extract optical flow estimates which are then propagated forward using the iterative framework. On the other hand, when brightness constancy constraints are available, a modulation mask with well-conditioned matrix is desirable. Given ground-truth BC constraints, we reconstruct sub-frames of Golf dataset at 16X temporal super-resolution under three different masks (Table 3.1). We see that random mask of P2C2 offers significant advantage over the 'all one' mask and flutter shutter. We believe that proper theoretical analysis will lead to the design of optimal modulation masks and this is an area of future work.

	P2C2	**'All one'**	**FS**
PSNR in dB	26.2	21	16

Table 3.1: Reconstructing Golf dataset at 16X temporal super-resolution with different masks with ground truth BC constraints.

Comparison with prior art: We compare P2C2 with flexible voxels (FV) [59] on a fast phenomenon shown in Figure 3.9. Flexible voxels reconstruction suffers from two disadvantages: spatial smoothness is introduced in moving parts of the scene leading to blurred features and since the coding sequence for flexible voxels is mostly zeros, it leads to a highly light-inefficient capture method leading

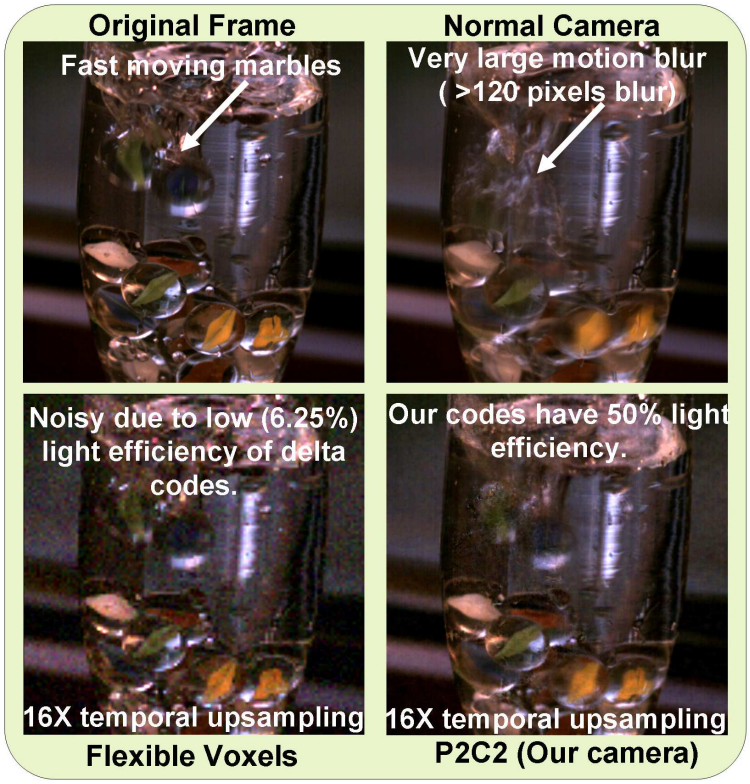

Figure 3.9: One frame of a video sequence of marble dropped in water. 40dB sensor noise was added. Our reconstruction is less noisy (zoom for better view) than those of flexible voxels due to higher light efficiency of P2C2.

to performance degradation in the presence of noise. In 16× temporal upsampling example shown in Figure 3.9, the high temporal resolution reconstruction of FV is noisier than our reconstruction.

Effect of spatio-temporal upsampling: To evaluate the impact of varying upsampling factors, we perform statistical evaluation of sub-frame reconstructions using P2C2 on several challenging high-speed videos. These videos have very different spatial and motion characteristics. We carefully selected the dataset to ensure that it spans a large range of videos in terms of spatial texture, light level, motion magnitude, motion complexity, number of independent moving objects, specularities, varying material properties. Shown in Figure 3.10 is a plot of reconstruction

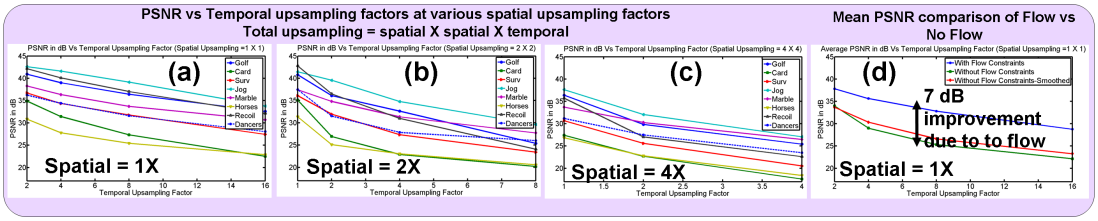

Figure 3.10: PSNR vs compression factors. (a) Spatial compression is kept at 1 and temporal compression is varied (b) Spatial compression is 2 in both dimensions. (c) Spatial compression is 4 in both dimensions. Notice that the video reconstruction fidelity remains high (PSNR>30dB) even at total compression factors of $16 - 32$. (d) Brightness Constancy constraints significantly improves the reconstruction.

PSNR (in dB) as a function of spatial and temporal upsampling for various datasets. From our visual inspection we note that reconstructions with PSNR of 30dB or greater have sufficient textural sharpness and motion continuity to be called good quality reconstructions. From the figure, it is clear that we can achieve $8 - 16$X temporal upsampling and retain reconstruction fidelity. Also, when we perform 32X spatio-temporal super-resolution using P2C2 (2X2X8 or 4X4X2) we obtain acceptable reconstruction fidelity. Few frames from the reconstructions and their corresponding PSNR values are also shown in Figure 3.6 and 3.8 to relate visual quality to PSNR.

Benefits: Our imaging architecture provides three advantages over conventional imaging architectures. It significantly reduces the bandwidth requirement at the sensor by exploiting the compressive sensing paradigm. It improves light throughput of the system compared to acquiring a short exposure low frame-rate video and allows acquisition at low light levels. These are significant advantages since the prohibitive cost of high-speed imagers, is essentially due to the requirement for high bandwidth and high light sensitivity. Finally, the imaging architecture

is flexible allowing incorporation of several other functionalities including high dynamic range (HDR) [93], assorted pixels [163] and flexible voxels [59].

Limitations: P2C2 exploits spatio-temporal redundancy in videos. Scenes such as a bursting balloon cannot be directly handled by the camera. Since the spatio-temporal redundancy exploited by traditional compression algorithms and our imaging architecture are very similar, as a thumb rule one can assume that scenes that are compressed efficiently can be captured well using our method. Our prototype uses a binary per-pixel shutter and this causes a 50% reduction in light throughput. Since most sensors already have the ability to perform 'dual mode' integration (i.e., change the gain of pixels) we imagine the possibility of non-binary modulation in future. The algorithm is not real-time and this precludes the direct-view capability.

Summary: We presented Programmable Pixel Compressive Camera, a new imaging architecture for high-speed video acquisition, that (a) reduces capture bandwidth and (b) increases light efficiency compared to related works. We also highlighted the importance of explicitly exploiting the brightness constancy constraints.

Chapter 4

Compressive Background Subtraction and Tracking

4.1 Introduction

Presently, images are fully sampled using either charge-coupled device (CCD) or CMOS technologies but are compressed to a smaller size after capture. This sampling process is inexpensive for imaging at visible wavelengths as the conventional devices are built from silicon, which is sensitive to these wavelengths; however, if sampling at other optical wavelengths is desired, it becomes quite expensive to obtain estimates at the same pixel resolution as new imaging materials are needed. For example, a camera with an array of infrared sensors can provide night vision capabilities but can also cost significantly more than the same resolution CCD or CMOS camera.

Recently, a prototype single pixel camera (SPC) was built based on the new mathematical theory of *compressive sensing* (CS) [156]. The CS theory states that a signal can be perfectly reconstructed, or can be robustly approximated in the presence of noise, with sub-Nyquist data sampling rates, provided it is *sparse* in some

linear transform domain [24, 43]. That is, it has K nonzero transform coefficients with $K \ll N$, where N is the dimension of the transform space. For computer vision applications, it is known that natural images can be sparsely represented in the wavelet domain [91]. Then, according to the CS theory, by taking random projections of a scene onto a set of test functions that are incoherent with the wavelet basis vectors, it is possible to recover the scene by solving a convex optimization problem. Moreover, the resulting *compressive measurements* are robust against packet drops over communication channels with graceful degradation in reconstruction accuracy, as the image information is fully distributed.

Compared to conventional camera architectures, the SPC hardware is specifically designed to exploit the CS framework for imaging. An SPC fundamentally differs from a conventional camera by (i) reconstructing an image by using only using a single optical photodiode (infrared, hyperspectral, etc.) along with a digital micromirror device (DMD), and (ii) combining the sampling and compression into a single nonadaptive linear measurement process. An SPC can directly scale from the visual spectra to hyperspectral imaging with only a change of the single optical sensor. Moreover, enabled by the CS theory, an SPC can robustly reconstruct the scene from much fewer measurements than the number of reconstructed pixels which define the resolution, given that the image of the scene is compressible by an algorithm such as the wavelet-based JPEG 2000.

Conventional cameras can also benefit by processing in the compressive sensing domain if their data is being sent to a central processing location. The naïve approach is to transmit the raw images to the central location. This exacerbates

the communication bandwidth requirements. In more sophisticated approaches, the cameras perform motion compensation and then code the video. This requires an even smaller communication bandwidth than the compressive samples. However, the embedded systems needed to perform video coding are power hungry and expensive. In contrast, the compressive measurement process only requires cheaper embedded hardware to calculate inner products with a previously determined set of test functions to transmit information at the compressibility rate of the image. They trade off expensive embedded intelligence for more computational power at the central location, which reconstructs the images and is assumed to have unlimited resources.

Hence, compressive cameras are an interesting proposition for computer vision tasks such as tracking and surveillance in both hyper-spectral regime and multi-camera scenarios. For applications in computer vision such as surveillance, teleconferencing and 3-D modeling [39], background subtraction [47, 119] is fundamental in automatically detecting and tracking moving objects. Usually, the foreground or *the innovation* of interest occupies a sparse spatial support as compared to the background and may be caused by the motion and the appearance change of objects within the scene. By obtaining the object silhouettes on a single image plane or multiple image planes, a background subtraction algorithm can be performed.

An interesting intellectual challenge arises when one desires to directly reconstruct the sparse foreground innovations within a scene without any intermediate image reconstruction. The main idea is that the background subtracted images can be represented sparsely in the spatial image domain and hence the CS recon-

struction theory should be applicable for directly recovering the foreground. For natural images, we use wavelets as the transform domain. Pseudo-random matrices provide an incoherent set of test functions to recover the foreground image. Then, the following questions surface (i) how can we detect targets without reconstructing an image? and (ii) how can we directly reconstruct the foreground without reconstructing auxiliary images?

In this chapter, we describe a method to directly recover the sparse innovations (foreground) of a scene. We first show that the object silhouettes (binary background subtracted images) can be recovered as a solution of a convex optimization or an orthogonal matching pursuit problem. In our method, the object silhouettes are learned directly using the compressive samples without any auxiliary image reconstruction. We then discuss simultaneous appearance recovery of objects using the compressive measurements. In this case, we show that it may be necessary to reconstruct one auxiliary image. To demonstrate the performance of the proposed algorithm, we use field data captured by a compressive camera and provide background subtraction results. Further, we simulate multiple distributed compressive cameras and provide a method for 2D tracking and 3D voxel reconstruction. By assuming that camera geometry is known we treat the multi-view 2D tracking and 3D voxel reconstruction problem as that of estimating the sparse support of the object location. Our approach is similar to the multi-view tracking method proposed in [74]. But in our formulation the sparse support of the object is related to the estimated compressive foreground corresponding to multiple compressive cameras observing the scene. By estimating the sparse support using the

foreground information from all cameras, we recover the object occupancy.

While the idea of performing background subtraction in compressed image domain is not novel, there exist no cameras that directly record MPEG videos. Both Aggarwal et al. and Lamarre and Clark perform background subtraction on a MPEG-compressed video using the DC-DCT coefficients of I frames, limiting the resolution of the BS images by 64 [2, 78]. Our technique is tailored for CS imaging, and not compressed video files. Lamarre et al. and Wang et al. use DCT coefficients from JPEG images and MPEG videos, respectively for representation [78, 158]. Toreyin et al. similarly operate on the wavelet representation [143]. These methods implicitly perform decompression by working on every DCT/wavelet coefficient of every image. We never have to go to the high-dimensional images or representations during background subtraction, making our approach particularly attractive for embedded systems and demanding communication bandwidths. Compared to the eigenbackground work in [109], we use random projections which are universal and need no update. In our work the only basis needed is the sparsity basis for difference images, hence no training is required. The very recent work of Uttam, Goodman and Neifeld [146] considers background subtraction from adaptive compressive measurements, with the assumption that the background-subtracted images lie in a low-dimensional subspace. While this assumption is acceptable when image tiling is performed, background subtracted images are sparse in an appropriate domain, spanning a union of low-dimensional subspaces rather than a single subspace.

Our specific contributions are as follows:

1. We present a background subtraction algorithm for compressive cameras. We treat the foreground estimation as a sparse signal recovery problem where convex optimization and greedy methods can be applied. We employ Basis Pursuit Denoising methods [37] as well as total variation minimization [24] as convex objectives to process field data.

2. We show that it is possible to recover the silhouettes of foreground objects by learning a low-dimensional compressed representation of the background image. Hence, we show that it is not necessary to learn the background itself to sense the innovations or the foreground objects. We also explain how to adapt this representation so that our approach is robust against variations of the background such as illumination changes.

3. We develop an object detector directly on the compressive samples. Hence, no foreground reconstruction is done until a detection is made to save computation.

4. In a multi-camera setting we formulate 2-D tracking and 3-D voxel reconstruction peroblems as sparse estimation problems and use the estimated compressive foreground for its recovery.

The organization of the chapter is as follows. Section 4.2 reviews the relevant CS theory. Section 4.3 explains the details of the background subtraction for silhouette and appearance recovery. Section 4.4 presents the multi-view estimation formulation. Section 4.5 discusses the limitations of our approach. Section 4.6

demonstrates the effectiveness of our method for background subtraction with field data and presents multi-view estimation results.

4.2 The Compressive Sensing Theory

4.2.1 Sparse Representations

Suppose that we have an image $\boldsymbol{X}$ of size $N_1 \times N_2$ and we vectorize it into a column vector ($N = N_1 \times N_2$) by concatenating the individual columns of $\boldsymbol{X}$ in order. The nth element of the image vector $\boldsymbol{x}$ is referred to as $x(n)$, where $n = 1, \ldots, N$. Let us assume that the basis $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_N]$ provides a K-sparse representation of $\boldsymbol{x}$:

$$\boldsymbol{x} = \sum_{n=1}^{N} \theta(n)\boldsymbol{\psi}_n = \sum_{l=1}^{K} \theta(n_l)\boldsymbol{\psi}_{n_l}, \tag{4.1}$$

where $\theta(n)$ is the coefficient of the nth basis vector ψ_n ($\boldsymbol{\psi}_n$: $N \times 1$) and the coefficients indexed by n_l are the K-nonzero entries of the basis decomposition. Equation 4.1 can be more compactly expressed as follows

$$\boldsymbol{x} = \boldsymbol{\Psi}\boldsymbol{\theta}, \tag{4.2}$$

where $\boldsymbol{\theta}$ is an $N \times 1$ column vector with K-nonzero elements. Using $\|\cdot\|_p$ to denote the ℓ_p norm where the ℓ_0 norm simply counts the nonzero elements of $\boldsymbol{\theta}$, we call an image $\boldsymbol{X}$ as K-sparse if $\|\boldsymbol{\theta}\|_0 = K$.

Many different basis expansions can achieve a sparse representation of natural images, including wavelets, Gabor frames, and curvelets [24, 91]. However, in most cases, a natural image does not result in an exactly K-sparse representation; instead,

its transform coefficients decay exponentially fast to zero. The discussion below also applies to such images, denoted as compressible images, as they can be well-approximated using the K largest terms of $\boldsymbol{\theta}$.

4.2.2 Random/Incoherent Projections

In the CS framework, it is assumed that the K-largest $\theta(n)$ are not measured directly. Rather, $M < N$ linear projections of the image vector $\boldsymbol{x}$ onto another set of vectors $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1', \ldots, \boldsymbol{\phi}_M']'$ are measured:

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} = \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\theta}, \tag{4.3}$$

where the vector $\boldsymbol{y}$ $(M \times 1)$ constitutes the compressive samples and the matrix $\boldsymbol{\Phi}$ $(M \times N)$ is called the *measurement matrix*. Since $M < N$, recovery of the image $\boldsymbol{x}$ from the compressive samples $\boldsymbol{y}$ is underdetermined; however, as we discuss below, the additional K*-sparsity* assumption makes recovery possible.

The CS theory states that when (i) the columns of the sparsity basis $\boldsymbol{\Psi}$ cannot sparsely represent the rows of the measurement matrix $\boldsymbol{\Phi}$ and (ii) the number of measurements M is greater than $\mathcal{O}\left(K \log\left(\frac{N}{K}\right)\right)$, then it is possible to recover the set of nonzero entries of $\boldsymbol{\theta}$ from $\boldsymbol{y}$ [24, 43]. Then, the image $\boldsymbol{x}$ can be obtained by the linear transformation of $\boldsymbol{\theta}$ in 4.1. The first condition is called the incoherence of the two bases and it holds for many pairs of bases, e.g., delta spikes and the sine waves of the Fourier basis. Surprisingly, the incoherence also holds with high probability between an arbitrary basis and a randomly generated one, e.g., i.i.d. Gaussian or Bernoulli/Rademacher ± 1 vectors.

4.2.3 Signal Recovery via ℓ_1 Optimization

There exists a computationally efficient recovery method based on the following ℓ_1-optimization problem [24, 43]:

$$\widehat{\boldsymbol{\theta}} = \arg\min \|\boldsymbol{\theta}\|_1 \quad \text{s. t. } \boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\theta}. \tag{4.4}$$

This optimization problem, also known as *Basis Pursuit* [43], can be efficiently solved using polynomial time algorithms.

Other formulations are used for recovery from noisy measurements such as Lasso, Basis Pursuit with quadratic constraint [24]. In this chapter, we use the Basis Pursuit Denoising (BPDN) optimization for recovery:

$$\widehat{\boldsymbol{\theta}} = \arg\min \|\boldsymbol{\theta}\|_1 + \frac{1}{2}\beta\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\theta}\|_2^2, \tag{4.5}$$

where $0 < \beta < \infty$ [37]. When the images of interest are smooth, a strategy based on minimizing the total variation of the image works notably well [24].

4.3 CS for Background Subtraction

With background subtraction, our objective is to recover the location, shape and (sometimes) appearance of the objects given a test image over a known background. By definition, the background image contains no objects of interest. Let us denote the background, test, and difference images as $\boldsymbol{x}_b$, $\boldsymbol{x}_t$, and $\boldsymbol{x}_d$, respectively. The difference image is obtained by pixel-wise subtraction of the background image from the test image. Note that the support of $\boldsymbol{x}_d$, denoted as $\mathcal{S}_d = \{n | n = 1, \ldots, N; |\boldsymbol{x}_d(n)| \neq 0\}$, gives us the location and the silhouettes of

(a) (b) (c)

Figure 4.1: (a) Example background image. (b) Test image. (c) Difference image. Note that the vehicle appearance also shows the curb in the background, which it occludes. The images (a) and (b) are from the PETS 2001 database.

the objects of interest, but not their appearance (see Fig. 4.1).

4.3.1 Sparsity of Background Subtracted Images

Suppose $\boldsymbol{x}_b$ and $\boldsymbol{x}_t$ are typical real-world images in the sense that when wavelets are used as the sparsity basis for $\boldsymbol{x}_b$, $\boldsymbol{x}_t$, and $\boldsymbol{x}_d$, these images can be well approximated with the largest K coefficients with hard thresholding [90], where K is the corresponding sparsity proportional to the cardinality of the image support. The images $\boldsymbol{x}_b$ and $\boldsymbol{x}_t$ differ only on the support of the foreground, which has a cardinality of $P = |\mathcal{S}_d|$ pixels with $P \ll N$. Moreover, we assume that images have uniform complexity in space. We model the sparsity of the real world images as a function of their size: $K_{\text{scene}} = K_b = K_t = (\lambda_0 \log N + \lambda_1)N$, where $(\lambda_0, \lambda_1) \in \mathbb{R}^2$. We assume that the difference image is also a real-world image on a restricted support (see Fig. 4.1(c)), and similarly we approximate its sparsity as $K_d = (\lambda_0 \log P + \lambda_1)P$.

The number of compressive samples M necessary to reconstruct $\boldsymbol{x}_b$, $\boldsymbol{x}_t$, and $\boldsymbol{x}_d$ in N dimensions are then given by $M_{\text{scene}} = M_b = M_t \approx K_{\text{scene}} \log(N/K_{\text{scene}})$ and $M_d \approx K_d \log(N/K_d)$. When $M_d < M_{\text{scene}}$, a smaller number of samples is needed

to reconstruct the difference image than the background or foreground images. We empirically show in Sect. 4.6 that this condition is almost always satisfied when the sizes of the difference images are smaller than original image sizes for natural images.

4.3.2 The Background Constraint

Let us assume that we have multiple compressive measurements $\boldsymbol{y}_{bi}$ ($M \times 1$, $i = 1, \ldots, B$) of training background images $\boldsymbol{x}_{bi}$, where $\boldsymbol{x}_b$ is their mean. Each compressive measurement is a random projection of the whole image, whose distribution we approximate as an i.i.d. Gaussian distribution with a constant variance $\boldsymbol{y}_{bi} \sim \mathcal{N}(\boldsymbol{y}_b, \sigma^2 \boldsymbol{I})$, where the mean value is $\boldsymbol{y}_b = \boldsymbol{\Phi}\boldsymbol{x}_b$. When the scene changes to include an object which was not part of the background model and we take the compressive measurements, we obtain a test vector $\boldsymbol{y}_t = \boldsymbol{\Phi}\boldsymbol{x}_t$, where $\boldsymbol{x}_d = \boldsymbol{x}_t - \boldsymbol{x}_b$ is sparse in the spatial domain.

In general, the sizes of the foreground objects are relatively smaller than the size of the background image; hence, we model the distribution of the elements of the *literally* background subtracted vector as $\boldsymbol{y}_d = \boldsymbol{y}_t - \boldsymbol{y}_b \sim \mathcal{N}(\boldsymbol{\mu}_d, \sigma^2 \boldsymbol{I})$ ($M \times 1$). Note that the appearance of the objects constructed from the samples $\boldsymbol{y}_d$ would correspond to the literal subtraction of the test frame and the background; however, their silhouette is preserved (Fig. 4.1(c)).

The number of samples M in $\boldsymbol{y}_b$ is greater than M_d as discussed in Sect. 4.3.1, but is not necessarily greater than or equal M_b or M_t; hence, it may not be suffi-

cient to reconstruct the background. However, the background image $\boldsymbol{x}_b$ still needs to satisfy the constraint $\boldsymbol{y}_b = \boldsymbol{\Phi}\boldsymbol{x}_b$. To be robust against small variations in the background and noise, we consider the distribution of the ℓ_2 distances of the background frames around their mean $\boldsymbol{y}_b$:

$$\|\boldsymbol{y}_{bi} - \boldsymbol{y}_b\|_2^2 = \sigma^2 \sum_{n=1}^{M} \left(\frac{y_{bi}(n) - y_b(n)}{\sigma} \right)^2 . \tag{4.6}$$

When M is greater than 30, this sum can be well approximated by a Gaussian distribution due to the central limit theorem. Then, it is straightforward to show that we have $\|\boldsymbol{y}_{bi} - \boldsymbol{y}_b\|_2^2 \sim \mathcal{N}\left(M\sigma^2, 2M\sigma^4\right)$. When we have a test frame with a foreground object, the same distribution becomes $\|\boldsymbol{y}_t - \boldsymbol{y}_b\|_2^2 \sim \mathcal{N}\left(M\sigma^2 + \|\boldsymbol{\mu}_d\|_2^2, 2M\sigma^4 + 4\sigma^2\|\boldsymbol{\mu}_d\|_2^2\right)$.

Since σ^2 scales the whole distribution and $1/M \ll 1$, the logarithm of the ℓ_2 distances in (4.6) can be approximated quite accurately with a Gaussian distribution. That is, since $u \ll 1$ implies $1 + u \approx \mathrm{e}^u$, we have $\mathcal{N}\left(M\sigma^2, 2M\sigma^4\right) = M\sigma^2\mathcal{N}\left(1, \frac{2}{M}\right) = M\sigma^2\left(1 + \sqrt{\frac{2}{M}}\mathcal{N}\left(0,1\right)\right) \approx M\sigma^2 \exp\left\{\sqrt{\frac{2}{M}}\mathcal{N}\left(0,1\right)\right\}$. This derivation can also motivated by the fact that the square-root of the Chi-squared distribution can be well approximated by a Gaussian [33].

Hence, (4.6) can be used to approximate

$$\log \|\boldsymbol{y}_{bi} - \boldsymbol{y}_b\|_2^2 \sim \mathcal{N}\left(\mu_{bg}, \sigma_{bg}^2\right), \tag{4.7}$$

where the variance term does not depend on the additive noise in the pixel measurements. Equation (4.7) allows some variability around the constraint $\boldsymbol{y}_b = \boldsymbol{\Phi}\boldsymbol{x}_b$ that the background image needs to satisfy in order to cope with the small variations of the background and the measurement noise. However, the samples $\boldsymbol{y}_d = \boldsymbol{y}_t - \boldsymbol{y}_b$

can be used to recover the foreground objects. We learn the log-Normal parameters in (4.7) from the data using the maximum likelihood techniques.

4.3.3 Object Detector based on CS

Before we attempt any reconstruction, it is a good idea to determine if the test image has any differences from the background. Using the results from Sect. 4.3.2, the ℓ_2 distance of $\boldsymbol{y}_t$ from $\boldsymbol{y}_b$ can be subsequently approximated by

$$\log \|\boldsymbol{y}_t - \boldsymbol{y}_b\|_2^2 \sim \mathcal{N}\left(\mu_t, \sigma_t^2\right). \tag{4.8}$$

When the object is small, σ_t^2 should be on the same order size of σ_{bg}^2, while μ_t is different from μ_{bg} in (4.7). Then, to test the hypothesis of whether there is a new object, the optimal detector would be a simple threshold test since we would be comparing two Gaussian distributions with similar variances. When σ_t^2 is significantly different from σ_{bg}^2, the optimal test can be a two sided threshold test [149]. For our case, we simply use a constant times the standard deviation of the background as a threshold and declare that there is a new object if $|\log \|\boldsymbol{y}_t - \boldsymbol{y}_b\|_2^2 - \mu_{bg}| \geq c\sigma_{bg}$.

4.3.4 Foreground Reconstruction

For foreground reconstruction, we use BPDN with a fixed point continuation method [62] and total variation (TV) optimization with an interior point method [24] on the background subtracted compressive measurements. During reconstruction, we lose the actual appearance of the objects as the acquired measurements also contain information about the background. Although it is known that the subtracted

image is a sum of two components that exclusively appear in $\boldsymbol{x}_b$ and $\boldsymbol{x}_t$, it is difficult, if not impossible, to unmix them without taking enough measurements to recover $\boldsymbol{x}_b$ or $\boldsymbol{x}_t$. Hence, if the appearances of the objects are needed, a straightforward way to obtain them would be to either reconstruct the test image by taking enough compressive samples and then use the binary foreground image as a mask, or reconstruct and mask the background image and then add the result to the foreground estimate.

4.3.5 Adaptation of the Background Constraint

We define two types of changes in a background: drifts and shifts. A background drift consists of gradual changes that occur in the background such as illumination changes in the scene and may result in immediate unwanted foreground estimates. A background shift is a major and sudden change in the definition of the background, such as a new vehicle parked within the scene. Adapting to background shifts at the sensing level is quite difficult because high level logical operations are required, such as detecting the new object and deciding that it is uninteresting. However, adapting to background drifts is essential for a robust background subtraction system as it has immediate impacts on the foreground recovery.

The background constraint $\boldsymbol{y}_b$ needs to be updated continuously if the background subtraction system is to be robust against the background drifts. Otherwise, the drifts may accumulate and trigger unwanted detections. In the compressive sensing framework, this can be done as follows. Once we obtain an estimate of

the difference image $\widehat{\boldsymbol{x}}_d$ with one of the reconstruction algorithms discussed in the previous section, we determine the compressive samples that should be generated by it: $\widehat{\boldsymbol{y}}_d = \boldsymbol{\Phi}\widehat{\boldsymbol{x}}_d$. Since we already have $\boldsymbol{y}_d = \boldsymbol{y}_t - \boldsymbol{y}_b$, we can substitute the denoised difference estimate to obtain the background estimate of the current frame: $\widehat{\boldsymbol{y}}_b = \boldsymbol{y}_t - \widehat{\boldsymbol{y}}_d$. Then, a running average can be used to update the background with a learning rate of $\alpha \in (0, 1)$ as follows:

$$\boldsymbol{y}_b^{\{j+1\}} = \alpha \left(\boldsymbol{y}_t^{\{j\}} - \widehat{\boldsymbol{y}}_d^{\{j\}} \right) + (1 - \alpha) \boldsymbol{y}_b^{\{j\}}, \tag{4.9}$$

where j is the time index.

Unfortunately, this update rule does not suffice for compensating background shifts, such as global illumination changes and new stationary targets. Consider a pixel whose intensity value changes because of a background shift. This pixel will then be identified as an outlier in the background model. The corresponding pixel in the background model will not be updated in (4.9). Hence, for all future frames, the pixel will continue to be classified as part of the foreground. This problem can be handled by allowing for a second moving average of the frames, which updates all pixels within the image as in [70].

Hence, we use the following updates:

$$\begin{aligned} \boldsymbol{y}_{\text{ma}}^{\{j+1\}} &= \gamma \boldsymbol{y}_t^{\{j\}} + (1 - \gamma) \boldsymbol{y}_{\text{ma}}^{\{j\}}, \\ \boldsymbol{y}_b^{\{j+1\}} &= \alpha \left(\boldsymbol{y}_t^{\{j\}} - \widehat{\boldsymbol{y}}_e^{\{j\}} \right) + (1 - \alpha) \boldsymbol{y}_b^{\{j\}}, \end{aligned} \tag{4.10}$$

where $\boldsymbol{y}_{\text{ma}}$ is the simple moving average, $\gamma \in (0, 1)$ is the moving average learning rate, and $\widehat{\boldsymbol{y}}_e = \boldsymbol{\Phi}\widehat{\boldsymbol{x}}_{\text{ma}}$. Consider a global illumination change. The moving average update integrates the pixel's illumination change over time, whose speed depends on

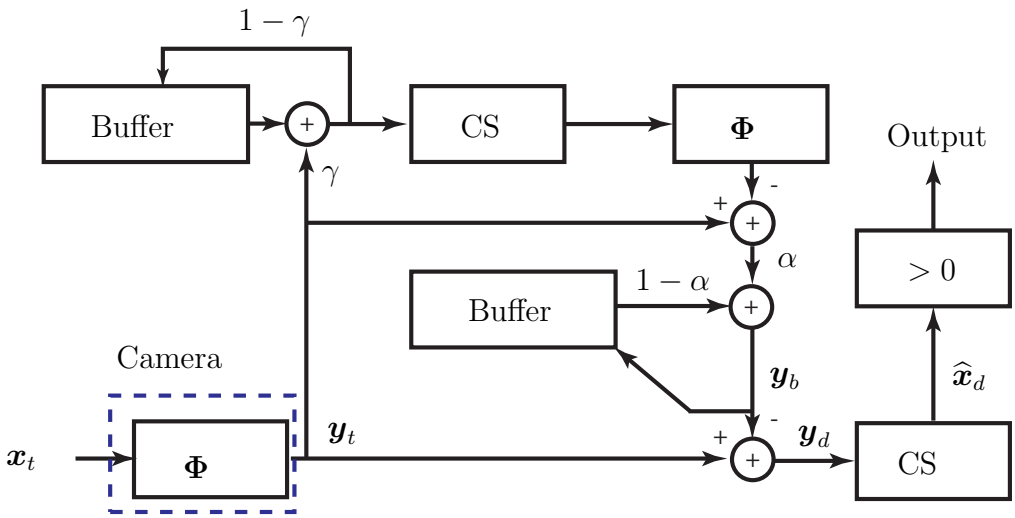

Figure 4.2: Block diagram of the proposed method.

γ. In subsequent frames, the value of the moving average will approach the intensity value observed at the pixel. This implies that when used as a detection image, the moving average will stop detecting the pixel as foreground. Once this happens, the pixel will be updated in the background update, making the background model adaptive to global changes in illumination. A disadvantage of this approach is that if the targets stay stationary for extended periods of time, they become part of the background. However, if they move again, they can be detected. Figure 4.2 illustrates the outline of the proposed background subtraction method.

4.4 Multi-view Estimation

In computer vision, silhouette images are used for various applications like tracking, activity recognition, building 3-D models using voxels etc. Silhouette images can be considered as sparse matrices where few pixels are in the foreground and most in the background. The sparsity of the silhouette images corresponds to the

sparsity of object parameters. Silhouette images are obtained by indicating which pixels of the difference image are non-zero. We utilize the sparsity of corresponding difference images, using their compressed samples, to directly recover the object parameters in a multi-view setting. We first formulate the multi-view tracking problem and then consider the formulation of 3-D voxel reconstruction task. We assume that multiple cameras are observing a scene.

Suppose we have the observation region $\boldsymbol{O}$ (which can be either 2-D or 3-D space), being observed by synchronized cameras $c = 1, ..., C$. We assume that most of the region is visible to all the cameras. At any frame $f \in \{1, 2, ..., F\}$ we have background subtracted difference images (i.e. foreground) $\boldsymbol{I}_f^c$ (of size $N_{row} \times N_{col}$) at each of the cameras. The foreground in the image is the moving object which we are interested in tracking and on which we would like to further focus and perform our analysis. The foreground is sparse in the image plane. This implies that in the corresponding observation region $\boldsymbol{O}$, the objects or people corresponding to the image foreground are sparse i.e., the area (or volume) occupied by the moving objects or people is very small compared to the area (or volume) of the observation region. We relate the foreground image and the corresponding objects in the observation region by a linear transformation. For simplicity, we first consider a 2-D observation region $\boldsymbol{O}$ where camera c is provided with homography $\boldsymbol{H}^c$ between the world plane and the image plane.

Assume that for some frame f the cameras are observing the 2-D observation space $\boldsymbol{O}$ as shown in Fig. 4.3. The region is divided into non-overlapping, tightly packed subregions $n = 1, ..., N$ where (x_n, y_n) are the coordinates of the representa-

tive point (like points on the ground plane in Fig. 4.3) of the subregion n, known at the cameras. In tracking application we would like to localize the objects to one of the regions. Assume that camera c observes the foreground image $\boldsymbol{I}_f^c$ (foreground has non-zero value and background is zero) with image coordinates (u, v). We define vectors $\boldsymbol{s}$ and $\boldsymbol{x}_d^c$ associated with the object location on the 2-D plane and the silhouette image respectively. $\boldsymbol{s}$ is a $N \times 1$ vector with $\boldsymbol{s}(n) \neq 0$ if object is present in the subregion n and 0 otherwise. In Fig. 4.3, $\boldsymbol{s}$ is the indicator of the objects at points on the ground plane and $\boldsymbol{x}_d^c$ is the foreground at corresponding points on the silhouette image at camera c. Typically in tracking scenarios $\boldsymbol{s}$ is sparse since the objects of interest occupy a small area in the observation region. For every frame f we would like to know the position of the objects, in other words our desired variable is $\boldsymbol{s}$. But, what the cameras instead observe is the background subtracted image $\boldsymbol{I}_f^c$ and for the multi-view estimation problem we assume that the camera (normal or compressive) can be setup to sense only the $N \times 1$ vector $\boldsymbol{x}_d^c$ defined as,

$$\boldsymbol{x}_d^c(i) = \boldsymbol{I}_f^c(u_i, v_i), \tag{4.11}$$

where the image coordinates (u_n, v_n) of camera c are related to the coordinate (x_n, y_n) of the representative point n by

$$\begin{bmatrix} u_n \\ v_n \\ 1 \end{bmatrix} \backsim \boldsymbol{H}^c \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix}, \tag{4.12}$$

(it should be noted that since (u_n, v_n) take integer values we round the right hand

side). The equation relating $\boldsymbol{s}$ and $\boldsymbol{x}_d^c$ is then given by

$$\boldsymbol{x}_d^c = \boldsymbol{s} + \boldsymbol{e}_d^c, \tag{4.13}$$

where $\boldsymbol{e}_d^c$ is the view dependent error when observing the desired vector $\boldsymbol{s}$. In our model, this error also encompasses the errors in silhouettes, rounding off errors and the errors when not all subregions in $\boldsymbol{O}$ are visible to camera c. Given $\boldsymbol{O}$, each camera can sense $\boldsymbol{x}_d^c$ as described above. A simple projection of the foreground at a single camera on the ground plane gives an estimate of the object location but this is not accurate since the parts of the object in parallax do not register under the homography projections. To accurately estimate the position of the objects on the ground plane from $\boldsymbol{x}_d^c$, information from multiple cameras is used. For this, cameras need to transmit the information to a centralized location where the computation can be performed. Noting that $\boldsymbol{s}$ is sparse, the compressive cameras can significantly decrease the amount of data sensed and hence the data transmitted to the central processing center by projecting the signals into lower dimensions and recovering it using the principle of CS. We assume the vector $\boldsymbol{s}$ to be K-sparse. This means that we can randomly project the vector $\boldsymbol{x}_d^c$ to lower dimensional $\boldsymbol{y}^c$ $(= \boldsymbol{\Phi}^c \boldsymbol{x}_d^c)$ using the $M \times N$ projection matrix $\boldsymbol{\Phi}^c$ with entries from Gaussian distribution $\mathcal{N}(0, 1/N)$ or random Bernoulli distribution. The resulting equation is

$$\boldsymbol{y}^c = \boldsymbol{\Phi}^c \boldsymbol{s} + \boldsymbol{e}^c. \tag{4.14}$$

where $\boldsymbol{e}^c = \boldsymbol{\Phi}^c \boldsymbol{e}_d^c$. Note that $\boldsymbol{y}^c$ is the compressive foreground measurement and it can be obtained from the background subtraction algorithm described in the

previous section. At the central location $\boldsymbol{y}^c$ are stacked to form vector $\boldsymbol{y}$

$$\begin{bmatrix} \boldsymbol{y}^1 \\ \boldsymbol{y}^2 \\ \vdots \\ \boldsymbol{y}^C \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}^1 \\ \boldsymbol{\Phi}^2 \\ \vdots \\ \boldsymbol{\Phi}^C \end{bmatrix} \boldsymbol{s} + \begin{bmatrix} \boldsymbol{e}^1 \\ \boldsymbol{e}^2 \\ \vdots \\ \boldsymbol{e}^C \end{bmatrix} \tag{4.15}$$

resulting in

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{s} + \boldsymbol{e} \tag{4.16}$$

Assuming $\boldsymbol{e}$ to be additive white Gaussian(AWG) noise, the vector $\boldsymbol{s}$ is recovered by solving (4.5). The recovery of $\boldsymbol{s}$ is a function of N allowing the algorithm to scale in number of cameras. For simplicity, we consider rectangular subregions $n = 1, ..., N$, with the representative points forming a $N_1 \times N_2$ rectangular grid.The problem described above easily extends to 3-D voxel reconstruction. We assume a 3-D $\boldsymbol{O}$ being observed by C cameras. At frame f, the cameras observe silhouette images $\boldsymbol{I}_f^c$. We assume that at camera c, the projection matrix $\boldsymbol{P}^c$ is known. Unlike multi-view ground plane tracking, in 3-D voxel reconstruction we need to recover all the three coordinates of the object to reconstruct the 3-D shape. We divide the 3-D observation region into N sufficiently dense subregions which are non-overlapping and tightly packed. Here the representative point of the subregion n has coordinates (x_n, y_n, z_n). Again, for simplicity we assume that the subregions are cuboidal volumes called voxels and the representative points form a $N_1 \times N_2 \times N_3$ grid. The sub-regions are denser compared to tracking and the object occupies a lot more subregions than it did in tracking scenario. Similarly, we define $N \times 1$ vectors $\boldsymbol{s}$ and $\boldsymbol{x}_d^c$. $\boldsymbol{s}(n) \neq 0$ if object occupies subregion n and 0 otherwise. Obviously, if $\boldsymbol{s}(i) \neq 0$

we would have $\boldsymbol{x}_d^c(i) = \boldsymbol{I}_f^c(u_i, v_i) \neq 0$ where

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \boldsymbol{P}^c \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}, \tag{4.17}$$

and (x_i, y_i, z_i) are the coordinates of the grid point in voxel i. All the voxels whose projection onto the image plane of camera c intersects with the silhouette of image $\boldsymbol{I}_f^c$ are assigned to be occupied — i.e., $\boldsymbol{x}_d^c(i) = 1$. Thus for any camera the number of voxels assigned as occupied is greater than the number of truly occupied voxels. Hence, for finding the true voxel occupation $\boldsymbol{s}$ we use silhouettes from multiple cameras. In the region $\boldsymbol{O}$ the volume occupied by the object is assumed to be sparse implying a sparse $\boldsymbol{s}$. To recover $\boldsymbol{s}$ from $\boldsymbol{x}_d^c$, $c = 1, \ldots, C$ we follow exactly the recovery procedure adopted in multi-view tracking.

4.5 Limitations

In this section, we discuss some of the limitations of our specific compressive sensing approach to the background subtraction. Some of these limitations can be caused by the hardware architecture, whereas others are due to our image models. Note that our formulation is general enough that we do not require an SPC for operation. If a centralized vision system is used with no background subtraction at the camera, then our methods can be used at conventional cameras for processing in the compressive domain to reduce communication bandwidth and be robust against

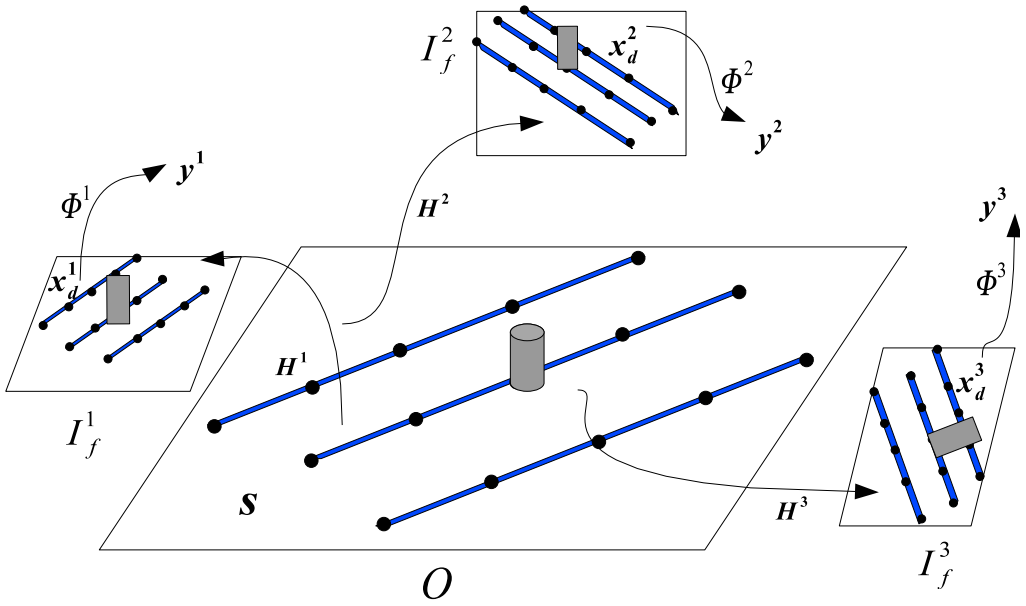

Figure 4.3: Ground plane tracking scenario. The observation region $\boldsymbol{O}$ is observed by 3 cameras. The points on $\boldsymbol{O}$ are the representative points of the subregions known at cameras. The homography $\boldsymbol{H}^c$ is provided at camera c. $\boldsymbol{I}_f^c$ are the silhouette images at camera c at frame f. The silhouette values at points on image $\boldsymbol{I}_f^c$ form $\boldsymbol{x}_d^c$. The sparsity of the silhouette image corresponds to the sparsity of the object(cylinder)

packet drops.

The SPC architecture uses a DMD to generate a random sampling pattern and sends the resulting inner product of the incident light field from the scene with the random pattern to the optical sensor to create a compressive measurement. By changing the random pattern in time, a set of M consecutive measurements can be made about the scene using the same optical sensor, which form the measurement vector $\boldsymbol{y}$. The current DMD arrays can change their geometric configuration approximately 10 to 40K times per second. For example, with a rate of 30K times per second, we can construct at most a 300×300 resolution background subtracted image with 1% compression ratios at 30fps. Although the resolution may not be sufficient for some applications, it will improve as the capabilities of the DMD arrays

increase.

In our background modeling, we assume that the background and foreground images exhibit sparsity. We argued that the background subtracted image has a lower sparsity and hence can be reconstructed with fewer number of samples that is necessary to reconstruct the background or the foreground images. When the images of interest do not show sparsity (e.g., they are white noise), our approach can still be applied. That is, the difference image $\boldsymbol{x}_d$ is always sparse regardless of the sparsities of $\boldsymbol{x}_b$ and $\boldsymbol{x}_t$ if its support cardinality P is much smaller than N.

In our formulations, we have used black and white images as opposed to color images. Although our ideas easily extend to multiple color planes, the required computation also proportionally increase. Finally, our Gaussian model for background images cannot cope with multimodal backgrounds. These backgrounds require more sophisticated processing, which is the focus of future work.

In the multi-view estimation formulation we made a simplifying assumption to map the ground plane to the image plane. We assume that the location vector $\boldsymbol{s}$ has values which are the same as the estimated difference image $\boldsymbol{x}_d^c$ value. Our assumption holds only in cases where the object has roughly the same appearance from all the directions. Further, for simplicity we assume that the noise $\boldsymbol{e}$ in the estimation problem is Gaussian but this often not the case.

4.6 Experiments

4.6.1 Background Subtraction with an SPC

We performed background subtraction experiments with an SPC; in our test, the background $\boldsymbol{x}_b$ consists of the standard test *Mandrill* image, with the foreground $\boldsymbol{x}_t$ consisting of a white rectangular patch as shown in Fig. 4.4. Both the background and the foreground were acquired using pseudorandom compressive measurements ($\boldsymbol{y}_b$ and $\boldsymbol{y}_t$, respectively) generated by a Mersenne Twister algorithm with a 64×64 pixel resolution [99]. We obtain measurements for the subtraction image as $\boldsymbol{y}_d = \boldsymbol{y}_t - \boldsymbol{y}_b$. We reconstructed both the background, test, and difference images, using TV minimization. The reconstruction is performed using several measurement rates ranging from 0.5% to 50%. In each case, we compare the subtraction image reconstruction with the difference between the reconstructed test and background images. The resulting images are shown in Fig. 4.4, and show that for low rates the background and test images are not recovered accurately, and therefore the subtraction performs poorly; however, the sparser foreground innovation is still recovered correctly from the difference of the measurements, with rates as low as 1% being able to recover the foreground at this low resolution.

4.6.2 The Sparsity Assumption

In our formulation, we assumed that the sparsity of natural images has the following form: $K = (\lambda_0 \log N + \lambda_1)N$. To test this assumption, we used the Berkeley Segmentation Data Set (BSDS) as a natural image database [97] and obtained

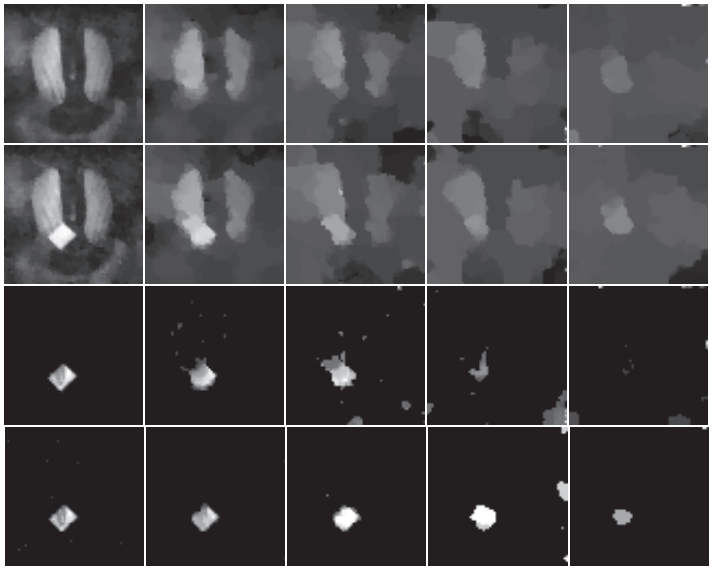

Figure 4.4: Background subtraction experimental results using an SPC. First row: reconstruction of background image from compressive measurements. Second row: reconstruction of the test image from compressive measurements. Third row: conventional background subtraction using the above images. Fourth row: reconstruction of difference image directly from compressive measurements. The columns correspond to measurement rates M/N of 50%, 5%, 2%, 1% and 0.5%, from left to right. Background subtraction from compressive measurements is feasible at lower measurement rates than standard background subtraction.

wavelet approximations of various block sizes varying from 2×2 to 256×256 pixels. To approximate the sparsity K of any given tile size, we determined the minimum number of wavelet coefficients that results in a compression with -40dB distortion with respect to the image itself. Figure 4.5(a) shows that our sparsity assumption is justified for natural images. Figure 4.5(b) illustrates that the necessary number of compressive samples is monotonic with the tile size. Therefore, if the innovations in the image are smaller than the image, it takes fewer compressive samples to recover them. In fact, the total number of samples necessary to reconstruct is rather close to linear: $M \approx \kappa N^{1-\delta}$ where $\delta \ll 1$. In general, the λ parameters are scene specific (Fig. 4.5(c)). Hence, the exact number of compressive measurements needed may

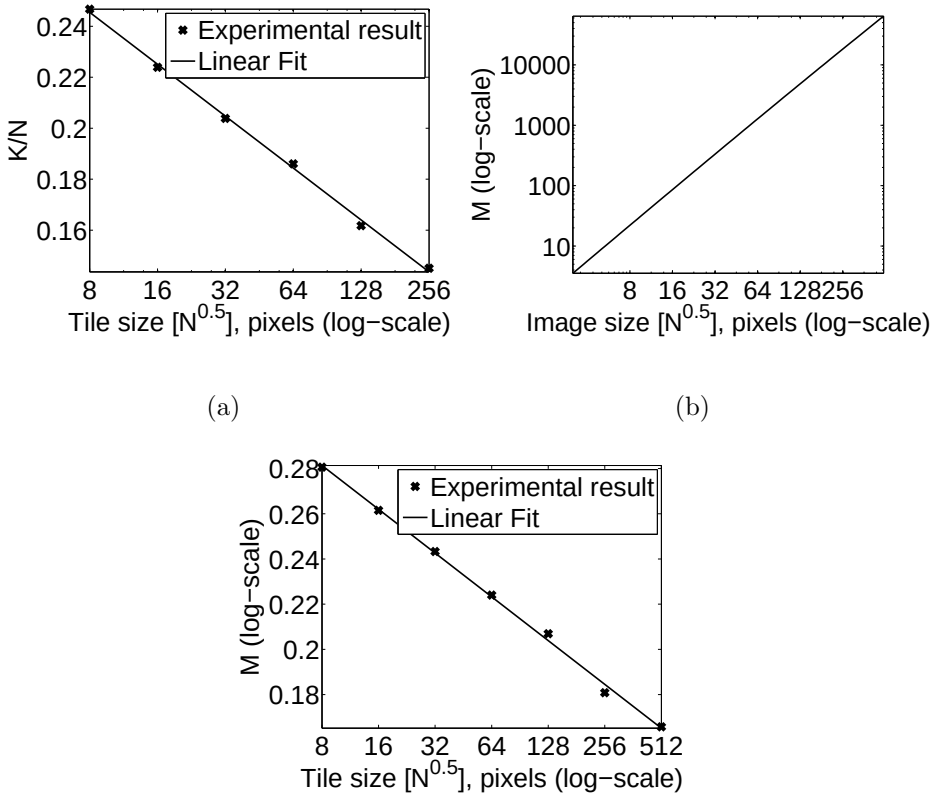

Figure 4.5: (a) Average sparsity over N as a function of the tile size for the images in BSDS. (b) Number of compressive measurements needed to reconstruct an image of different sizes from BSDS. (c) Average sparsity over N as a function of the tile size for the images in PETS 2001 data set.

vary.

4.6.3 Adaptation to Illumination Changes

To compare the performance of the background constraint adaptations (4.9) (drift adaptive) and (4.10) (shift adaptive), we test them on a sequence where there is a global illumination change due to sunlight. To emphasize the differences, we use the delta basis (0/1 in spatial domain) as the sparsifying basis $\mathbf{\Psi}$. This basis creates much noisier background subtraction images than wavelets, but it is quite

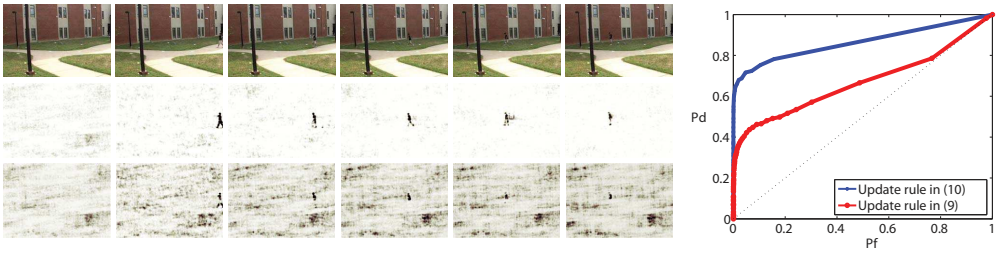

Figure 4.6: Background subtraction results on a sequence with changing illumination using (4.9) and (4.10) for background constraint updates. Outputs are shown with identical parameters used for both models. Note that for the same detection output, the update rule (4.10) produces much less false alarm. However, (4.10) has twice the computational cost as (4.9).

illustrative for the purposes of this comparison.

Figure 4.6 shows the results of the comparison. The images on top are the original images. The middle row corresponds to the update in (4.10) whereas the bottom row images correspond to the update in (4.9). The update in (4.10) allows the background constraint to keep track of the changing illumination. Hence, the resulting images are cleaner and continue to improve if the illumination does not change. This results in much lower false alarm rates for the same detection probability (see Fig. 4.6(*Right*)). For plotting the receiver operating characteristics (ROC) curves, we use the full images, run the background subtraction algorithm proposed in [70], and obtain baseline background subtracted images. We then compare the pixels on the resulting target from different updates to calculate the detection rate. We also compare the spurious detections in the rest of the images to generate the ROC curve.

4.6.4 Multi-view Ground Plane Tracking

We present the results of an experiment performed using video sequences collected by four cameras located in an outdoor area. The background-subtracted images at the cameras are of size 240×320 ($N_{row} = 240, N_{col} = 320$). First, we detect the objects in the scene and then track them over 400 frames. During detection the observation region $\boldsymbol{O}$ is a rectangular region 60ft$\times$55ft most of which is observed by all the 4 cameras. We place a sufficiently dense 101×101 ($N_1 = 101, N_2 = 101$) uniformly spaced grid on this region, implying $N = N_1 N_2 = 101^2$ subregions over which we detect and localize the objects. Following the procedure described in Section 4.4 we recover the vector $\boldsymbol{x}$ which indicates which sub-regions have the object. We detect 2 objects which we track over the next 400 frames. Since each object occupies more than a point, instead of recovering an exact 2-sparse vector $\boldsymbol{s}$ we get a more dense vector. The location is estimated by averaging over these 2 dense blobs. Once we detect the objects, we track them using a similar procedure but for tracking we confine our region of search to a rectangular region of size 20ft $\times$ 20ft centered at the detected object locations. For tracking, the observation region $\boldsymbol{O}$ at frame f is centered around the object location estimated at frame $f - 1$. On this we place a grid of size 26×26 ($N_1 = 26, N_2 = 26$) where unlike detection the grid points are distributed according to a Gaussian distribution centered at the object location and with variance 3.5ft in both directions. A Gaussian spaced grid allows us to account for the expected small movements as well as the large ones. It also decreases the complexity by decreasing N. The observed vector is randomly projected using a

matrix with IID Gaussian entries. The tracking results are shown in Fig. 4.7

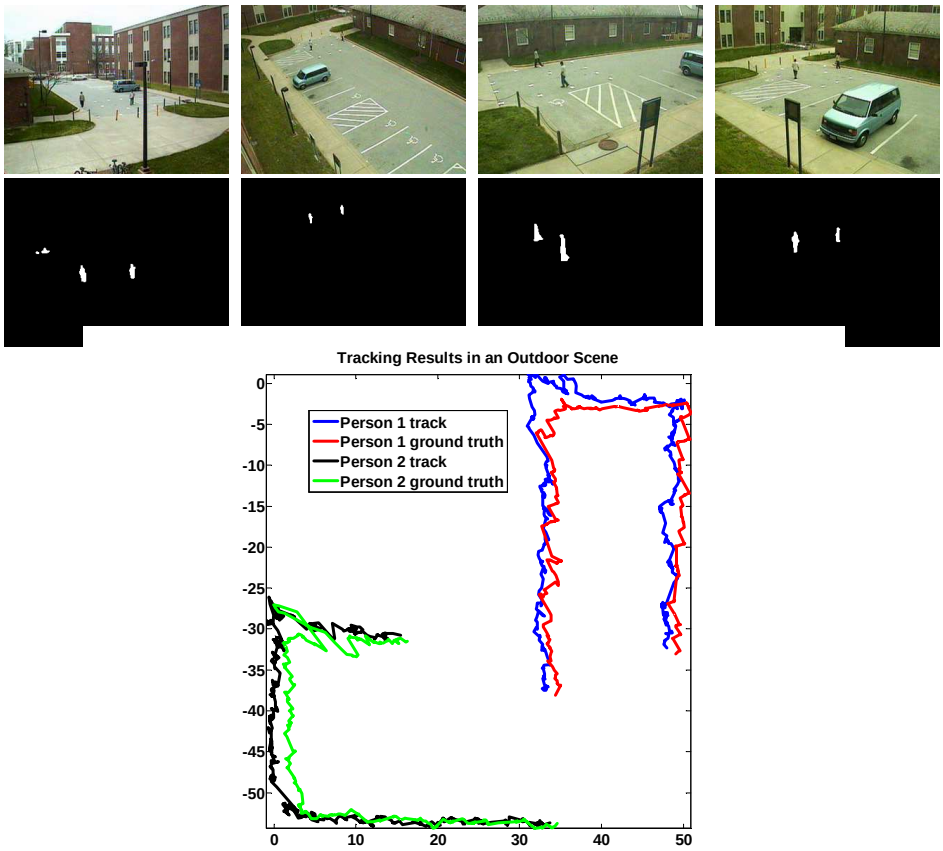

Figure 4.7: Outdoor scene of size 60ft $\times$ 55ft observed by $C = 4$ cameras. Tracking results on a video sequence of 400 frames. The first two rows show sample images and the estimated background subtracted silhouettes respectively. These background subtracted difference images are used to track objects on the ground plane. The bottom image shows the tracked points (blue) as well as the ground truth (black).

We performed the 3-D reconstruction experiment in an indoor setting for one frame where the object is being observed by $C = 8$ cameras placed around it. The images are of size 484×648. The observation region is a 0.8m $\times$ 0.82m $\times$ 1.62m space. We place a sufficiently dense uniformly space grid of size $81 \times 83 \times 163$ $((N_1 = 81, N_2 = 83, N_3 = 163))$ in this region. As in the tracking scenario we randomly project the observed vector using matrix with IID Gaussian entries. To decrease the complexity of the recovery algorithm we divide the grid into smaller

chunks of size N_1 along one of the rows. We recover $\boldsymbol{s}$ which has non-zeros values corresponding to the region occupied by the object and lower values in the empty regions. We can see that our method of reconstructing 3D-voxels is robust to errors in the silhouette images. Figure 4.8(*Top*) shows the ground the difference image reconstructed using CS, which incorporates elements from the background, such as the camera setup behind the subject, affecting the final reconstruction. Hence, the difference images do not always result in the desired silhouettes. Figure 4.8(*Bottom*) shows the voxel reconstruction with four cameras with 40% compression, which is visually satisfactory despite the artifacts in the difference images.

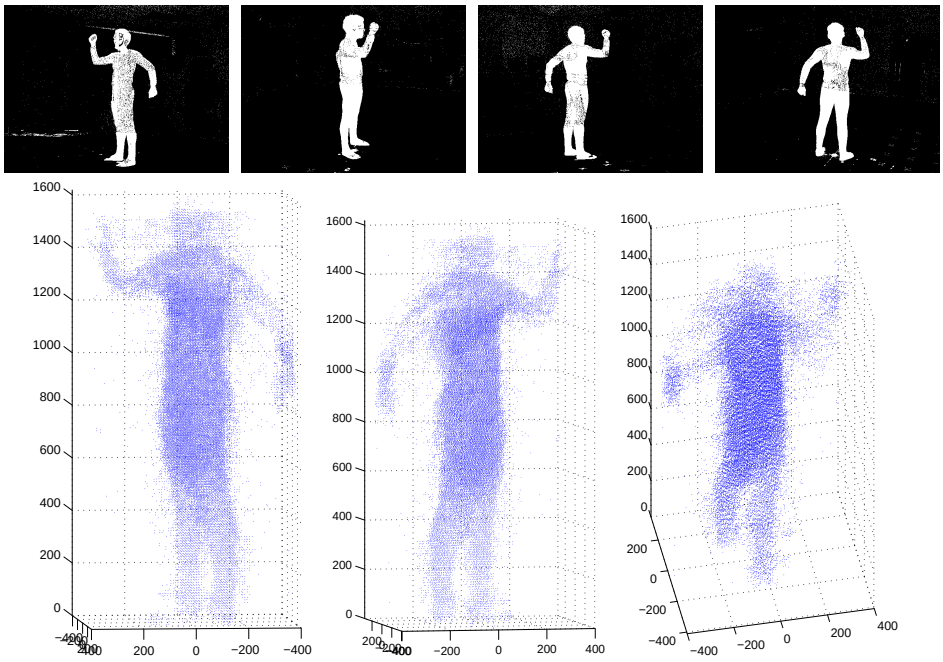

Figure 4.8: Indoor scene of size 0.8m $\times$ 0.82m $\times$ 1.62m overlayed with grid of size $81 \times 83 \times 163$ observed by $C = 8$ cameras. (Top) Estimated background subtracted silhouette images . (Bottom) Three views of the reconstructed object.

4.7 Summary

We demonstrated that the compressive sensing framework can be used to directly reconstruct sparse innovations on a background scene with a significantly fewer data samples than the conventional methods. As opposed to acquiring the minimum amount of measurements to recover a background and the test image, we exploit the sparsity of the foreground to perform background subtraction by using even fewer measurements (M_d measurements as opposed to M_b). We illustrated that due to the linear nature of the measurements, it is still possible to adapt to the changes in the background directly in the compressive domain. In addition, it is possible to formulate an object detector. We formulated the multi-view tracking and 3D reconstruction problems using sparse estimation framework. We estimate the location and occupancy of the object by relating these quantities with background subtracted difference images.

Chapter 5

Joint Compressive Video Sensing and Background Subtraction

5.1 Introduction

Cameras are often used to sense dynamic scenes for various tasks such as surveillance, tracking [164], activity recognition [145] and 3D reconstruction [103]. In these applications we are interested in separating the interesting moving objects from the static parts of the video. For example, surveillance videos have a person or a vehicle moving in the scene and the rest of it is mostly static. Similarly, other videos can also be decomposed into two processes, a slow background and the more interesting foreground. For many taks including visual inference, we focus on the foreground and hence wish to separate the interesting foreground from the background. Typically, foreground refers to the region of the image where a static or moving object is present and background refers to the static part with noisy variations due to illumination change, motion of foliage, rain etc.

Current video sensing architecture first senses the video frames and then performs preprocessing tasks such as background subtraction (BS) [140], [48] for sub-

sequent inference. Typical background subtraction involves two steps, estimating the background and separating the foreground. Often, Once the foreground is separated, the background pixels are often discarded or incorporated into the background model. This means that we first sense the frames at full resolution and then throw away majority of pixels from the background. While sensing is inexpensive in the visual range, it is significantly costly in the hyper-spectral regime. Even in visual range where the CMOS sensors are inexpensive, when the available processing power and communication bandwidth are limited, it is undesirable for a camera to first perform computationally costly background modeling and subtraction and then transmit the foreground. Since the interesting foreground often occupies only a small portion of the scene and majority of the background is slowly varying or static, it raises the following question: Can videos be sensed robustly using fewer measurements by exploiting the redundancy in both the slow and fast processes? In this chapter we address this question.

Compressive sensing (CS) [43],[24] has emerged as a potential solution to the problem of parsimonious sensing in the presence of signal redundancy. When sampling cost is high due to expensive sensors, such as in hyperspectral imaging, compressive sensing of video frames provides a solution. CS allows a signal which is sparse in a transform basis to be sensed with fewer samples and provides a recovery scheme to exactly reconstruct the signal from these samples. An advantage of CS is that the amount of sensed data is proportional to the dimension of the innovations rather than the ambient signal dimension. This fact has been used to build prototype compressive imagers such as the 'single pixel camera' (SPC) [45]. Imaging

architectures like SPC help us sense with far fewer pixels than what would otherwise be needed. In practice compressive sensing of images is effective only when the images are sparse. Since natural images are not strictly sparse but compressible, CS imaging leads to poor image quality. In domains where signals tend to be sparse, CS does provide the level of detail comparable to traditional sensing as illustrated by its applications in magnetic resonance imaging (MRI) [87] and synthetic aperture radar (SAR) imaging [114].

Compared to images, videos are redundant not only in each frame but more so in temporal dimension [104]. This has lead to CS research in exploiting redundancy in successive frames which have similar appearance due to objects in motion. Techniques have been proposed to exploit inter-frame redundancy. Methods which compressively sense videos can be found in [125], [112], [95], [72], [41], [139], [165]. CS acquisition methods for a restricted class of videos are presented in [71], [152], [151] for MRI, [154] for repetitive visual signals found in automation and [130] for videos satisfying a linear dynamical model. CS methods aimed towards a narrower class of signals with stronger signal models have tended to provide more impressive results. This has lead to interest in building compressive video sensing schemes for specific applications where in addition to the sparsity within a frame, inter-frame redundancy is also exploited. In this chapter, we present our work on a restricted but important class of videos, namely the fixed camera surveillance videos.

We present a compressive sensing and recovery scheme for videos which are composed of fast and slowly varying processes. Our recovery scheme not only recovers the videos accurately but can separate the slowly varying background from the

foreground. We show that by using a different measurement matrix during each new frame we can not only capture the sparse foreground information but also capture the background accurately without any sparsity assumptions. We formulate the recovery and background modeling problem in the framework of distributed compressive sensing (DCS) [11]. We treat each frame in a temporal window as acquired from a different sensor in DCS context and separate it into a component common to all the frames and a unique foreground. The common component and unique sparsity model for frames from a surveillance video allow us to achieve the state-of-the-art results by avoiding sparsity assumption on the background frame altogether. We make sparsity assumption only on the foreground which allows us to reconstruct it accurately.

The contributions of our work are as follows.

1. A new compressive sensing and recovery scheme to capture and recover surveillance type videos.

2. Formulation of background modeling in compressive frames as DCS.

3. A simple reconstruction algorithm to robustly recover the frames and separate the foreground from the background.

Related work:

We discuss here the previous efforts related to our approach and contrast them with our method.

CS Scheme: Since we propose a new CS sensing scheme where the measurement matrix in every frame is changed, we discuss different CS acquisition archi-

tectures which enable such acquisition. MRI [87] is a natural compressive imaging system where only a few Fourier coefficients are measured and from which the entire image slice is constructed by assuming bounded total variation. Each frame is acquired by capturing Fourier coefficients corresponding to the directions from which the scan is performed. This imaging system can be used for our problem by changing the set of scan directions in every frame leading to different measurement matrices for consecutive frames. Previous reconstruction methods for dynamic MRI include motion compensation [71]. Methods have been also proposed to reconstruct the videos by assuming small changes in Fourier coefficients of consecutive frames [151]. Our method sensing and reconstruction can be be applied to MRI when the video has a sparse moving foreground with a slowly varying background.

The first prototype demonstrating compressive imaging is the SPC [45]. The authors proposed a digital micro-mirror device (DMD) array based modulator which collects many measurements for each frame by taking random projections of the scene in quick succession. We envision using SPC for capturing different sets of random projections at successive frames. Since the projections are multiplexed in time it would be a simple matter of running through different projections continuously only repeating them after a fixed number of frames as opposed to every frame.

Other imagers include task based imager [108], random lens imager [52] similar in principle to SPC, coded aperture imager [96] where pixels in a local region corresponding to the defocus blur are mixed. While different aperture masks can be used for consecutive frames, coded aperture imaging has a sparse measurement matrix which is not suitable for recovering the sparse foreground. Similarly a CMOS

compressive imager [68] has been proposed where the pixels are mixed once they are captured. Compressive imagers have also been built for increasing the temporal resolution for periodic videos [154] by modulating the shutter and its generalization [125]. Since in these cases the imaging is done by temporal multiplexing, our method is not applicable here. A spectral compressive imaging system [155] which multiplexes different frequencies is also available but not suitable for our application.

CS for Videos: Recent work in compressive video sensing attempts to reconstruct videos from compressed measurements by exploiting redundancy in the temporal dimension in addition to the spatial sparsity. [112] proposed compressive video sensing using the SPC architecture and reconstruction using wavelet lifting. [125] proposed a compressive camera architecture by modifying existing slower cameras to collect an order of magnitude faster videos by explicitly exploiting motion constraints during reconstruction. Similarly, CS video capture for processes satisfying a linear dynamical model was proposed by [130]. Similarly [152], [151] proposed a Kalman filtered based recovery scheme and extended it to a general framework. Marcia [95] proposed a coded aperture based compressive acquisition system for videos which are recovered using a joint compressive framework. Further, [67] proposed an algorithm based on motion compensation to recover compressive videos.

Compressive video sensing based on sensing a keyframe at full measurements and then other frames at reduced measurements have been proposed in [72], [139] and [165]. The difficulty with key frame approaches is that either it should be sensed fully or reconstructed with priors on the image which degrades its quality. Such degradation in key frames propagates to other frames as well. Compared to

other methods we do not have to change the number of measurements at every frame, only the measurement matrix changes. This way we are blind to different processes generating the video.

A compressive video coding and analysis work has been proposed in [41] where CS projections are used as video coding and simple object tracking is performed on the compressive measurements.

Foreground-background separation:

Significant amount of work exists for background subtraction on traditional videos. In [140] the background is modeled pixelwise as a mixture-of-Gaussian whereas in [48] the background is modeled at each pixel using a non-parametric density. PCA based decomposition is used in [110] to separate the background from the foreground and [76] uses a Kalman filter based approach to model the background. A good overview of these algorithms can be found in [120] and [40]. Background subtraction problem has also been formulated as robust PCA [32], [42] where the error is the foreground and the principle components capture the background. Recent work in background modeling has also taken into account the motion of the camera [88]. These algorithms can be broadly divided into two kinds, those which model the background pixel-wise and those which model the entire image based on a collection of past images. Pixel-based methods have the advantage of both increased speed and low memory requirements. Since the goal of our work is to separate the foreground from the background and reconstruct the images, modeling the background in the compressive domain is significantly challenging. This means that effective background subtraction algorithms based on mixture-of-Gaussians (MoG)

are difficult to adapt since the pixels are mixed but the inference is pixel-wise.

An algorithm for background subtraction using compressive sensing ideas was proposed in [35]. There the foreground is assumed to be spatially sparse and an adaptive background model is proposed to update the changes in the background. Nevertheless the background model is simplistic and is sensitive to moderate changes in illumination in the scene. Further, this approach recovers only the foreground (albeit with lesser measurements than traditional CS) but fails to recover the background. Retaining only the foreground information implies that the background which often provides visual context is lost. Hence, it is desirable for both visual and hyperspectral imaging to sense data with fewer measurements in the first place but where no information is lost. Our work is closely related to [35] but builds on it by proposing a joint framework for both separation of foreground and background and also estimating the video sequence.

5.2 Compressive Imaging

Compressive sensing theory [24] states that a signal sparse in a transform basis can be sensed and recovered accurately from fewer measurements of it. Let the signal $\boldsymbol{x}$ of size $N \times 1$ be sparse in some transform basis $\boldsymbol{\Psi}$. Then the transformation between the signal and its sparse transformation coefficients $\boldsymbol{s}$ is given by

$$\boldsymbol{x} = \boldsymbol{\Psi}\boldsymbol{s}. \tag{5.1}$$

The signal $\boldsymbol{x}$ can be recovered from M measurements $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}$ where $M << N$ provided the coefficients $\boldsymbol{s}$ are sparse (say $\|s\|_0 = K$) and the matrix $\boldsymbol{A} = \boldsymbol{\Phi}\boldsymbol{\Psi}$

satisfies a sufficient condition termed 'Restricted Isometry Property' (RIP) [25] for a given sparsity. $\boldsymbol{x}$ can be recovered by solving for sparse $\boldsymbol{s}$ through either convex programming such as Basis Pursuit [38] or greedy algorithms [115], [107]. It has been shown that random Gaussian or Bernoulli matrices and partial Fourier matrix satisfy RIP with a high probability.

To realize above compressive sensing scheme, it is sufficient if the sensor can take projections satisfying RIP such that it preserves the information and allows the signal to be computationally reconstructed. For instance, MRI is performed by sensing the partial Fourier measurements and since the image has bounded total variation, it is accurately reconstructed from the partial measurements. For natural images, both in visual and hyperspectral regime, SPC which sequentially takes M measurements in time corresponding to different projections is applicable. The projection matrix is drawn from a random '1' , '0' Bernoulli process. The architecture is realized using a digital micromirror device (DMD) array which modulates the incoming light and then transmits it to single pixel. Each configuration of DMD corresponds to a row of $\mathbf{\Phi}$ and M such configurations provide sufficient measurements for sensing an image.

SPC uses a single pixel and elegantly illustrates the idea of CS at its extreme but other architectures are possible which either use multiple pixels or completely novel architectures [68]. Since we are concerned with sensing a surveillance video and separating the foreground from the background we assume an architecture which senses by mixing the pixel intensities in space as done in MRI and SPC as opposed to mixing in time [154], [125], [101], [144] or frequency [155]. Nevertheless, the above

architectures are limited in their effectiveness since natural images are rarely truly sparse in any known basis. Instead they are compressible and their reconstruction quality is upper bounded by the sparse approximation to the image.

To overcome lack of redundancy in space, we can exploit redundancy in time. The similarity of consecutive frames $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$ is used while reconstructing them from their compressive measurements $\boldsymbol{y}_t = \boldsymbol{\Phi}\boldsymbol{x}_t$. The CS architecture achieves video sensing by continuously taking the same projections of the incoming light in every frame. For example, in SPC the DMD runs through M projections for a frame and then repeats the same cycle for the next frame. Similarly, MRI devices take particular Fourier measurements for a frame and cycle through them for the next frame. Note that when the architecture sequentially takes measurements we assume that the motion during each frame is limited compared to that between consecutive frames.

In this chapter, we propose to modify the architecture for sensing videos made up of two processes. Such a modification in sensing architecture can be achieved using existing CS camera realizations such as SPC without any changes to its hardware. We simply propose to take different measurements every frame such that after a small duration R the projections span the N dimensional image space. We sense M measurements $\boldsymbol{y}_t = \boldsymbol{\Phi}_t\boldsymbol{x}_t$ such that the MR dimensional concatenated measurement matrix $[\boldsymbol{\Phi}_1^T\boldsymbol{\Phi}_2^T \ldots \boldsymbol{\Phi}_R^T]^T$ has rank N. This can be achieved in SPC by simply running through the MR measurements in the duration of R frames. This simple but important modification allows us to avoid the sparsity assumption during reconstruction of a slowly changing video. If the frame $\boldsymbol{x}_1$ is similar to $\boldsymbol{x}_R$ then we

can reconstruct them without any assumption of some transform domain sparsity. When the frames $\boldsymbol{x}_1$ and $\boldsymbol{x}_R$ differ in appearance due to motion of objects in the foreground we make no sparsity assumptions on the slowly changing background but assume that the quickly changing foreground is spatially sparse. In a surveillance video each frame $\boldsymbol{x}_t$ can be decomposed into two processes, a quickly varying foreground $\boldsymbol{x}_t^f$ and a slowly varying background $\boldsymbol{x}_t^b$.

$$\boldsymbol{x} = \boldsymbol{x}_t^f + \boldsymbol{x}_t^b \tag{5.2}$$

For example in Figure 5.2, the moving clouds are slowly changing and treated as background. The illumination change is faster than that of clouds but slower than the foreground and is eventually incorporated in the background. The moving person is treated as foreground since the motion is significantly large between consecutive frames. The car which was moving quickly was initially treated as foreground but eventually incorporated into the background when it became stationary. In Figure 5.1 the motion of foliage is fast and hence treated as foreground for video recovery.

In an earlier work [35], a technique to sense surveillance videos using CS architecture was proposed. In it the number of measurements needed are dependent on the sparsity of foreground and a procedure to separate and estimate the foreground was proposed. The procedure was based on updating the background and performed satisfactorily on a simple dataset but failed on the challenging PETS dataset. The primary drawback of the approach was that only the foreground could be estimated. This meant that the context in which the foreground object is moving was unavailable.

In this chapter we propose a joint compressive video sensing and background subtraction method by modeling the video as a background and a foreground process. We apply the distributed CS framework [11] over frames $\boldsymbol{x}_t$ in a temporal window of size W. The frames though different have similarities and this fact is leveraged to sense the frames (both background and foreground) using the same number of measurements as previous technique [35]. This is made possible by modifying the sensing scheme which allows us to sense the background by avoiding sparsity assumption thereby increasing the quality of reconstruction.

DCS is a framework for compressively sensing signals generated by distributed sources but which share similarities. For example, a signal $\boldsymbol{x}_t$ indexed by the source t (frame number in our case) could share common sparse support of its coefficient $\boldsymbol{s}_t$ with other signals. Or the signals could have a common component (sparse or non-sparse) and a sparse innovation.

5.3 Compressive Video Reconstruction and Background Subtraction

Given a sequence of frames, we define the background as the component common to all the frames in the window and the foreground as the respective innovation in each frame. In background subtraction [140], [48] on images, the background pixels are determined based on the multi-modal distribution at the pixel. Adapting these techniques to compressive frames involves making inferences about the pixel distribution's membership to one of the mixtures. Such an inference, without explic-

itly reconstructing the frames from the projections, is hard due to the exponential number of possibilities. The same inference on images has only linear number of possibilities since each pixel is treated independently but the compressive projections are a mixture of pixel intensities. This makes multi-modal foreground-background separation extremely difficult.

Hence, we define an intuitive, appearance based foreground and background in the compressive domain. Our approach can be interpreted as modeling the background using a single distribution thereby avoiding the exponential possibilities involved in using MoG model. Given a sequence of frames $\boldsymbol{x}_{t-W+1}, \boldsymbol{x}_{t-W+2}, \ldots, \boldsymbol{x}_t$ where $W > R$, we define background as the image component common to this collection of W frames and the foreground of individual frames as the innovations which are different from the common component

$$\boldsymbol{x}_{t+i} = \boldsymbol{x}_t^b + \boldsymbol{x}_{t+i}^f. \tag{5.3}$$

Since, there is no unique way of defining the background or the innovations, we impose the condition that $\boldsymbol{x}_{t+i}^f$ should be sparse i.e. it has a sparse support $\{\Omega_i\}$. Such an assumption is valid especially in far field surveillance videos where the foreground regions is small and the foreground component is assumed to be sparse.

In the compressive domain, where we sense measurements $\boldsymbol{y}_{t-W+1}, \boldsymbol{y}_{t-W+2}, \ldots, \boldsymbol{y}_t$, we define the background and the foreground similarly.

$$\boldsymbol{y}_{t+i} = \boldsymbol{\Phi}_{t+i}\boldsymbol{x}_{t+i} \tag{5.4}$$

and

$$\boldsymbol{y}_{t+i} = \boldsymbol{y}_{t+i}^b + \boldsymbol{y}_{t+i}^f \tag{5.5}$$

Here $\boldsymbol{y}_{t+i}^b = \boldsymbol{\Phi}_{t+i}\boldsymbol{x}_t^b$ is the background projected on the measurement matrix at time $t+i$ and $\boldsymbol{y}_{t+i}^f = \boldsymbol{\Phi}_{t+i}\boldsymbol{x}_{t+i}^f$ is the compressive component of the sparse foreground innovations. We drop the subscript t where the context is obvious and we retain it when specific reference to time is made. Note that our goal is to estimate the sparse $\boldsymbol{x}_i^f$ and its support as well as the common component $\boldsymbol{x}^b$ from the sequence of compressive frames $\boldsymbol{y}_{-W+1}, \boldsymbol{y}_{-W+2}, \ldots, \boldsymbol{y}_0$ without explicitly recovering the corresponding frames $\boldsymbol{x}_{-W+1}, \boldsymbol{x}_{-W+2}, \ldots, \boldsymbol{x}_0$. Our formulation is inspired by the idea of distributed compressive sensing (DCS), especially the joint signal model (JSM3) where the sources have a common non-sparse component and a sparse innovation component. It has been shown in [11] that both the common and innovation components can be extracted from the compressive measurements at a central location through an alternating algorithm.

We recover the background image $\boldsymbol{x}^b$ without any sparsity assumptions by solving a simple least-squares problem as described below. Though the background image is compressible in wavelet basis we avoid enforcing it. Instead we rely on the fact that different projections $\boldsymbol{y}_i^b$ of the background image span the entire space i.e. $[\boldsymbol{\Phi}_{-W+1}^T \boldsymbol{\Phi}_{-W+2}^T \ldots \boldsymbol{\Phi}_0^T]^T$ is full rank. Our iterative least squares-based algorithm is based on the following idea.

Suppose we have an accurate estimate of the compressive background $\hat{\boldsymbol{x}}^b$, then we can subtract its contribution $\hat{\boldsymbol{y}}_i^b = \boldsymbol{\Phi}_i\hat{\boldsymbol{x}}^b$ from the compressive frames $\boldsymbol{y}_i$ and reconstruct the sparse foreground $\hat{\boldsymbol{x}}_i^f$ and its support $\{\hat{\Omega}_i\}$. Instead if the sparse foreground $\hat{\boldsymbol{x}}_i^f$ and its support $\{\hat{\Omega}_i\}$ are known, we partition the measurements into two parts: the projection onto span$(\{\boldsymbol{\Phi}_i\}_{n\in\hat{\Omega}_i})$ and the component orthogonal to

that span. We define a $M \times (M - |\hat{\Omega}_i|)$ matrix $\boldsymbol{Q}_i$ whose orthogonal columns span the orthogonal complement of $\Phi_{\hat{\Omega}_i}$.

We then define the component of the compressive frame which does not have any contribution from those of the foreground columns

$$\tilde{\boldsymbol{y}}_i = \boldsymbol{Q}_i^T \boldsymbol{y}_i \tag{5.6}$$

and similarly the component of the measurement matrix without the contribution from foreground columns as

$$\tilde{\boldsymbol{\Phi}}_i = \boldsymbol{Q}_i^T \boldsymbol{\Phi}_i. \tag{5.7}$$

The modified measurements $\tilde{\boldsymbol{y}} = [\tilde{\boldsymbol{y}}^T_{-W+1} \tilde{\boldsymbol{y}}^T_{-W+2} \dots \tilde{\boldsymbol{y}}^T_0]^T$ and the modified basis $\tilde{\boldsymbol{\Phi}} = [\tilde{\boldsymbol{\Phi}}^T_{-W+1} \tilde{\boldsymbol{\Phi}}^T_{-W+2} \dots \tilde{\boldsymbol{\Phi}}^T_0]^T$ can be used to refine the estimate of the compressive background.

$$\hat{\boldsymbol{x}}^b = arg\min_{\boldsymbol{x}^b} \sum_{i=-W+1}^{0} \|\tilde{\boldsymbol{y}}_i - \tilde{\boldsymbol{\Phi}}_i \boldsymbol{x}^b\|^2 \tag{5.8}$$

$$\hat{\boldsymbol{x}}^b = \tilde{\boldsymbol{\Phi}}^{\dagger} \tilde{\boldsymbol{y}} \tag{5.9}$$

We use the above fact to refine the foreground and background estimates by alternating between estimates of each. If the estimate of the background is not accurate then we can use the estimated foreground to refine it and vice versa. The alternating approach used for estimating the foreground is presented in Algorithm 1.

5.3.1 Compressive Video Reconstruction

In practice the least squares problem in equation (5.8) is solved using incremental gradient method [18] since the size of the images prohibits us from computing

Algorithm 1: Joint Compressive Video Reconstruction and Background Subtraction: Batch

Input: Compressive measurements $\boldsymbol{y}_{-W+1}, \boldsymbol{y}_{-W+2}, \ldots, \boldsymbol{y}_0$ and corresponding measurement matrices $\boldsymbol{\Phi}_{-W+1}, \boldsymbol{\Phi}_{-W+2}, \ldots, \boldsymbol{\Phi}_0$.

Output: Background estimate $\hat{\boldsymbol{x}}^b$, foreground estimate $\hat{\boldsymbol{x}}_i^f$ and foreground region $\{\hat{\Omega}_i\}$ for $i = -W+1, -W+2, \ldots, 0$.

Initialize:

Set $\hat{\Omega}_i = \emptyset$ for $i = -W+1, -W+2, \ldots, 0$.

Set the iteration counter $l = 1$.

while $l < L$ **do**

Estimate background:

Update background $\hat{\boldsymbol{x}}^b$ according to equation (5.8).

Compressive background estimate $\hat{\boldsymbol{y}}_i^b = \boldsymbol{\Phi}_i \hat{\boldsymbol{x}}^b$.

Estimate foreground support:

For $i = -W+1, -W+2, \ldots, 0$, subtract compressive background $\hat{\boldsymbol{y}}_i^b$ from compressive frame, $\hat{\boldsymbol{y}}_i = \boldsymbol{y}_i - \hat{\boldsymbol{y}}_i^b$.

Estimate the sparse support $\hat{\Omega}_i$ of the foreground $\boldsymbol{x}_i^f$.

end

Estimate the foreground:

For $i = -W+1, -W+2, \ldots, 0$, estimate the foreground $\hat{\boldsymbol{x}}_{i,\hat{\Omega}_i}^f = \boldsymbol{\Phi}_{i,\hat{\Omega}_i}^{\dagger} \hat{\boldsymbol{y}}_i$.

Output the background $\hat{\boldsymbol{x}}^b$.

the large matrix inverse. Note that the definition of background in compressive domain is robust and can capture the changes in illumination but it cannot handle the faster background variation such as the motion of a tree. As mentioned earlier, our pixel is modeled using an unimodal distribution and hence we cannot handle multi-modal distributions as modeled by MoG-based methods.

Recall that the above foreground background separation is for frames $x_{t-W+1}, x_{t-W+2}, \ldots, x_t$. To find the background and foreground for the next set of frames, obtained by shifting the window by a single frame, we would have to apply Algorithm 1 to frames $x_{t-W+2}, x_{t-W+3}, \ldots, x_{t+1}$. At time instances t and $t+1$ we calculate the backgrounds $\boldsymbol{x}_t^b$ and $\boldsymbol{x}_{t+1}^b$ respectively and also calculate two foreground sets of size W each. Such an approach is computationally inefficient since the background $\boldsymbol{x}_{t+1}^b$ of the new set is only slightly different from $\boldsymbol{x}_t^b$. Similarly, the foregrounds differ little as well. Also, the foreground computation is unnecessary except for frame $t+1$. In such batch-based approach, every frame $t+i$ is involved in computation of W different backgrounds as the windows are shifted past it and hence W different foregrounds corresponding to each of the wondows. Moreover, it solves the expensive sparse recovery problem LW times before estimating the background and W foregrounds. To overcome the computationally inefficiencies in the batch algorithm we propose an incremental approach where only a single foreground is estimated at every frame. The background for the frame is estimated based on the W compressive measurements and estimated foregrounds. Once we compute foregrounds for frames j ($j < t$), we do not compute them again. We estimate the foreground $\hat{\boldsymbol{x}}_t^f$ given the compressed frame $\boldsymbol{y}_t$, background estimate $\hat{\boldsymbol{x}}_t^b$

and the measurement matrix $\boldsymbol{\Phi}_t$. The background $\hat{\boldsymbol{x}}_t^b$ is estimated from the previous set of W compressed frames $\boldsymbol{y}_{t-W}, \boldsymbol{y}_{t-W+1}, \ldots, \boldsymbol{y}_{t-1}$ and estimated foreground $\hat{\boldsymbol{x}}_{t-W}^f, \hat{\boldsymbol{x}}_{t-W+1}^f, \ldots, \hat{\boldsymbol{x}}_{t-1}^f$ according to equation (5.8). We propose Algorithm 2 to find the compressive background and the foreground.

5.3.2 Background Subtraction

If the measurement matrix is not changed at every frame as in traditional CS imaging, we have $\boldsymbol{\Phi}_t = \boldsymbol{\Phi}$. Then equations (5.4) and (5.5) are modified as

$$\boldsymbol{y}_{t+i} = \boldsymbol{\Phi}\boldsymbol{x}_{t+i} \tag{5.10}$$

and

$$\boldsymbol{y}_{t+i} = \boldsymbol{y}^b + \boldsymbol{y}_{t+i}^f \tag{5.11}$$

Since each frame has same projection, it is impossible to estimate the background $\hat{\boldsymbol{x}}^b$ without additional priors. But the foreground $\hat{\boldsymbol{x}}_i^f$ can be estimated by enforcing spatial sparsity. By estimating the compressive background $\hat{\boldsymbol{y}}^b$ and then subtracting from the compressed frame we retain only the foreground information which can be recovered through sparse recovery. To estimate the compressive background we solve

$$\hat{\boldsymbol{y}}^b = arg\min_{\boldsymbol{y}^b} \sum_{i=-W+1}^{0} \|\tilde{\boldsymbol{y}}_i - \boldsymbol{Q}_i^T \boldsymbol{y}^b\|^2. \tag{5.12}$$

The algorithm for the foreground is a modification of Algorithm 1 and is given by Algorithm 3.

Algorithm 2: Joint Compressive Video Reconstruction and Background Subtraction: Incremental

Input: Compressive measurements $\boldsymbol{y}_t$, measurement matrices $\boldsymbol{\Phi}_t$ for

$t = 1, 2, \ldots, T$.

Output: Background estimate $\hat{\boldsymbol{x}}_t^b$, foreground estimate $\hat{\boldsymbol{x}}_t^f$ and foreground region

$\{\hat{\Omega}_t\}$ for $t = 1, 2, \ldots, T$.

Initialize:

Given compressive measurements $\boldsymbol{y}_t$, measurement matrices $\boldsymbol{\Phi}_t$ for $t = 1, 2, \ldots, W$, use Algorithm 1 to estimate the foregrounds $\hat{\boldsymbol{x}}_t^f$, foreground regions $\{\hat{\Omega}_t\}$ and orthogonal basis $\boldsymbol{Q}_t$ for $t = 1, 2, \ldots, W$,.

Set $t = W + 1$.

while $t \leq T$ **do**

Estimate background:

Use the compressive measurements $\boldsymbol{y}_{t-i}$, measurement matrices $\boldsymbol{\Phi}_{t-i}$ and orthogonal basis $\boldsymbol{Q}_{t-i}$ for $i = 1, 2, \ldots, W$ in equation (5.8) to estimate the background $\hat{\boldsymbol{x}}_t^b$.

Compressive background estimate $\hat{\boldsymbol{y}}_t^b = \boldsymbol{\Phi}_t \hat{\boldsymbol{x}}_t^b$.

Estimate foreground support:

Subtract compressive background estimate $\hat{\boldsymbol{y}}_t^b$ from the compressive frame, $\hat{\boldsymbol{y}}_t = \boldsymbol{y}_t - \hat{\boldsymbol{y}}_t^b$.

Estimate the sparse support $\hat{\Omega}_t$ of the foreground $\boldsymbol{x}_t^f$.

Estimate the foreground:

Estimate the foreground $\hat{\boldsymbol{x}}_{t,\hat{\Omega}_t}^f = \boldsymbol{\Phi}_{t,\hat{\Omega}_t}^{\dagger} \hat{\boldsymbol{y}}_t$.

Output the background $\hat{\boldsymbol{x}}_t^b$.

end

Algorithm 3: Background Subtraction: Batch

Input: Compressive measurements $\boldsymbol{y}_{-W+1}, \boldsymbol{y}_{-W+2}, \ldots, \boldsymbol{y}_0$ and measurement matrices $\boldsymbol{\Phi}$.

Output: Foreground estimate $\hat{\boldsymbol{x}}_i^f$ and foreground region $\{\hat{\Omega}_i\}$ for $i = -W+1, -W+2, \ldots, 0$.

Initialize:

Set $\hat{\Omega}_i = \emptyset$ for $i = -W+1, -W+2, \ldots, 0$.

Set the iteration counter $l = 1$.

while $l < L$ **do**

Estimate background:

Update compressive background estimate $\hat{\boldsymbol{y}}^b$ according to equation (5.12).

Estimate foreground support:

For $i = -W+1, -W+2, \ldots, 0$, subtract compressive background estimate $\hat{\boldsymbol{y}}^b$ from compressive frame, $\hat{\boldsymbol{y}}_i = \boldsymbol{y}_i - \hat{\boldsymbol{y}}^b$.

Estimate the sparse support $\hat{\Omega}_i$ of the foreground $\boldsymbol{x}_i^f$.

end

Estimate the foreground:

For $i = -W+1, -W+2, \ldots, 0$, estimate the foreground $\hat{\boldsymbol{x}}^f_{i,\hat{\Omega}_i} = \boldsymbol{\Phi}^\dagger_{i,\hat{\Omega}_i} \hat{\boldsymbol{y}}_i$.

5.4 Experiments

In this section, we present the experimental results on recovery of videos from their compressive measurements. We perform our experiments on the PETS 2001 dataset which is a typical outdoor surveillance video. Currently, our approach has not been tested on videos from compressive imager but it is applicable to sensing frameworks where each frame is sensed with a different, dense measurement matrix.

The PETS dataset is challenging since the background changes over time. For example, cars which are initially moving become stationary and become part of the background and vice versa. There is also the slow motion of clouds in the background and this leads to changes in illumination which need to be incorporated in to background as well. Further, there is motion of the foliage which while quickly changing is part of the background.

First, we show the performance of the video reconstruction algorithm on the dataset. Then we evaluate the stand alone background subtraction algorithm.

5.4.1 Compressive Video Reconstruction

For joint compressive video reconstruction and background subtraction we sense each frame using $M = 0.2N$ measurements. We vary the measurement matrix in each frame and cycle through the same set of matrices after setting $R = 20$ ensuring that the concatenated matrix is of rank N. We set the temporal window $W = 50$ and estimate the background using the procedure described in Algorithm 2. Initialization of the background and the foregrounds necessary to define the

backgrounds in later frames is described below. For solving the least squares problem in equation (5.8) we use the incremental gradient method presented in [18]. Since solving least squares of size N with MW constraints is computationally expensive we rely on an incremental approach. For solving the sparse foreground we use SPGL1 [148], [147] an implementation of basis pursuit denoising (BPDN). While the foreground is sparse it also occupies contiguous pixels. This fact was exploited in an MRF based approach to recover sparse signals [34] but we do not enforce this constraint in our recovery. We note that, enforcing joint sparsity enables significant decrease in number of measurements needed to solve for sparse signal but here we focus on our ability to extract the foreground and reconstruct the background. Once we estimate the foreground $\hat{\boldsymbol{x}}_t^f$, pixels above a threshold are assigned to the foreground region $\{\hat{\Omega}_t\}$.

We calculate the projection of compressive measurement $\boldsymbol{y}_i$ on the basis $\boldsymbol{Q}_i$ orthogonal to the foreground columns $\{\boldsymbol{\Phi}_i\}_{n\in\hat{\Omega}_i}$, by first performing QR decomposition on the foreground columns and then removing from $\boldsymbol{y}_i$ its projection on the span$(\{\boldsymbol{\Phi}_i\}_{n\in\hat{\Omega}_i})$. We note that this step is computationally expensive but can be performed off-line during the recovery of the sensed video. The projection of the the compressed measurements onto the orthogonal basis of foreground effectively prevents the foreground from contributing to the estimate of the background in the subsequent frames. In computationally constrained scenarios, at loss of some quality in reconstruction of background, the effect of foreground can be subtracted from the compressive measurement instead.

The results of video reconstruction on two different surveillance videos are

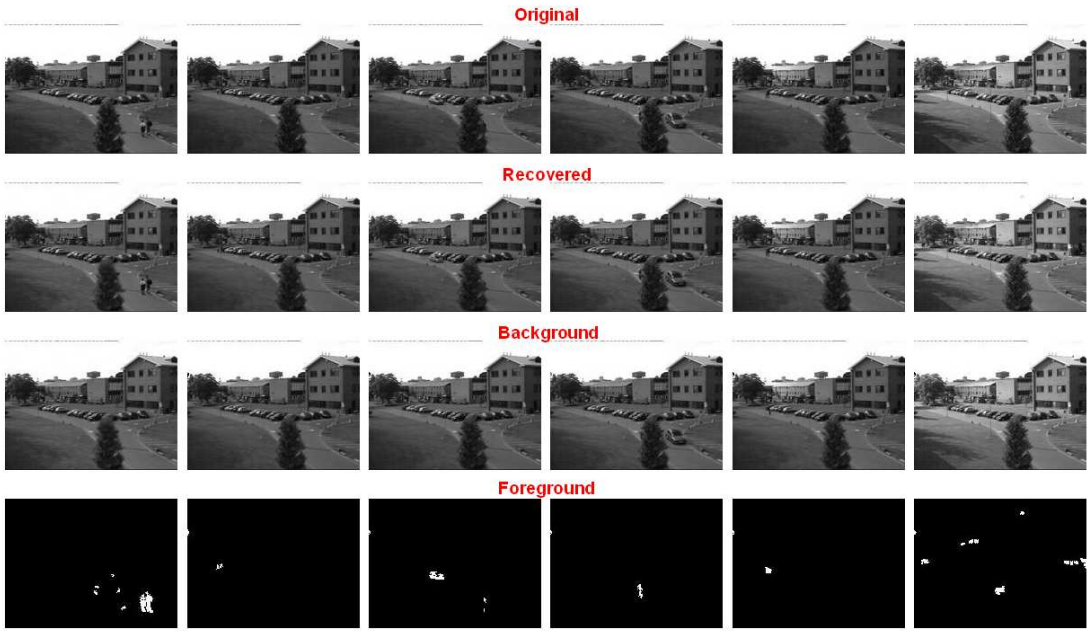

Figure 5.1: Reconstruction of video from compressive measurements. The video is reconstructed by decomposing it into a slowly changing background and a quickly changing foreground. Note that our method handles changes in illuminations and incorporates them in the background.

presented in Figures 5.1 and 5.2. In the first video of a outdoor scene people and cars are moving against a backdrop of buildings and trees. Over the course of the video the the scene brightened due to change in illumination. Our approach accurately estimates the background and the foreground. In the first three frames, moving people and car have been accurately detected and estimated. Note that there is also foreground detection corresponding to the edges of the tree due to its swaying. Since we model the background unimodally, we have classified few of the tree's edges as foreground. The car in the third frame is now nearer to the tree in the fourth frame. In the video the car briefly stops at this location to backup. Our approach accurately incorporates the stationary car in the background. In the fifth frame, we capture the changing illumination at the far end of the building and

correctly incorporate it in the background. Finally, in the last frame when there is a significant change in foreground our background model adapts. In the foreground there are some false positives in few locations but overall our algorithm performs well.

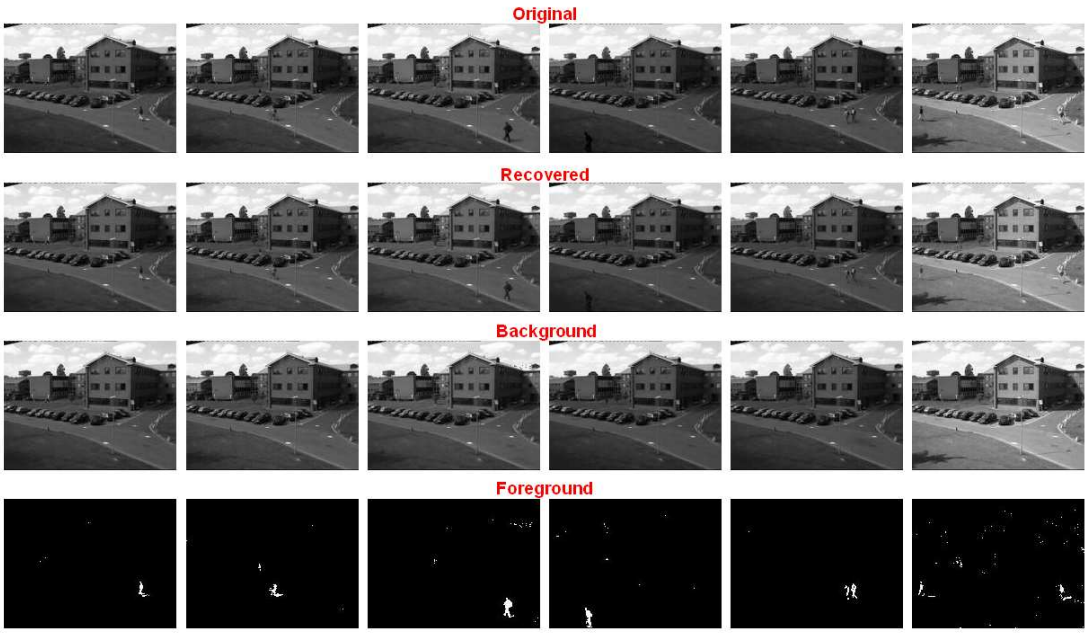

Figure 5.2: Reconstruction of video from compressive measurements. Note that even the small changes in the shape of the cloud is accurately reconstructed.

The second video is similar to the first scene but from slightly different viewpoint. Over the video, the brightness in the scene rises a bit, drops and then rises dramatically due to motion of the clouds. Despite a significant change in illumination, our method can accurately estimate the background and the foreground. Like the previous video, the final frame has false positives in the foreground at few locations due to quick change in illumination. Note that clouds in the sky slowly move over the course of the video and we captured their subtle change in shape.

Since our method is appearance based, scenes with camouflage cannot be de-

tected. If the foreground and the background has the same intensity in a frame then they are both classified as a background. Also, since background is modeled using an unimodal distribution, we classify motion due to foliage and large changes in illumination as foreground. To eliminate false positives, a MoG [140] based foreground detector can be applied on the estimated foreground images. Such a post-processing step can significantly remove the spurious foreground compared to the simple thresholding step described above.

In our experiments, we initialize the background and foreground by implementing the batch background and foreground estimating Algorithm 1. Since, the computations are performed offline, the background and foreground corresponding to each of the frames in the window are estimated in an iterative fashion. Once, the foregrounds are available then background for the next frame is estimated according to equation 5.8.

5.4.2 Background Subtraction

Here, we experiment on only the background subtraction version of our algorithm. When the measurement matrix is not changed over time, additional priors are needed to estimate the background. Since sparse priors result in unsatisfactory reconstruction of images, we estimate only the foreground from these frames. Since the foreground images are spatially sparse, they can be reconstructed even if the background is not. This was illustrated in [35] where a model for the compressive background is adaptively built and then subtracted from the compressive

measurement to estimate the foreground. In our approach, we similarly estimate the compressive background but with two important differences. In [35] during the update of the background model, the foreground from the previous frame is excluded by subtracting it from the compressive measurement. In our case, we do not subtract the foreground component but instead project the compressive measurements onto subspace orthogonal to the foreground columns. This completely prevents the foreground columns from contributing to the estimate of the background. Whereas [35] estimates the foreground needed for background update from a parallel process, we use the foreground estimated from the previous estimates to update the next background. Our approach is principled since we define the background as that which is common to a collection of frames and devise and algorithm to estimate it and the unique innovations.

We experiment on a video from PETS 2001 dataset by setting $M = 0.2N$ and implementing Algorithm 3. The results of our approach in comparison with [35] are shown in Figure 5.3. Note that our approach absorbs a stationary car into the background and performs significantly better than [35]. Contrast the third and fourth frames. Note that pixels corresponding to car in the third frame are labeled foreground in the fourth frame. This is because the algorithm is adapting to the fact that the car in the third frame was stationary and was eventually absorbed into the background. Subsequently when the car moved to the location shown in the fourth frame, both the previous and the current location are labeled foreground. This disappears after a few frames in the fifth frame.

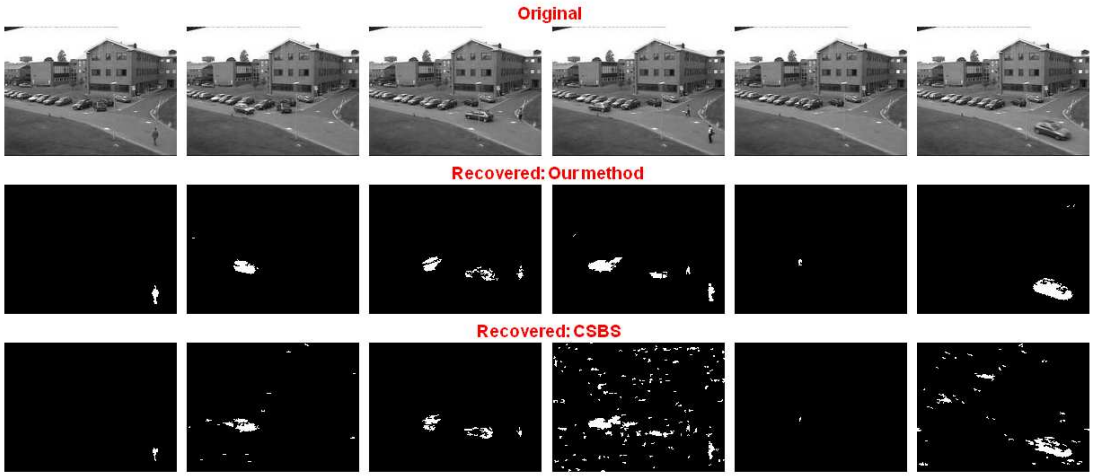

Figure 5.3: Background subtraction on compressive video. Compared to [35] our method is accurate in estimating the foreground.

5.5 Conclusions

In this chapter, we presented a compressive sensing and video reconstruction approach for surveillance videos. Our method performs joint video reconstruction and background subtraction by formulating the problem in a distributed compressive sensing framework. The video is modeled by decomposing into two parts, a slowly varying background and quickly varying but spatially sparse foreground. By sensing each frame with a unique set of measurement projections, the video is reconstructed by separately reconstructing the background and the foreground using a simple iterative algorithm. We performed experiments on standard PETS 2001 dataset to illustrate the effectiveness of our method.

Chapter 6

Enforcing integrability via ℓ_1 minimization

6.1 Introduction

Surface reconstruction from gradient fields is an important final step in many vision and graphics applications involving gradient domain processing. These can be broadly classified as (a) manipulating gradients and/or (b) estimating gradients before integration. Methods such as Photometric Stereo (PS) [161] and Shape from Shading (SfS) [63] estimate the gradient field of the surface from captured images. Applications such as image editing [116], stitching [84], HDR compression [50] etc. first apply local/global manipulations to the gradient field of single/multiple images. The final image is then reconstructed from the modified gradient field. The reader is referred to the recent course [5] for more details.

Typically, the resulting gradient field is non-integrable due to linear/non-linear gradient manipulations, or due to the presence of noise/outliers in gradient estimation (figure 6.1). For a reconstruction algorithm, two important considerations are (a) robustness or ability to handle outliers and (b) local error confinement (LEC) [3].

Robustness means that surface features are well reconstructed in the presence of outliers. A related property is LEC, which ensures that distortions in the integrated surface are confined spatially close to the errors in the underlying gradient field.

It is well-known that least squares estimate is not robust in presence of outliers. While several integration techniques have been proposed before, we analyze robust surface integration as an error correction problem. We are inspired from recent work in compressed sensing [24], particularly the $\ell_0 - \ell_1$ equivalence. We propose to obtain the 3D surface by finding the gradient field which best fits the corrupted gradient field in the ℓ_1-norm sense. While minimizing the ℓ_1-norm is not new as a robust statistic, we analyze the properties of ℓ_1 solution and provide new insights using linear algebra and graph analogy. We compare with existing techniques and show that ℓ_1 solution performs well across all scenarios without the need for any tunable parameter adjustments.

6.1.1 Contributions

- We analyze robust gradient integration as error correction by utilizing ideas from sparse signal recovery literature.
- We show that the location of errors is as important as the number of errors for gradient integration, which is not typically explored when considering $\ell_0 - \ell_1$ equivalence.
- We exhaustively analyze the properties of ℓ_1 solution in terms of robustness and LEC for various outlier patterns and noise in given gradient field.

6.1.2 Related work

Enforcing integrability: The simplest approach is to find an *integrable* gradient field (or the surface) which best fits the given gradient field, by minimizing the least squares cost function. This amounts to solving the Poisson equation [138]. Frankot & Chellappa [54] project the given gradient field onto integrable functions using Fourier basis to enforce integrability. Cosine basis functions were proposed in [56], while Kovesi [77] proposed *shapelets* as a redundant set of basis functions. Petrovic et al. [118] used a loopy belief propagation algorithm to find the corresponding integrable field. Methods based on ℓ_2-norm cannot handle large outliers in the gradient field.

Robust estimation: There has been large body of work on robust parametric estimation using RANSAC [53], which becomes combinatorial in nature as the number of parameters increases. For gradient integration on $N \times N$ grid, there are N^2 unknowns (pixel values) and $2N^2$ observations (x and y gradients). Thus, RANSAC is computationally prohibitive [6]. M-estimators modify the least squares cost function to reduce the influence of outliers. Several such influence functions such as Huber, Tukey, etc. have been proposed [64, 122].

Agrawal et al. [6] proposed a general framework for robust gradient integration by gradient transformations, such as anisotropic weighting and affine transformations. The diffusion algorithm in [6] solves a modified Poisson equation by applying edge preserving affine transformations to the gradient field. To calculate the local surface edge direction, the algorithm uses gradient values in a neighborhood. We

show that it performs poorly when the neighborhood of an edge is corrupted by outliers.

Our approach instead minimizes the ℓ_1-norm of gradient errors. Minimizing the ℓ_1-norm has been shown to be effective in correcting outlier errors and recovering sparse signals [38, 73, 84]. Traditionally, the ℓ_1-norm is not preferred since the cost function is not analytically differentiable and minimization is computationally expensive. However, there has been a renewed interest in using ℓ_1 cost functions due to $\ell_0 - \ell_1$ equivalence for sparse reconstructions under the *restricted isometry property* (RIP). We use RIP to show that for gradient integration, the location of outliers is as important as their number. In addition, we use the expander graph structure of gradient-curl pairs to understand the distribution of outliers which can be corrected.

Graph-based approach: To avoid the combinatorial nature of RANSAC, a greedy graph based technique was proposed in [3]. This approach treats the underlying 2D grid as a graph, gradients as edges and unknown surface values as nodes. The outlier gradients are heuristically determined by thresholding the curl values over the graph and the corresponding edges are removed. If the graph remains connected, surface could be integrated using the remaining edges (gradients). Else, a minimal set of edges are chosen to connect the graph by assigning edge weights using gradient magnitude or curl values. However, the underlying heuristic of using curl values as a 'goodness' measure often fails in presence of noise. We show that [3] performs poorly in the presence of noise and that minimizing the ℓ_1-norm effectively handles noise as well as corrects sparsely distributed outliers in the gradient field.

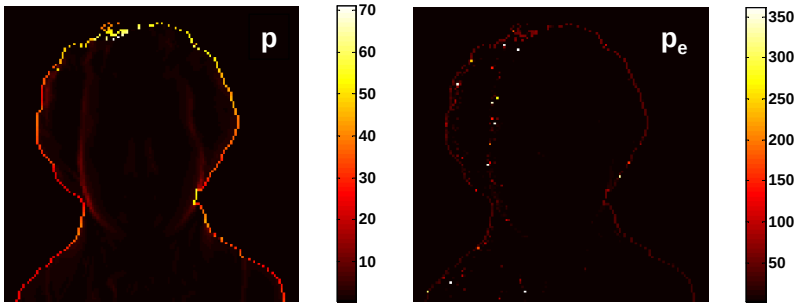

Figure 6.1: (Left) Ground truth p for Mozart (Right) Outliers along possible shadow regions in the gradient field obtained from PS. The magnitude of the outliers is 5 times the largest ground truth gradient values.

In addition, when the outliers are concentrated, the LEC property is maintained.

Denoising and TV regularization: Image denoising is a classical problem and several approaches for feature preserving denoising have been successfully demonstrated. Anisotropic filtering [117] takes into account the local edge direction in a PDE based framework. Rudin et al. [128] proposed total variation (TV) regularization, which penalizes the ℓ_1-norm of the gradients of the estimated (denoised) image. Note that our approach is different: we minimize the ℓ_1-norm of *gradient errors*, not gradients themselves. Thus, we do not employ any assumptions on the underlying surface such as natural image statistics (distribution of gradients is peaked at zero).

6.2 Gradient integration as error correction

We use the terminology from [3]. Let $S(y, x)$ be the desired surface over a rectangular grid of size $H \times W$. In vector form, we denote it by $\mathbf{s}$. Let (p, q) denote the given non-integrable gradient field, possibly corrupted by noise and outliers. The goal is to estimate S from (p, q). The integrable gradient field of S is given by

the forward difference equations

$$\begin{aligned} p^0(y,x) &= S(y,x+1) - S(y,x) \\ q^0(y,x) &= S(y+1,x) - S(y,x). \end{aligned} \tag{6.1}$$

In vector form (6.1) can be written as

$$\mathbf{Ds} = \begin{bmatrix} p^0 \\ q^0 \end{bmatrix} = \mathbf{g}^0, \tag{6.2}$$

where $\mathbf{g}^0$ denotes the stacked gradients and $\mathbf{D}$ denotes the gradient operator matrix. Each row of $\mathbf{D}$ has two non-zero entries: ± 1 in pixel positions corresponding to that particular gradient. The curl of the gradient field can be defined as loop integrals around a box of four pixels [3]

$$\texttt{curl}(y,x) = p(y+1,x) - p(y,x) + q(y,x) - q(y,x+1)$$

which can be written as

$$\mathbf{d} = \mathbf{C} \begin{bmatrix} p \\ q \end{bmatrix} = \mathbf{Cg}. \tag{6.3}$$

Here, $\mathbf{d}$ denotes the vector of stacked curl values and $\mathbf{C}$ denotes the curl operator matrix. Each row of $\mathbf{C}$ has only four non-zero entries (± 1) corresponding to the gradients associated with the loop integral.

Since the gradient field $\mathbf{g}^0$ is integrable, $\mathbf{Cg}^0 = 0$. However, for the given non-integrable gradient field $\mathbf{g}$, $\mathbf{Cg} \neq 0$. Decomposing $\mathbf{g}$ as the sum of $\mathbf{g}^0$ and a *gradient error field* $\mathbf{e}$, we get

$$\mathbf{g} = \mathbf{g}^0 + \mathbf{e} = \mathbf{Ds} + \mathbf{e}. \tag{6.4}$$

Applying the curl operator on both sides, we obtain

$$\mathbf{d} = \mathbf{C}\mathbf{e} \tag{6.5}$$

Thus, integrability can also be defined as error correction: the goal is to estimate the gradient error field $\mathbf{e}$ given the curl $\mathbf{d}$ of the corrupted gradient field. Note that in this formulation, there are $M = HW$ knowns (curl values) and $N = 2HW$ unknowns (error gradients), leading to an under-determined system of linear equations. We use $\|\cdot\|_p$ to denote the ℓ_p-norm. $\|\mathbf{e}\|_0$ simply counts the nonzero elements of $\mathbf{e}$.

Poisson solver finds a least squares fit to the gradients by solving

$$\widehat{\mathbf{e}} = \arg\min \|\mathbf{e}\|_2 \quad \texttt{s.t.} \quad \mathbf{d} = \mathbf{C}\mathbf{e}. \tag{6.6}$$

The least squares estimate is optimal when the gradient errors obey a Gaussian distribution. If the errors contain outliers, then the estimate is skewed leading to severe artifacts in the reconstructed surface or image. Outliers in the gradient field can be understood as arbitrarily large errors and could obey any distribution. An example of the errors in gradients obtained from PS is shown in figure 6.1.

ℓ_0-minimization: An approach to handle outliers is to combinatorially search for the possible locations of outliers, estimate them subject to the curl constraint (6.5) and pick the combination which satisfies the constraints the best. This can be written mathematically as

$$\widehat{\mathbf{e}} = \arg\min \|\mathbf{e}\|_0 \quad \texttt{s.t.} \quad \mathbf{d} = \mathbf{C}\mathbf{e}. \tag{6.7}$$

This problem is NP-hard and hence computationally infeasible.

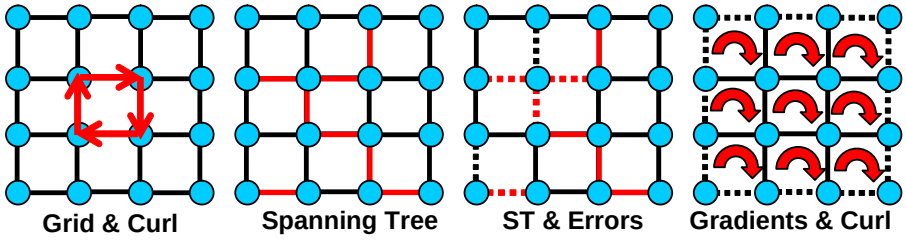

Figure 6.2: (a) Graph with pixels as nodes & gradients as edges. Curl is calculated along 2×2 loops (b) Spanning tree edges in black (c) Gradient errors in dashed lines (d) Solid black lines and red curl loops have expander graph structure

ℓ_1-minimization: Instead, we solve a convex relaxation of (6.7) by replacing the ℓ_0-norm of the gradient error $\mathbf{e}$ with the ℓ_1-norm. The conditions under which this equivalence holds true are described in detail in Sec. 6.4.

$$\widehat{\mathbf{e}} = \arg\min \|\mathbf{e}\|_1 \quad \texttt{s.t.} \quad \mathbf{d} = \mathbf{C}\mathbf{e}. \tag{6.8}$$

Equation (6.8) can be solved using convex optimization algorithms in polynomial time.

6.3 Graph based interpretation

In [3], a graph-based interpretation is provided for integrating the gradient field corrupted by outliers. We discuss this method and borrow its framework to explain our approach. [3] treats the pixel grid as a graph (G, E), where the pixels are the nodes of the graph and gradients correspond to the edges of the graph (figure 6.2(a)). Let us first assume that the location of outliers (bad gradients) are known. [3] proposes to remove the corresponding edges from the graph. If the resulting sub-graph remains connected, then integration could be done using the remaining edges/gradients. Else, the graph is connected using a minimal set of

edges, by assigning edge weights based on gradient magnitude or curl values. Since in practice, the locations of outlier gradients are not known, [3] thresholds the curl values as a heuristic.

Relationship with ℓ_0-minimization: Note that the key idea is that for integration to be possible, the resulting graph has to be connected. Thus, the minimal set of gradients required for integration should correspond to a spanning tree (ST) [6] as shown in figure 6.2(b). First, let us assume that the graph remains connected after removing the edges corresponding to outlier gradients. Then, it is easy to see that [3] is a greedy algorithm for ℓ_0-minimization. This is because the resulting sub-graph trivially minimizes the ℓ_0-norm of gradient errors.

However, the important point is that even if we know the location of outliers, it does not guarantee error-free reconstruction, since the resulting sub-graph needs to be connected. For example, it is easy to see that if all 4 edges of a node are removed, the graph does not remain connected (figure 6.3, clique-5). On other hand, if the errors are distributed as shown in figure 6.3 (right), perfect reconstruction can be achieved. Thus, even ℓ_0-minimization does not guarantee perfect reconstruction. It can handle up to 25% outliers[1], but can fail for as low as 4 outliers. While recent work in compressed sensing [24] has focused on the *number* of errors (outliers), the location of outliers is equally important for gradient reconstruction problem. Since ℓ_0-minimization can fail depending on spatial distribution of errors, it is important to consider it while analyzing $\ell_0 - \ell_1$ equivalence.

[1]In general, ℓ_0-minimization can handle up to 50% outliers. For gradient integration, a unique solution can be obtained only for maximum of 25% outliers

RANSAC: In gradient integration, RANSAC would search over different realizations of ST and pick the one which rejects most outliers. As shown in [6], since the number of parameters are large, RANSAC is computationally prohibitive.

6.3.1 Performance under noise

Note that a robust algorithm should also be able to work well in presence of noise. In [3], a heuristic is used to estimate outlier errors, assuming that the non-zero curl values are related to outlier gradients. However, this assumption is well suited only when the gradient field is corrupted by outliers and fails in presence of noise. Under noise, the algorithm in [3] confuses correct gradients as outliers and performs poorly as shown in figure 6.5.

In presence of noise, gradient error $\mathbf{e}$ is non-sparse with the largest components corresponding to outliers. To handle noise, the cost function is modified to

$$\widehat{\mathbf{e}} = \arg\min \|\mathbf{e}\|_1 \quad \texttt{s.t.} \quad \|\mathbf{d} - \mathbf{C}\mathbf{e}\|_2 \leq \epsilon \tag{6.9}$$

for an appropriate ϵ.

6.4 $\ell_0 - \ell_1$ equivalence

One of the earliest methods in sparse signal recovery by minimizing the ℓ_1-norm is Basis Pursuit [38] but it is recently that conditions for equivalence between minimizing ℓ_0 and ℓ_1-norm have been provided in the compressed sensing literature [24, 28, 25]. In fact, the gradient error correction problem is similar to the classical error correction problem analyzed in [28], but the location of errors is

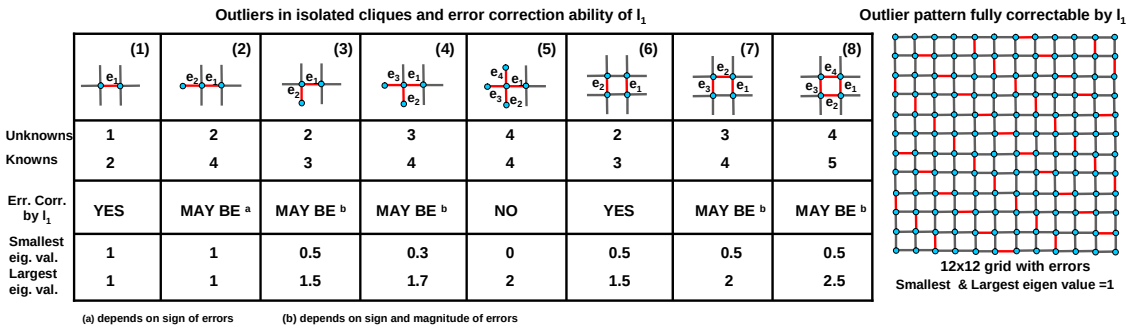

Figure 6.3: (Left) Top row: Isolated cliques (set T) in error (red). Second row: Number of unknown (gradients) and known (curls) variables. Third row: Shows whether ℓ_1-minimization can correct these errors. Fourth row: Smallest & largest eigenvalue (λ_{min} & λ_{max}) of $\mathbf{C}_T^*\mathbf{C}_T$. (Right) Distribution of outliers (red) on 12×12 grid which can be corrected perfectly by ℓ_0 and ℓ_1. Note that the errors are distributed apart and follow clique-1 structure.

equally important as discussed in section 6.3. Continuing the notation, we present sufficient conditions for $\ell_0 - \ell_1$ equivalence as described in [25]. They are

- $\mathbf{e}$ is k-sparse ($\|\mathbf{e}\|_0 = k$)
- The matrix $\mathbf{C}$ obeys RIP with isometry constant δ_{2k}

fig: Exp: RampOutliersfig: Exp: RampNoise

RIP (with δ_{2k}) is a sufficient condition on a matrix ($\mathbf{C}$) which guarantees recovery of *all* k-sparse vectors ($\mathbf{e}$) from its projection ($\mathbf{d}$) using ℓ_0-minimization (if $\delta_{2k} \leq 1$) or ℓ_1-minimization (if $\delta_{2k} < \sqrt{2}-1$). This implies that ℓ_1-minimization can recover a k-sparse vector as well as ℓ_0-minimization when $\delta_{2k} < \sqrt{2} - 1$. $\mathbf{C}$ is said to satisfy RIP with isometry constant δ_{2k}, if the eigenvalues of $\mathbf{C}_T^*\mathbf{C}_T$ [2] lie between $(1 - \delta_{2k})$ and $(1 + \delta_{2k})$ for every submatrix $\mathbf{C}_T$, formed by choosing $2k$ columns with index set T. Note that the condition to recover k-sparse $\mathbf{e}$ is actually on $2k$ columns of $\mathbf{C}$. This is to ensure that the true k-sparse vector is not confused with any other

[2] $\mathbf{C}^*$ is the transpose of $\mathbf{C}$

k-sparse vector with the same projection $\mathbf{d}$, thereby ensuring a *unique* solution. Typically, dense matrices such as *i.i.d.* Gaussian or partial Fourier matrices [24] satisfy RIP for large k.

As discussed in section 6.3, if all 4 edges of a node are in error, they can't be corrected even if we knew their locations. It implies that the recovery of 4-sparse gradient error vector $\mathbf{e}$ using either ℓ_0 or ℓ_1-minimization is impossible. Thus, RIP doesn't hold for $k = 4$ and hence for all $k > 4$. But, the constant δ_{2k} corresponding to a $2k$ edge set T does inform us whether any k gradient errors in T can be corrected using either ℓ_0 or ℓ_1-minimization

For a $2k$ edge set T, $\delta_{2k} < 1$ means that $\mathbf{C}_T^* \mathbf{C}_T$ is non-singular. This implies that the 2D graph remains connected after removing the corresponding $2k$ edges T. Conversely, in figure 6.3 clique-5, $\delta_{2k} = 1$ since the graph does not remain connected when all the four edges are in error.

6.4.1 Spatial distribution of errors

Figure 6.3 lists several spatial distribution of errors in a isolated neighborhood. We qualitatively analyze which of these can be corrected with the help of isometry constant δ_{2k}. Few of them are described in detail below. For example, in clique-2 ($2k = 2$), $\delta_{2k} = 0.5$ implying clique-1 ($k = 1$) can be corrected perfectly by ℓ_0-minimization. However, in practice ℓ_1-minimization can also correct the single outlier although $\delta_{2k} > \sqrt{2}-1$. Likewise, $\delta_{2k} = 0.5$ in clique-8 ($2k = 4$) implies clique-6 ($k = 2$) can be corrected perfectly by both ℓ_0 & ℓ_1-minimization. This confirms

that conditions on δ_{2k} are just sufficient. Nevertheless, the conditions provide insight into the error locations that can be corrected.

Since the condition for ℓ_1 recovery is stronger than ℓ_0 recovery, there exist outlier distributions which ℓ_0-minimization can correct but ℓ_1 cannot. For example, since $\delta_{2k} = 0.5$ in clique-8, ℓ_0-minimization can correct clique-3 but ℓ_1 cannot always. Conversely, if ℓ_0-minimization cannot correct a gradient error $\mathbf{e}$ then neither can ℓ_1. In other words, ℓ_1-minimization corrects less errors compared to ℓ_0.

We generalize to other outlier spatial distributions. Let T denote the indices of some $2k$ edge locations and T^c the complement edges. If T^c is not a connected subgraph, the matrix $\mathbf{C}_T^*\mathbf{C}_T$ is singular and $\delta_{2k} = 1$. This implies that there exist k error locations in T, which ℓ_0-minimization cannot correct uniquely. If T^c is a connected subgraph, then the matrix $\mathbf{C}_T^*\mathbf{C}_T$ is non-singular and $\delta_{2k} < 1$ suggesting ℓ_0-minimization can correct any k error locations in T. For sufficiently small k we will have $\delta_{2k} < \sqrt{2} - 1$ and ℓ_1-minimization corrects all of them. For example, ℓ_1-minimization can correct outliers distributed as shown in figure 6.3 (right).

6.4.2 Expander graph structure

Unlike typical dense matrices in compressed sensing, the curl matrix $\mathbf{C}$ is sparse and hence doesn't satisfy RIP for even few edges in error. Each curl value carries information about four gradients and each gradient contributes to two curl values. In the graph obtained by removing the border edges of the grid, the gradients and curl values have an *expander graph* relationship where every gradient

contributes to two curl values and every curl value has contribution from four gradients. The truncated curl matrix $\mathbf{C}_{int}$ corresponding to gradients in the interior has the structure of an adjacency matrix of an expander graph, where the gradients are the left nodes U and the curl values, the right nodes V (figure 6.2(d)). But, each column of $\mathbf{C}_{int}$ has both $+1$ and -1 entries unlike the adjacency matrix which has only $+1$ as both entries.

In the compressed sensing literature, the concept of RIP has been extended to sparse matrices such as the adjacency matrix of an expander graph [17]. Theorem 1 in [17] states that if any matrix $\mathbf{C}_{ex}$ of size $M \times N^{'}$ is the adjacency matrix of an (k, α) expander $G = (U, V, E)$ with left degree d such that $1/\alpha, d$ are smaller than $N^{'}$, then the scaled matrix $\mathbf{C}_{ex}/d^{1/p}$ satisfies the $\text{RIP}_{p,k,\delta}$ property, for $1 \leq p \leq 1 + 1/\log n$ and $\delta = \beta\alpha$ for some absolute constant $\beta > 1$.

Although $\mathbf{C}_{int}$ is not truly an adjacency matrix, it follows the proof of Theorem 1 in [17] for the case $p = 1$ in a straightforward way with parameters $d = 2$ and $\alpha \sim 3/4$. $\alpha \sim 3/4$ implies a poor expander and hence ℓ_1-minimization fails to correct the errors for even simple outlier distributions. Nevertheless, the expander graph structure of the problem provides a nice framework to analyze the error distributions which can be corrected completely (such as figure 6.3 (right)) and also opens the door for greedy algorithms which can correct such error distributions. For example, the standard decoding algorithm for expander codes with $d = 2$ and $\alpha \sim 3/4$ would first look for two neighboring curl values which have been affected by a corrupt edge and then account for that edge in the curl values and iterate this search. This procedure indicates that for a decoding algorithm to be successful on the 2D

	ℓ_1-minimization	Least Squares	Diffusion	Shapelets	Algebraic
Ramp-peaks Noise only	0.5581	0.2299	0.3980	0.7221	4.5894
Ramp-peaks Outliers only	0.3136	9.9691	2.0221	24.7759	0.2430
Ramp-peaks Noise & Outliers	0.5064	6.8096	1.8382	16.8603	3.1849
Mozart PS	550.1	575.8	521.4	1179.1	708.7

Table 6.1: MSE of reconstructed surfaces using different methods on Ramp-peaks dataset and PS experiment on Mozart dataset.

graph (poor expander), the gradient errors should be distributed apart as shown in figure 6.3 (right)).

6.5 Experiments and Results

We compare the performance of our algorithm with the the least squares [138], Shapelets [77], algebraic approach [3] and the Diffusion algorithm [6]. For shapelets, we use the default parameters (nscales=6, minradius=1, mult=2). Shapelets produce a scaled surface with unknown scale, which is fixed by setting the surface mean to the mean of the ground truth surface (for synthetic experiments). We assume Neumann boundary conditions for integration, which results in an unknown additive constant of integration. This needs to be set for meaningful comparisons among approaches for which we align the median of the reconstructed surface values[3]. Note that although mean square error (MSE) values in table 6.1 are indicative of the algorithm performance, it may not be related to the visual performance.

To solve (6.9), we use the regularized formulation: $\arg\min \mu\|\boldsymbol{e}\|_1 + 1/2\|\boldsymbol{d} -$

[3]Effective as long as 50% of the surface values remain uncorrupted after surface reconstruction.

$\boldsymbol{Ce}\|_2^2$, since faster software [75] exists for the latter. μ is the only parameter which we need to set and to enforce sparsity in the gradient error $\boldsymbol{e}$, we found that $\mu = 10^{-3}$ works over a wide range of problems and outlier distributions.

Effect of noise without outliers: First, we compare the performance of the algorithms when the the gradient field is corrupted only by noise. We added Gaussian noise with $\sigma = 10\%$ of the maximum gradient value to the gradients of **Ramp-peaks** synthetic dataset shown in figure 6.5. ℓ_1-minimization performs as well as Least squares in presence of noise in the gradient field. The algebraic method performs poorly due to the simplifying assumptions it makes about the relationship between curl and gradient error. The MSE numbers are reported in table 6.1.

Effect of outliers without noise: To analyze the effect of outliers, we added outliers to 10% of the ground truth gradient field. The outliers are salt and pepper noise with a range five times that of the original gradient field. The reconstructed surfaces are shown in figure 6.6. ℓ_1-minimization performs as well as the algebraic approach as shown in table 6.1. Note that ℓ_1-minimization corrects most of the outliers and preserves the surface edges and details. It also confines the errors locally when it fails to correct them.

We also analyze the performance of various algorithms as the percentage of outliers increase. In figure 6.4(a), we vary the percentage of outliers in the Ramp-peaks gradient field and compute the percentage of surface values in error. We declare a surface value to be in error if it deviates more than 5% from the maximum surface value. The plot shows that the algebraic approach is the most effective in correcting outliers with similar performance by ℓ_1-minimization. Note that both

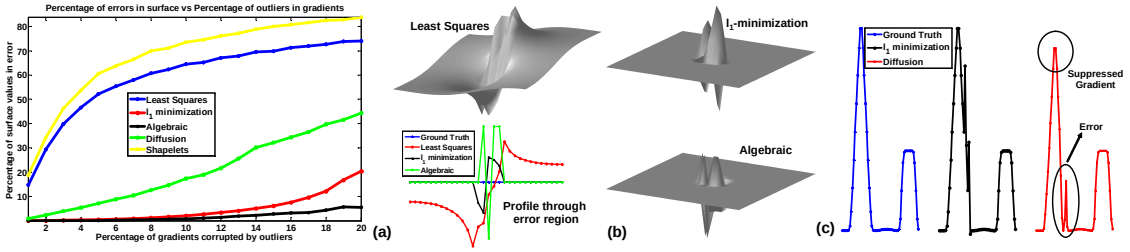

Figure 6.4: (a) Plot showing the number of surface values in error versus varying outlier percentage. (b) LEC is satisfied by the ℓ_1-minimization method and the algebraic approach but fails in the least squares. (c) In presence of outliers near a sharp edge, the Diffusion technique results in artifacts.

the least squares and Shapelets fail to preserve the surface shape even for small percentage of outliers. For this experiment, we averaged the performance over 200 realizations for every percentage of outliers.

Effect of noise and outliers: The true test of a robust algorithm is it's performance in presence of both noise and outliers. We test the realistic scenario of both noise and outliers by adding outliers to 7% of the gradients and Gaussian noise with $\sigma = 7\%$ of the maximum gradient value. ℓ_1-minimization performs better than all the other methods. It captures the characteristic of least squares to handle noise and that of a combinatorial method such as an algebraic approach to correct outliers.

Photometric stereo (PS): We perform a PS experiment on Mozart synthetic dataset to simulate the realistic occurrence of outliers in gradient fields. We first generate images assuming Lambertian reflectance model, distant point source lighting and constant albedo. Then we estimate the surface normals (n_x,n_y,n_z) and albedo through PS on images corrupted by random noise ($\sigma = 5\%$ of the maximum intensity). The estimated gradient field is given by $p = \frac{-n_x}{n_z}$ and $q = \frac{-n_y}{n_z}$ and is

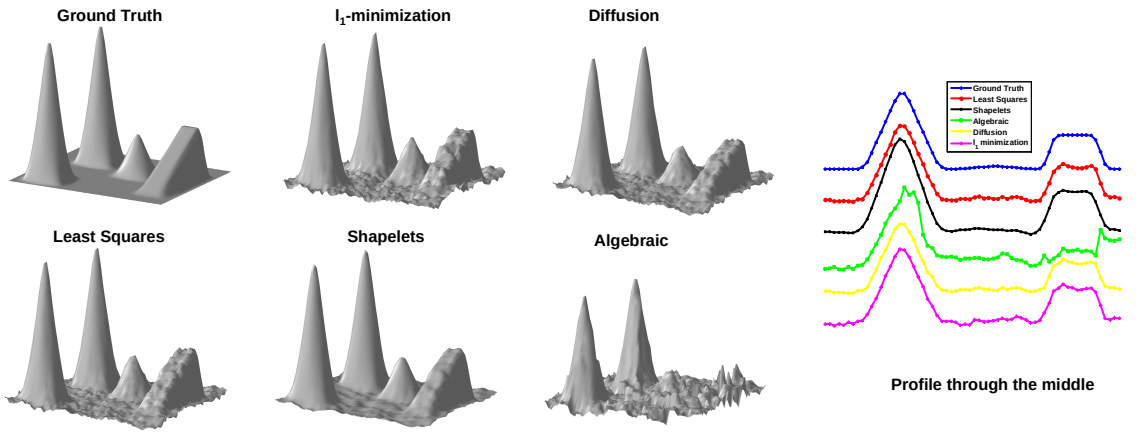

Figure 6.5: Reconstructed surface when the gradient field is corrupted by only Gaussian noise (σ=10% of the maximum gradient value). Note that the algebraic approach performs poorly under noise and ℓ_1-minimization performs as well as the least squares.

corrupted by outliers as shown in figure 6.1. Figure 6.8 shows that our method and the diffusion algorithm give the best results. Both these methods correct the outlier errors which corrupt the gradient field during gradient estimation. Although ℓ_1-minimization is marginally less successful compared to the diffusion algorithm in terms of MSE, note that our method corrects more outliers on the side of the face and also avoids the pinching artifacts near the flatter regions of the surface. It should be noted that the diffusion algorithm introduces artifacts close to sharp edges corrupted by outliers as illustrated in figure 6.4(c).

Local error confinement: We show that even when ℓ_1-minimization fails to correct the outliers, it confines the errors locally. In figure 6.4(b), we add outliers in a 5×5 region on a 20×20 flat surface. Both ℓ_1-minimization and the least squares fail to correct the errors. However, least squares method spreads the error globally, while ℓ_1-minimization confines it locally. By further regularizing the gradients themselves as in TV regularization, these remaining errors could be removed.

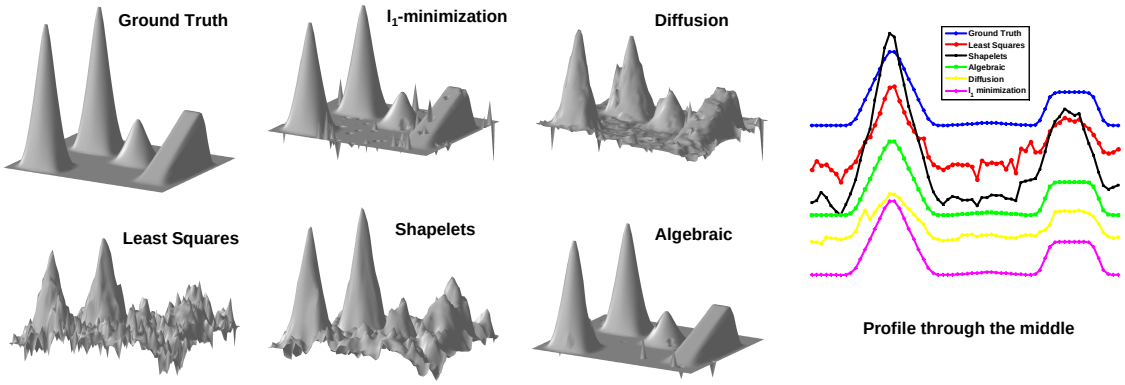

Figure 6.6: The reconstructed surface when 10% of the gradient field is corrupted by outliers with no noise. Note that because there is no noise, the algebraic approach performs best. The ℓ_1-minimization method also reconstructs with high fidelity. Other techniques perform poorly. Even when ℓ_1-minimization can't correct all the errors, it confines the errors locally and preserves sharp edges in the surface.

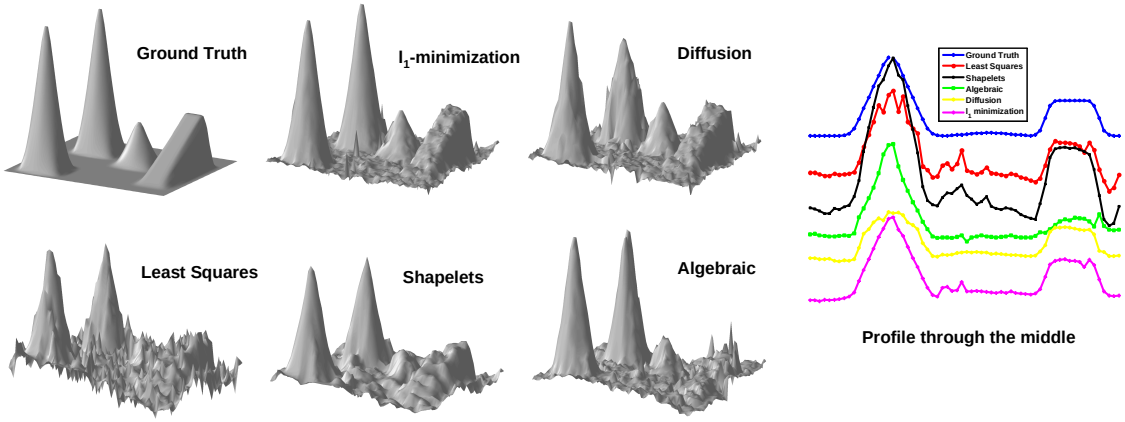

Figure 6.7: The reconstructed surface when the gradient field is corrupted by both outliers (at 7% locations) and noise (Gaussian with σ=7% the maximum gradient value). The ℓ_1-minimization method performs significantly better with the best characteristic of algebraic approach for outliers and the least squares for noise.

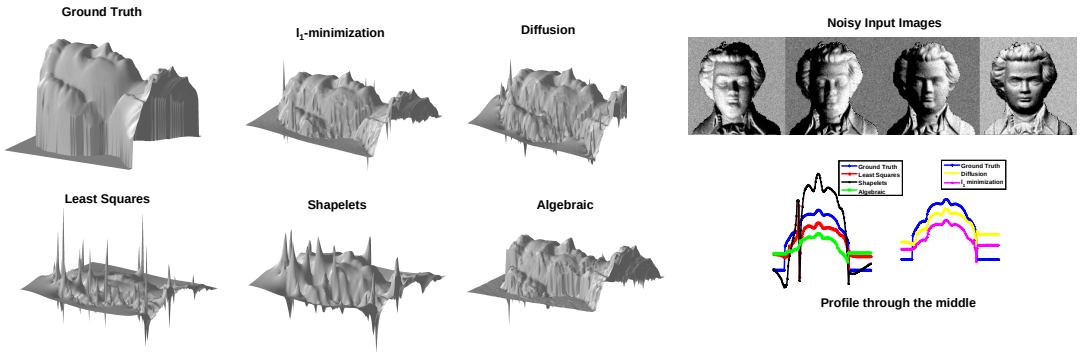

Figure 6.8: Surface of the Mozart bust reconstructed from the gradient field obtained from PS. The gradient errors are shown in figure 6.1. Both ℓ_1-minimization and diffusion perform significantly better compared to other methods but the flatter regions of the surface have artifacts in the diffusion method.

Chapter 6

Summary and Future Research Directions

In this dissertation we presented applications of sparse modeling and reconstruction for video acquisition, background subtraction, tracking and surface reconstruction. Our research illustrates that CS theory offer a distinct advantage when the redundancy underlying a problem can be expressed as a sparse combination of basis vectors. In our experience, CS theory is most applicable in low-level vision, particularly representation and acquisition of visual signals.

Sparsity is a powerful concept during signal synthesis since the idea is formulated in a linear framework and lends itself to simple expression of the notion of parsimony. On the other hand, analysis of signal is computationally expensive. Nevertheless, this limitation is being overcome through novel algorithms with relation to convex optimization and algorithms. Sparse techniques occupy an important place in the toolbox of parsimonious data modeling among manifolds, graphical models, subspace representation etc. Like other generative models for signals, sparse representation allows us to design and implement better algorithms for learning and classification. Further, the interpretation of sparse representation as a generaliza-

tion of subspace representation is important since it allows us to describe data with a much richer union-of-subspace model.

Sparse models while rich in capturing the notion of parsimony are often not sufficient to describe real data such as images, videos and plenoptic functions. Sparse modeling needs to be appropriately adapted to the problem domain. For instance, images are not sparse but compressible a model should reflect this for improved reconstruction. Similarly, videos with wide variety of motion cannot be described using an overcomplete dictionary. Richer models are needed to capture the redundancy in a spatio-temporal volume by building on the idea of sparsity.

Sparsity enforcing constraints can also be interpreted in a regularization framework (used widely in CV). Sparse regularization quantifies the underlying simplicity of data. This fact is commonly used in statistics in the form of ℓ_1 norm and other robust measures (e.g. Huber penalty). In view of increased understanding of sparse modeling, the regularization used in computer vision problems could be revisited. Also, notion of sparsity generalized to matrices as low-rank [49] would find potential use in many CV problems such as structure from motion (SfM). The missing data problem in SfM [66] could be solved using low rank modeling [102].

Specifically, this dissertation proposes following problems for further exploration.

Coded strobing photography: Coded strobing photography has many applications in monitoring industrial processes where the automation is periodic in nature. If due to faulty functioning, few of the periods of the signal are disturbed we would like our method to detect and estimate the faulty periods. This raises

an interesting question of reconstructing an anomaly in the signal which does not obey the periodicity assumption. A simpler problem to investigate would be the conditions under which faulty periods are recoverable.

Currently, the technique assumes that the camera is stationary and that the periodic phenomena is of mechanical nature such as mill-tool and toothbrush. To extend the method to repetitive biological processes such as jogging and vocal fold vibrations, the non-rigid deformation of the objects should be accounted in reconstruction. Similarly, to extend the reconstruction to a hand-held camera, the camera shake must accounted as well. Solving these problems would enable the technique to be applicable in unconstrained real world settings.

P2C2: An open question is the optimal coding scheme to be used for capturing fast processes. Would such a code be signal dependent and how should that be estimated?

Another interesting question which needs further investigation is the limits of spatio-temporal super-resolution. In our work we present temporal upsampling of upto 8 with satisfactory results. How much can this be pushed? We conjecture that temporal super-resolution is intimately linked to spatial resolution. For any scene, more the spatial resolution, the higher we will be able to super-resolve temporally. This needs to be investigated. On a related note, the detail in the spatial content would affect the temporal super-resolution and this needs to be investigated as well.

Further, the appropriate spatio-temporal representation needs further investigation. We have presented an approach of keeping the spatial and temporal redundancy distinct. An interesting question would be to combine the motion information

in time with patch similarity which has shown huge promise in video restoration techniques.

More broadly investigation of different forms of redundancy in the plenoptic function is an interesting avenue of research. Identifying the redundancy would allow us to devise novel ways of capturing and manipulating the plenoptic function. Further it would be interesting to evaluate if low-rank models can be used for the plenoptic function and how it can be exploited for capture and display [79].

Tracking: Currently, we estimate the foreground and the background in a compressed video. An interesting avenue of research is to track the foreground and use it improve the estimate of the foreground in the subsequent frames. Such a system would enable the deployment of compressive cameras in realistic scenarios.

Integrability: Currently we use an ℓ_1 minimization approach to solving for the gradient errors. It would be interesting to investigate the connection between sub-modularity and the graph structure of the gradients to develop greedy algorithms for estimating the errors.

Chapter A

Appendix

A.1 Overview of CS theory

In signal processing the idea of decomposing a signal into simple components, compressing a signal, de-noising it and solving inverse problems such as de-convolution are very common. The central underlying idea is that although data (signal) manifests itself in a certain form with a certain number of variables, it can typically be explained using a far fewer number of variables by looking at it appropriately.

Compressed sensing, with its central theme as 'sparsity', is an emerging theory which studies the established ideas in signal processing in a fundamental way. The core idea is that a signal which is sparse in an appropriate basis can be represented using far fewer variables than its original dimension and that the original signal can be recovered with the help of realistically implementable algorithms. This development has led to a burst of new research in the area of signal and image processing and related fields. The scope of this theory is vast enough to warrant a second look at the Shannon-Nyquist sampling theorem. Consequently, it's effect can be felt on most signal processing ideas. Behind this theory is sophisticated mathematics

which can provide insight into old problems. In this dissertation, we investigated few problems in imaging and computer vision where this new theory enables new algorithms to improve the state-of-the-art.

First, we provide a brief introduction to the ideas encompassed by the term 'compressed sensing'. Introduction to the field of compressed sensing can be found in [8, 31, 24, 20]. Given an under-determined system of equations

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} \tag{A.1}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{m \times n}$ is a full rank matrix with $m < n$, there are infinitely many solutions. The central question in compressed sensing is to seek the sparsest one. This raises the question of conditions under which a unique sparse solution exists and how to recover such a solution. This problem can be extended to scenarios encountered in practice where the signal $\boldsymbol{x}$ is not sparse but is 'compressible' and when the signal $\boldsymbol{y}$ is corrupted by noise. The conditions under which a unique sparse $\boldsymbol{x}$ exists has been given in [28] and [43]. The *Restricted Isometry Property*(RIP) introduced in [28] and re-proved using elementary ideas in [25] is the most popular one. Generally, it is a combinatorial problem to check if a matrix $\boldsymbol{\Phi}$ satisfies RIP. Nevertheless, it has been shown in [26] that a random Fourier ensemble satisfies such a condition with high-probability. Similarly, it has been shown in [29] using concentration of measure ideas that a random IID Gaussian matrix satisfies RIP with high probability. The formulation in (A.1) can be easily extended to cases when $\boldsymbol{x}$ is sparse in some basis $\boldsymbol{\Psi}$ i.e. $\boldsymbol{x} = \boldsymbol{\Psi}\boldsymbol{\theta}$. In such cases the condition for recovering sparse solutions would be on the matrix product $\boldsymbol{\Phi}\boldsymbol{\Psi}$.

To recover a sparse $\boldsymbol{x}$, one would need to solve a combinatorial problem which minimizes the ℓ_0 pseudo-norm.

$$(P0): \qquad min||\boldsymbol{x}||_0 \;\; s.t \;\; \boldsymbol{y} = \boldsymbol{\Phi x} \tag{A.2}$$

But this is NP-hard and infeasible to implement. Instead, the ℓ_1-norm is minimized since it is the convex equivalent closest to ℓ_0 pseudo norm.

$$(P1): \qquad min||\boldsymbol{x}||_1 \;\; s.t \;\; \boldsymbol{y} = \boldsymbol{\Phi x} \tag{A.3}$$

It has been known in machine learning research [142] that (P1) leads to sparse solutions and this fact was also used in signal processing literature in the form of *Basis Pursuit* [37]. The conditions under which (P1) gives the same unique solution as (P0) for random Fourier ensemble was given in [26] and a stronger version of RIP was shown to ensure recovery of unique solution using (P1) [28, 25]. (P1) is typically implemented in practice using convex programming.

There exists another branch of recovery algorithms which as compared to convex programming based approaches are greedy in nature. For instance, *Matching Pursuit*(MP) introduced in [92] sequentially finds the largest elements in $\boldsymbol{x}$ contributing to the observation $\boldsymbol{y}$. A significantly better versions in terms of accuracy and speed are the algorithms *Orthogonal Matching Pursuit*(OMP) [115] and CoSaMP [107]. Greedy algorithms have a significant advantage in speed over convex programming based methods but at the cost of slightly lower accuracy. Also, these algorithms need a stronger condition on RIP than (P1).

The above mentioned recovery algorithms extend naturally to scenarios where the signal $\boldsymbol{x}$ is not sparse but compressible. It has been shown in [27] that the

recovered signal $\hat{\boldsymbol{x}}$ using (P1) will be close to the sparse approximation of the original signal. The stability of the recovered signal $\hat{\boldsymbol{x}}$ under observation noise was also shown in the same paper.

Based on the above ideas, new imaging techniques have been invented. For instance the 'single pixel camera' introduced in [156] acquires images in a compressed form based on the idea that images are compressible in wavelet basis. The idea is to reconstruct the image in software from far fewer measurements than that of a conventional camera. In this dissertation, we present simple computer vision techniques such as background subtraction and tracking of moving objects in images acquired using such a compressive camera.

The fundamental ideas in compressed sensing and the mathematics behind it are relevant to problems in imaging and computer vision. For instance, the familiar problem of face recognition in computer vision when looked at from a point of view of signal processing encompasses the idea of compression where a face can be represented by it's PCA coefficients. Similarly, under mild conditions on geometry and albedo, a face belonging to a class under varying illumination can be explained using 9 parameters and a face can be affected by non-idealities such as occlusion and cast shadows [13]. Compressed sensing can help in explaining and providing new solutions. But, it must be noted that ideas from compressed sensing can be useful in few scenarios and may not be relevant in others. For instance, using the ideas of sparse representations the authors in [162] make face recognition robust to occlusions but other aspects of representing and recognizing faces remain a open problem and may or may not borrow ideas from compressed sensing.

Bibliography

[1] www.photron.com.

[2] A. Aggarwal, S. Biswas, S. Singh, S. Sural, and A. K. Majumdar. Object Tracking Using Background Subtraction and Motion Estimation in MPEG Videos. In *ACCV*, pages 121–130. Springer, 2006.

[3] A. Agrawal, R. Chellappa, and R. Raskar. An algebraic approach to surface reconstruction from gradient fields. In *Proc. Int'l Conf. Computer Vision*, pages 174–181, 2005.

[4] A. Agrawal, M. Gupta, A. Veeraraghavan, and S.G. Narasimhan. Optimal coded sampling for temporal super-resolution. In *IEEE Conference on CVPR*, pages 599–606, 2010.

[5] A. Agrawal and R. Raskar. Gradient domain manipulation techniques in vision and graphics. ICCV short course, 2007.

[6] A. K. Agrawal, R. Raskar, and R. Chellappa. What is the range of surface reconstructions from a gradient field? In *Proc. European Conf. Computer Vision*, volume 3951, pages 578–591. Springer, 2006.

[7] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(9):1167–1183, 2002.

[8] R. Baraniuk. Compressive sensing. In *Information Sciences and Systems, 2008. CISS 2008. 42nd Annual Conference on*, pages iv–v, March 2008.

[9] R. G. Baraniuk. Compressive Sensing. *Signal Processing Magazine, IEEE*, 24(4):118–121, 2007.

[10] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. Model-based compressive sensing, 2008.

[11] D. Baron, M. F. Duarte, M. B. Wakin, S. Sarvotham, and R. G. Baraniuk. Distributed compressive sensing. *CoRR*, abs/0901.3403, 2009.

[12] B. Bascle, A. Blake, and A. Zisserman. Motion deblurring and super-resolution from an image sequence. In *ECCV '96: Proceedings of the 4th European Conference on Computer Vision-Volume II*, pages 573–582, London, UK, 1996. Springer-Verlag.

[13] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, 2003.

[14] S. Belongie and J. Wills. Structure from periodic motion. In *Workshop on Spatial Coherence for Visual Motion Analysis (SCVMA)*, Prague, Czech Republic, 2004. Springer Verlag, Springer Verlag.

[15] M. Ben-Ezra and S. K. Nayar. Motion-based motion deblurring. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:689–698, 2004.

[16] M. Ben-Ezra and S.K. Nayar. Motion-based motion deblurring. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 689–698, 2004.

[17] R. Berinde, A. C. Gilbert, P. Indyk, H. J. Karloff, and M. J. Strauss. Combining geometry and combinatorics: A unified approach to sparse signal recovery. *CoRR*, abs/0804.4666, 2008.

[18] D. P. Bertsekas. A new class of incremental gradient methods for least squares problems. *SIAM J. on Optimization*, 7:913–926, April 1997.

[19] I. Bilinskis and A. K. Mikelson. *Randomized Signal Processing*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1992.

[20] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images, 2007.

[21] G. Bub, M. Tecza, M. Helmes, P. Lee, and P. Kohl. Temporal pixel multiplexing for simultaneous high-speed, high-resolution imaging. *Nature Methods*, 7(3):209–211, 2010.

[22] E. Candes. Compressive sampling. In *Proceedings of the International Congress of Mathematicians, Madrid, Spain*, volume 3, pages 1433–1452, 2006.

[23] E. Candes and J. Romberg. Sparsity and incoherence in compressive sampling. *INVERSE PROBLEMS*, 23(3):969, 2007.

[24] E. J. Candès. Compressive sampling. In *Proc. International Congress of Mathematicians*, volume 3, pages 1433–1452, Madrid, Spain, 2006.

[25] E. J. Candès. The restricted isometry property and its implications for compressed sensing. *Compte Rendus de l'Academie des Sciences*, 2008.

[26] E. J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[27] E. J. Candes, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[28] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Trans. Inform. Theory*, 51:4203–4215, Dec. 2005.

[29] E. J. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

[30] E.J. Candes and T. Tao. Decoding by linear programming. *Information Theory, IEEE Transactions on*, 51(12):4203–4215, Dec. 2005.

[31] E.J. Candes and M.B. Wakin. An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):21–30, March 2008.

[32] Emmanuel J. Candès, Xiaodong Li, Y. Ma, and J. Wright. Robust principal component analysis? *CoRR*, abs/0912.3599, 2009.

[33] V. Cevher, R. Chellappa, and J. H. McClellan. Gaussian approximations for energy-based detection and localization in sensor networks. In *IEEE Statistical Signal Processing Workshop*, Madison, WI, 26–29 August 2007.

[34] V. Cevher, C. Hegde, M. F. Duarte, and R. G. Baraniuk. Sparse signal recovery using markov random fields, 2008.

[35] V. Cevher, A. Sankaranarayanan, M. Duarte, D. Reddy, R. Baraniuk, and R. Chellappa. Compressive sensing for background subtraction. In D. Forsyth, Philip Torr, and A. Zisserman, editors, *Computer Vision ECCV 2008*, volume 5303 of *Lecture Notes in Computer Science*, pages 155–168. Springer Berlin / Heidelberg, 2008.

[36] Checkline. Industrial stroboscopes. http://www.checkline.com/stroboscopes/, 2008.

[37] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic Decomposition by Basis Pursuit. *SIAM Journal on Scientific Computing*, 20:33, 1998.

[38] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998.

[39] G. K. M. Cheung, T. Kanade, J. Y. Bouget, and M. Holler. Real time system for robust 3D voxel reconstruction of human motions. In *CVPR*, pages 714–720, 2000.

[40] S. C. S. Cheung and C. Kamath. Robust techniques for background subtraction in urban traffic video. In S. Panchanathan & B. Vasudev, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 5308 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 881–892, January 2004.

[41] M. Cossalter, G. Valenzise, M. Tagliasacchi, and S. Tubaro. Joint compressive video coding and analysis. *Multimedia, IEEE Transactions on*, 12(3):168 –183, april 2010.

[42] F. De la Torre and M.J. Black. Robust principal component analysis for computer vision. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 362 –369 vol.1, 2001.

[43] D. L. Donoho. Compressed Sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.

[44] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R.G. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):83–91, 2008.

[45] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, Ting Sun, K. F. Kelly, and R.G. Baraniuk. Single-pixel imaging via compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):83 –91, 2008.

[46] H. Edgerton. Rapatronic Photographs. http://simplethinking.com/home/rapatronic_photogr 1951-1963.

[47] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *IEEE FRAME-RATE Workshop*. Springer, 1999.

[48] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In D. Vernon, editor, *Computer Vision ECCV 2000*, volume 1843 of *Lecture Notes in Computer Science*, pages 751–767. Springer Berlin / Heidelberg, 2000.

[49] A. Eriksson and A. van den Hengel. Efficient computation of robust low-rank matrix approximations in the presence of missing data using the l¡inf¿1¡/inf¿ norm. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:771–778, 2010.

[50] R. Fattal, D. Lischinski, and M. Werman. Gradient domain high dynamic range compression. In *SIGGRAPH*, pages 249–256, 2002.

[51] R. Fergus, B. Singh, A. Hertzmann, S.T. Roweis, and W.T. Freeman. Removing camera shake from a single photograph. In *ACM SIGGRAPH 2006*.

[52] R. Fergus, A. Torralba, and W. T. Freeman. Random lens imaging. Technical report, MIT Computer Science and Artificial Intelligence Laboratory, 2006.

[53] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[54] R. T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Trans. Pattern Anal. Machine Intell.*, 10(4):439–451, 1988.

[55] W.T. Freeman, T.R. Jones, and E.C. Pasztor. Example-based super-resolution. *Computer Graphics and Applications, IEEE*, 22(2):56–65, Mar/Apr 2002.

[56] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(6):643–660, 2001.

[57] H. Greenspan, S. Peled, G. Oz, and N. Kiryati. MRI inter-slice reconstruction using super-resolution. In *MICCAI '01: Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 1204–1206, London, UK, 2001. Springer-Verlag.

[58] J. Gu, Y. Hitomi, T. Mitsunaga, and S. Nayar. Coded rolling shutter photography: Flexible space-time sampling. In *IEEE International Conference on Computational Photography*, pages 1 –8, 2010.

[59] M. Gupta, A. Agrawal, A. Veeraraghavan, and S. Narasimhan. Flexible Voxels for Motion-Aware Videography. *ECCV*, 2010.

[60] M. G. L. Gustafsson. Nonlinear structured-illumination microscopy: Wide-field fluorescence imaging with theoretically unlimited resolution. *Proceedings of the National Academy of Sciences of the United States of America*, 102(37):13081–13086, 2005.

[61] E. T. Hale, W. Yin, and Y. Zhang. A fixed-point continuation method for l1-regularized minimization with applications to compressed sensing. *CAAM TR07-07, Rice Univ.*, 2007.

[62] E. T. Hale, W Yin, and Y. Zhang. A fixed-point continuation method for ℓ_1-regularized minimization with applications to compressed sensing. Technical Report TR07-07, Rice University Department of Computational and Applied Mathematics, Houston, TX, 2007.

[63] B. K. P. Horn. Height and gradient from shading. *Int'l J. Computer Vision*, 5(1):37–75, 1990.

[64] P. J. Huber. *Robust Statistics*. John Wiley and Sons, New York, 1981.

[65] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graph. Models Image Process.*, 53(3):231–239, 1991.

[66] D. Jacobs. Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:206, 1997.

[67] N. Jacobs, S. Schuh, and R. Pless. Compressive sensing and differential image-motion estimation. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 718 –721, 2010.

[68] L. Jacques, P. Vandergheynst, A. Bibet, V. Majidzadeh, A. Schmid, and Y. Leblebici. Cmos compressed imaging by random convolution. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 1113 –1116, 2009.

[69] J. Jia, Y. Tai, T. Wu, and C. Tang. Video repairing under variable illumination using cyclic motions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(5):832–839, May 2006.

[70] S. Joo and Q. Zheng. A Temporal Variance-Based Moving Target Detector. In *Proceedings of the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2005.

[71] H. Jung and J. C. Ye. Motion estimated and compensated compressed sensing dynamic magnetic resonance imaging: What we can learn from video compression techniques, 2009.

[72] L. W. Kang and C. S. Lu. Distributed compressive video sensing. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 1169 –1172, april 2009.

[73] Q. Ke and T. Kanade. Robust l1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *Proc. Conf. Computer Vision and Pattern Recognition*, June 2005.

[74] S. M. Khan and M. Shah. A multi-view approach to tracking people in crowded scenes using a planar homography constraint. In *European Conference on Computer Vision*, volume 4, pages 133–146, 2006.

[75] Seung-Jean Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale l1-regularized least squares. *Selected Topics in Signal Processing, IEEE Journal of*, 1(4):606–617, Dec. 2007.

[76] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. In *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 126 –131 vol.1, October 1994.

[77] P. Kovesi. Shapelets correlated with surface normals produce surfaces. In *Proc. Int'l Conf. Computer Vision*, pages 994–1001, 2005.

[78] M. Lamarre and J. J. Clark. Background subtraction using competing models in the block-DCT domain. In *ICPR*, 2002.

[79] D. Lanman, M. Hirsch, Y. Kim, and R. Raskar. Content-adaptive parallax barriers for automultiscopic 3d display. In *ACM SIGGRAPH 2010 Talks*, SIGGRAPH '10, pages 54:1–54:1, New York, NY, USA, 2010. ACM.

[80] I. Laptev, S. J. Belongie, P. Perez, and J. Wills. Periodic motion detection and segmentation via approximate sequence alignment. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 816–823, Washington, DC, USA, 2005. IEEE Computer Society.

[81] H. Larsson, S. Hertegård, and B. Hammarberg. Vocal Fold Vibrations: High-Speed Imaging, Kymography, and Acoustic Analysis: A Preliminary Report. *The Laryngoscope*, 110(12):2117, 2000.

[82] A. Levin, P. Sand, T. S. Cho, F. Durand, and W.T. Freeman. Motion-invariant photography. In *ACM SIGGRAPH 2008*.

[83] A. Levin, P. Sand, T. S. Cho, F. Durand, and W.T. Freeman. Motion-invariant photography. In *Proc. ACM SIGGRAPH*, volume 8, 2008.

[84] A. Levin, A. Zomet, S. Peleg, and Y. Weiss. Seamless image stitching in the gradient domain. In *Proc. European Conf. Computer Vision*, pages 377–389. Springer-Verlag, 2004.

[85] Z. Lin and H. Y. Shum. Fundamental limits of reconstruction-based superresolution algorithms under local translation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(1):83–97, 2004.

[86] C. Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Cambridge, MA, USA, 2009.

[87] M. Lustig, D.L. Donoho, J.M. Santos, and J.M. Pauly. Compressed sensing mri. *Signal Processing Magazine, IEEE*, 25(2):72 –82, 2008.

[88] Vijay Mahadevan and Nuno Vasconcelos. Background subtraction in highly dynamic scenes. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pages 1–6, 2008.

[89] D. Mahajan, F.C. Huang, W. Matusik, R. Ramamoorthi, and P. Belhumeur. Moving gradients: a path-based method for plausible image interpolation. In *ACM SIGGRAPH 2009*.

[90] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[91] S. Mallat and S. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Trans. on Signal Processing*, 41(12):3397–3415, Dec. 1993.

[92] S.G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Processing*, 41(12):3397–3415, Dec 1993.

[93] H. Mannami, R. Sagawa, Y. Mukaigawa, T. Echigo, and Y. Yagi. Adaptive dynamic range camera with reflective liquid crystal. *J. Vis. Comun. Image Represent.*, 18:359–365, October 2007.

[94] R. F. Marcia and R. M. Willett. Compressive coded aperture video reconstruction. In *EUSIPCO 2008*, Lausanne, Switzerland, August 2008.

[95] R. F. Marcia and R. M. Willett. R.: Compressive coded aperture video reconstruction. In *In European Signal Processing Conference (EUSIPCO*, 2008.

[96] R.F. Marcia and R.M. Willett. Compressive coded aperture superresolution image reconstruction. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 31 2008.

[97] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[98] F. A. Marvasti. *Nonuniform Sampling: Theory and Practice*. Kluwer Academic/Plenum Publishers, 2001.

[99] M. Matsumoto and T. Nishimura. Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1):3–30, 1998.

[100] P. Mergell, H. Herzel, and I. R. Titze. Irregular vocal-fold vibrationHigh-speed observation and modeling. *The Journal of the Acoustical Society of America*, 108:2996, 2000.

[101] M. Mishali and Y. C. Eldar. From theory to practice: Sub-nyquist sampling of sparse wideband analog signals. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):375 –391, april 2010.

[102] K. Mitra, S. Sheorey, and R. Chellappa. Large-scale matrix factorization with missing data under additional constraints. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1651–1659. 2010.

[103] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231 – 268, 2001.

[104] MPEG-7 "Multimedia Content Description Interface" Documentation. WWW page, 1999. `http://www.darmstadt.gmd.de/mobile/MPEG7`.

[105] E. Muybridge. *Animals in Motion* . Dover Publications, 1957. First ed., Chapman and Hall London 1899.

[106] S. K. Nayar, V. Branzoi, and T. E. Boult. Programmable imaging: Towards a flexible camera. *Int'l J. Computer Vision*, 70:7–22, Oct 2006.

[107] D. Needell and J. A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples, 2008.

[108] M. A. Neifeld, A. Ashok, and Pawan K. Baheti. Task-specific information for imaging system analysis. *J. Opt. Soc. Am. A*, 24(12):B25–B41, Dec 2007.

[109] N. Oliver, B. Rosario, and A. Pentland. A Bayesian Computer Vision System for Modeling Human Interactions. In *ICVS*. Springer, 1999.

[110] N.M. Oliver, B. Rosario, and A.P. Pentland. A bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):831 –843, August 2000.

[111] A. V. Oppenheim, R. W. Schafer, and J. R. Buck. *Discrete-Time Signal Processing*. Prentice-Hall, 1999.

[112] J. Y. Park and M.B. Wakin. A multiscale framework for compressive sensing of video. In *Picture Coding Symposium, 2009. PCS 2009*, pages 1 –4, May 2009.

[113] J.Y. Park and M.B. Wakin. A multiscale framework for compressive sensing of video. In *IEEE Picture Coding Symposium, 2009*, pages 1–4.

[114] V.M. Patel, G.R. Easley, D.M. Healy, and R. Chellappa. Compressed synthetic aperture radar. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):244 –254, april 2010.

[115] Y. C. Pati, R. Rezaiifar, and P.S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44 vol.1, Nov 1993.

[116] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, 2003.

[117] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 12, pages 629–639, 1990.

[118] N. Petrovic, I. Cohen, B. J. Frey, R. Koetter, and T. S. Huang. Enforcing integrability for surface reconstruction algorithms using belief propagation in graphical models. *Proc. Conf. Computer Vision and Pattern Recognition*, 1:743, 2001.

[119] M. Piccardi. Background subtraction techniques: a review. In *2004 IEEE International Conference on Systems, Man and Cybernetics*, volume 4, 2004.

[120] M. Piccardi. Background subtraction techniques: a review. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4, pages 3099 – 3104 vol.4, 2004.

[121] Pointgrey Research. "PGR IEEE-1394 Digital Camera Register Reference". 2006.

[122] H. V. Poor. *An Introduction to Signal Detection and Estimation*. Springer-Verlag, 1988.

[123] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: motion deblurring using fluttered shutter. In *ACM SIGGRAPH 2006*.

[124] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: motion deblurring using fluttered shutter. *ACM Trans. Graph.*, 25(3):795–804, 2006.

[125] D. Reddy, A. Veeraraghavan, and R. Chellappa. P2c2: Programmable pixel compressive camera for high speed imaging. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pages 329–336, 2011.

[126] S. Ri, M. Fujigaki, T. Matui, and Y. Morimoto. Accurate pixel-to-pixel correspondence adjustment in a digital micromirror device camera by using the phase-shifting moiré method. *Applied optics*, 45(27):6940–6946, 2006.

[127] S. Ri, Y. Matsunaga, M. Fujigaki, T. Matui, and Y. Morimoto. Development of DMD reflection-type CCD camera for phase analysis and shape measurement. In *Proceedings of SPIE*, volume 6049, 2005.

[128] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992.

[129] A. Sankaranarayanan, P. Turaga, R. Baraniuk, and R. Chellappa. Compressive Acquisition of Dynamic Scenes. *ECCV*, pages 129–142, 2010.

[130] A. Sankaranarayanan, P. Turaga, R. Baraniuk, and R. Chellappa. Compressive acquisition of dynamic scenes. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision ECCV 2010*, volume 6311 of *Lecture Notes in Computer Science*, pages 129–142. Springer Berlin / Heidelberg, 2010.

[131] G. Schade, M. Hess, F. Mller, T. Kirchhoff, M. Ludwigs, R. Hillman, and J. Kobler. [physical and technical elements of short-interval, color-filtered double strobe flash-stroboscopy]. *HNO*, 50(12):1079–1083, Dec 2002.

[132] S.M. Seitz and C.R. Dyer. View-Invariant Analysis of Cyclic Motion. *International Journal of Computer Vision*, 25(3):231–251, 1997.

[133] Q. Shan, J. Jia, A. Agarwala, et al. High-quality motion deblurring from a single image. *ACM SIGGRAPH 2008*.

[134] Q. Shan, Z. Li, J. Jia, and C.K. Tang. Fast image/video upsampling. In *ACM SIGGRAPH Asia 2008*.

[135] H. S. Shaw and D. D. Deliyski. Mucosal wave: A normophonic study across visualization techniques. *Journal of Voice*, 22(1):23–33, January 2008.

[136] E. Shechtman, Y. Caspi, and M. Irani. Space-time super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:531–545, 2005.

[137] E. Shechtman, Yaron Caspi, and M. Irani. Increasing space-time resolution in video. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 753–768, London, UK, 2002. Springer-Verlag.

[138] T. Simchony, R. Chellappa, and M. Shao. Direct analytical methods for solving poisson equations in computer vision problems. *IEEE Trans. Pattern Anal. Machine Intell.*, 12(5):435–446, 1990.

[139] V Stankovic, L Stankovic, and S Cheng. *Compressive video sampling*, pages 2–6. Number Eusipco. IEEE, 2009.

[140] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, 1999.

[141] C. Theobalt, I. Albrecht, J. Haber, M. Magnor, and H. Seidel. Pitching a baseball: tracking high-speed motion with multi-exposure images. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 540–547, New York, NY, USA, 2004. ACM.

[142] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[143] B. U. Töreyin, A. E. Çetin, A. Aksay, and M. B. Akhan. Moving object detection in wavelet compressed video. *Signal Processing: Image Communication*, 20(3):255–264, 2005.

[144] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk. Beyond nyquist: Efficient sampling of sparse bandlimited signals, 2009.

[145] P. Turaga, R. Chellappa, V.S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1473 –1488, nov. 2008.

[146] S. Uttam, N. A. Goodman, and M. A. Neifeld. Direct reconstruction of difference images from optimal spatial-domain projections. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 7096 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, August 2008.

[147] E. van den Berg and M. P. Friedlander. SPGL1: A solver for large-scale sparse reconstruction, June 2007. http://www.cs.ubc.ca/labs/scl/spgl1.

[148] E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2008.

[149] H. L. Van Trees. *Detection, Estimation, and Modulation Theory, Part I.* John Wiley & Sons, Inc., 1968.

[150] N. Vaswani. Kalman filtered compressed sensing. In *IEEE International Conference on Image Processing*, pages 893–896, 2008.

[151] N. Vaswani. Ls-cs-residual (ls-cs): Compressive sensing on least squares residual. *Signal Processing, IEEE Transactions on*, 58(8):4108 –4120, 2010.

[152] N. Vaswani and Wei Lu. Modified-cs: Modifying compressive sensing for problems with partially known support. *Signal Processing, IEEE Transactions on*, 58(9):4595 –4607, 2010.

[153] A. Veeraraghavan, D. Reddy, and R. Raskar. Coded strobing photography: Compressive sensing of high speed periodic videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):671 –686, 2011.

[154] A. Veeraraghavan, D. Reddy, and R. Raskar. Coded strobing photography: Compressive sensing of high speed periodic videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):671 –686, 2011.

[155] A. Wagadarikar, R. John, R. Willett, and D. Brady. Single disperser design for coded aperture snapshot spectral imaging. *Appl. Opt.*, 47(10):B44–B51, Apr 2008.

[156] 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 *ICIP*, pages 1273–1276, Atlanta, GA, Oct. 2006.

[157] B. Wandell, P. Catrysse, J. DiCarlo, D. Yang, and A. El Gamal. Multiple capture single image architecture with a cmos sensor. In *the International Symposium on Multispectral Imaging and Color Reproduction for Digital Archives*, pages 11–17. Society of Multispectral Imaging of Japan, 1999.

[158] W. Wang, D. Chen, W. Gao, and J. Yang. Modeling background from compressed video. In *IEEE Int. Workshop on VSPE of TS*, pages 161–168, 2005.

[159] B. Wilburn, N. Joshi, V. Vaish, E. V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 765–776, New York, NY, USA, 2005. ACM.

[160] B. Wilburn, N. Joshi, V. Vaish, E.V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. *ACM SIGGRAPH 2005*.

[161] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *OptEng*, 19(1):139–144, 1980.

[162] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(2):210–227, 2009.

[163] F. Yasuma, T. Mitsunaga, D. Iso, and S.K. Nayar. Generalized Assorted Pixel Camera: Postcapture Control of Resolution, Dynamic Range, and Spectrum. *IEEE Transactions on Image Processing*, 19(9):2241–2253, 2010.

[164] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38, December 2006.

[165] J. Zheng and E. L. Jacobs. Video compressive sensing using spatial domain sparsity. *Optical Engineering*, 48(8):087006–+, August 2009.