University of Alberta

Maximum Frame Rate Video Acquisition Using Adaptive Compressed Sensing

by

Zhaorui Liu

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Digital Signals and Image Processing

Department of Electrical and Computer Engineering

Spring 2011
Edmonton, Alberta

Examining Committee

Dr. H. Vicky Zhao, Electrical and Computer Engineering

Dr. Mrinal Mandal, Electrical and Computer Engineering

Dr. Bin Han, Mathematical and Statistical Sciences

Abstract

Compressed sensing is a novel technology to acquire and reconstruct sparse signals below the Nyquist rate. This thesis explores the temporal redundancy in videos, and proposes a block-based adaptive framework for compressed video sampling. The proposed framework classifies blocks into different types depending on temporal correlation, and adjusts the sampling and reconstruction strategy accordingly. This framework also considers the texture complexity of regions, and adaptively adjusts the number of measurements collected. A frame rate selection module is included to select the maximum achievable frame rate under the hardware sampling rate and the perceptual quality constraints. Simulation results show that compared to the raster scan, the proposed framework can increase the frame rate by up to six times depending on the scene and the video quality constraints. A $1.5 \sim 7.8$dB gain in the average PSNR of the reconstructed frames is observed when compared with prior works on compressed video sensing.

Acknowledgements

First of all, I would like to express my sincere thanks to my supervisor Dr. H. Vicky Zhao for offering me this valuable opportunity to develop research skills and many other abilities. All the advices, encouragement and support she provided are really important for the completion of this work, and are very helpful for me to stay positive when challenges arise during research.

I also want to thank all the members working together on the Terahertz imaging project, especially Dr. A. Y. Elezzabi, for all valuable discussions. It has been a truly enlightening experience to work with so many intelligent people.

I am grateful for all the lab members and friends. Their help in my study and daily life makes these two years more unforgettable.

I deeply appreciate the continuous support of my family. Their love and understanding contribute to my every step forward.

Last but not least, I gratefully acknowledge the financial support of Terahertz imaging group, the Department of Electrical and Computer Engineering at University of Alberta, and my supervisor.

Contents

List of Tables

List of Figures

Acronyms

Acronyms	**Definition**
CCD	charge-coupled device
DCT	discrete cosine transform
DWT	discrete wavelet transform
A/D	analog-to-digital
3-D	three dimensional
DMD	digital micromirror device
SBHE	scrambled block hadamard ensemble
CS	compressed sensing
SFE	scrambled fourier ensemble
OMP	orthogonal matching pursuit
CoSaMP	compressive sampling matching pursuit
GPSR	gradient projection for sparse reconstruction
RIP	restricted isometry property
min-TV	total variation minimization
MRI	magnetic resonance imaging
NSCT	nonsubsampled contourlet transform
CT	computed tomography
MARX	model-based adaptive recovery of compressive sensing
PAR	piecewise autoregressive

PSNR	peak signal-to-noise ratio
MSE	mean square error
CIF	common intermediate format
fps	frames per second
mps	measurements per second
GOP	group of pictures

List of Symbols

Symbol	Definition
n	block size
Φ	measurement matrix
Ψ	transform matrix
α	the transform coefficients
K_0	the number of measurements assigned for each reference block
C_{fr}	frame rate enhancement ratio
f_{cs}	frame rate of the adaptive compressed sensing scheme
f_{rs}	frame rate of the traditional raster scan
$\mathbf{x}_t^k$	block k in the frame t
$\mathbf{x}_{t,t-1}^k$	the block difference between two co-located blocks
$\mathbf{y}_{M_0,t}^k$	the partial measurement vector of block $\mathbf{x}_t^k$
Φ_{M_0}	matrix containing the first M_0 rows of Φ
$\mathbf{y}_d^k$	the difference of partial measurements
T_1	the lower threshold for block classification
T_2	the higher threshold for block classification
M_0	the number of measurements for each static block
M_1	the number of measurements for each small-change block
M_2	the number of measurements required for a large-change block
$\tilde{S}$	the estimated block sparsity

l	the threshold used to estimate block sparsity
ε	the MSE between the reconstructed and the original blocks
M_{lb}	the theoretical lower bound for M_2
C	coefficient for the computation of $\underline{M}$
R	the l_1 norm of the transform coefficients α
$\mu(\phi, \psi)$	coherence between the sensing matrix ϕ and the sparsity basis ψ
i_t	the nearest indicator frame for frame t
M_{max}	the maximum # of measurements the can be acquired per second
M_f	the number of measurements for each non-reference frame
M_{f2}	the number of measurements for all large-change blocks in a frame
N_0	the total number of static blocks in a frame
N_1	the total number of small-change blocks in a frame
N_2	the total number of large-change blocks in a frame
λ	Lagrange multiplier
$\hat{M}_2^i$	the optimal number of measurements for large-change block i
$\hat{M}_2^l$	the smallest # of measurements assigned for a block within the frame
$\hat{M}_2^h$	the largest # of measurements assigned for a block within the frame
M_e	the number of extra measurements
f_j	the level-j frame rate
f_{lcm}	the least common multiple of the candidate frame rates
$\mathbf{Y}_{t,\ lcm}$	the collected measurements for frame t sampled at frame rate f_{lcm}
$\mathbf{Y}_{t,j}$	the collected measurements for frame t sampled at frame rate f_j
$M_{sec,j}$	the required number of measurements per second at frame rate f_j
b	the number of bits used to quantize measurement vectors

Chapter 1

Introduction

1.1 Motivation

With the rapid advance of multimedia technologies, digital videos are widely used in commercial and professional applications, for example, digital television, video surveillance, medical diagnosis, etc. A video signal is a sequence of two-dimensional (2-D) images. The spatial resolution of a video sequence is limited by the number of sensors in the video camera used to capture the video sequence. Most current video cameras use charge-coupled device (CCD), which is a very small solid-state silicon chip. The chip contains thousands or even millions of light-sensing sensors that are arranged in horizontal rows and vertical stacks, and each corresponds to one pixel. When a CCD chip contains more sensors, the captured resulting screen image is sharper and has higher spatial resolution [2]. The temporal resolution of a video is measured by its frame rate, which is defined as the number of frames per second. A high frame rate is often desired to avoid obvious flicker effects, especially for fast moving scenes.

Traditionally, a video camera uniformly samples the scene in each dimension according to the desired spatial and temporal resolutions. Videos are captured in a frame-by-frame manner, and neighboring frames are separated by a fixed time interval. For each snapshot of frame, the optical signal reaching a sensor in the CCD is converted to an electronic signal. The light intensity values captured for the current frame are first stored in a buffer, and are then read out sequentially in a raster scan order [3].

With the uniform sampling scheme, to obtain a video with high spatial and temporal resolutions, a large quantity of pixel values need to be collected. To facilitate efficient storage/transmission of a video, compression algorithms are usually applied to convert the raw (uncompressed) data into a relatively small bit stream without significant quality degradation. Video compression algorithms [4, 5] are extensively studied in the literature, and they explore the spatial and the temporal redundancy in videos to reduce the file size. Each frame in the sequence consists of separate areas indicating the object surfaces. Neighboring pixels in such areas are likely to have the same or similar values, which is called the spatial redundancy [6] and can be removed using 2D decorrelating transforms, e.g., Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT). In addition, adjacent frames have similar contents and are highly correlated, and this temporal redundancy can be removed by motion estimation and compensation. During the video compression process, among all the data acquired at the sampling stage, only a small portion representing the most important information are preserved, while most of them are discarded to remove redundancy and to reduce the file size.

Even though the video acquisition and transmission/storage scheme described above is still widely used nowadays, it has two main drawbacks:

- In traditional video acquisition, higher spatial resolution requires denser sampling, which requires a large number of sensors. For example, most current consumer digital cameras are in the megapixel range. However, in some applications such as terahertz imaging and high sensitivity cooled infrared imaging, large sensor arrays cannot be incorporated or are too expensive to implement.

- Traditional video acquisition requires enormous data collection to obtain a high quality video. Compression is then applied to remove the spatial and temporal redundancy and to reduce data size. Since the sampling and the compression modules are designed independently, the spatial and temporal redundancy of a video are not considered at the sampling stage, which wastes a lot of valuable resources.

To transcend hardware limitations and to avoid large data sets, compressed sensing [7–10] is proposed as an innovative concept in signal processing, and provides a new way to collect data under the Nyquist rate. Compressed sensing combines the data sampling and the compression stages into a single linear measurement process, and can directly acquire signals in a compressed form if they are sparse or compressible in certain transform domains. By sparse, it means that in certain transform domain, the signal has only a few non-zero coefficients. By compressible, it means that even if the signal is not sparse in the strict sense, the sorted magnitudes of the signal coefficients decay quickly and most of the coefficients can be discarded without introducing much information loss. Compressed sensing theory states that, a sparse or compressible signal can be recovered exactly or approximately from a small number of measurements.

For video acquisition, the benefits of compressed sensing technique are manifold.

- First of all, it helps transcend the limitations of hardware when large sensor arrays are not feasible. For example, with the single pixel camera [11], the whole scene can be acquired with only one sensor. Similarly, the works in [12, 13] proposed a terahertz imaging system that uses a single pixel detector to enable high-speed image acquisition.

- Second, it improves the sampling efficiency. Traditional sampling adopts raster scan that collects a large amount of raw data, which wastes a lot of resources. Also, in some applications such as X-ray imaging, over-sampling may have negative impact on the object being imaged [14]. On the contrary, compressed sensing technique directly collects compressed data and, therefore, significantly reduces the number of samples to collect and shortens the sampling time. Note that super resolution technique [15,16] is another option to reduce the number of sensors and the number of measurements collected. For example, the work in [17] applied super resolution technique to enhance the low quality image acquired with a few measurements collected using uniform sampling. However, super resolution algorithms are usually based on some specific assumptions, which may not always hold. For example, in learning-based super resolution [18], the images to be processed are assumed to belong to a certain class, e.g., face or text images, and in multi-frame super resolution [19], the motion model is usually assumed as translation only.

- Third, the compressed sensing procedure has an error-correction effect, and the compressed measurement stream is robust against packet loss [20]. This is because all measurements are of equal priority and none of them are more important than others. Therefore, losing one or a few measurements does not corrupt the entire reconstruction.

Being highly compressible, video sequences are good candidates for compressed sensing applications and, therefore, based on compressed sensing theory, more efficient video acquisition schemes can be designed.

1.2 Challenges in Compressed Video Sensing

In the literature, some compressed video sensing frameworks have been proposed. However, there are still some challenging issues unsolved.

Compressed sensing can take advantage of the compressibility of a signal to reduce the number of measurements collected at the sampling stage. The compressibility of videos comes from the spatial redundancy and temporal redundancy. Compressed sensing for 2-D image acquisition and reconstruction has been well studied in the literature [11, 21–24], and can successfully explore the spatial redundancy. By considering each frame in the video sequence independently, the compressed imaging techniques can be easily extended to 3-D videos [11]. However, this simple extension is essentially a 2-D method and fails to address the temporal redundancy in videos. Therefore, how to effectively explore the temporal correlation between neighboring frames is a critical issue in compressed video acquisition.

Also, as pointed out in [14], in compressed sensing, the same sampling scheme can be performed on all signals. This non-adaptive, signal-independent sampling scheme is generally considered as a strength of compressed sensing, and has been adopted by most existing compressed sensing frameworks. However, the signal sampling and reconstruction process cannot avoid the issue of signal adaptation. This is because different regions in a video sequence have different texture complexity or temporal redundancy. For example, in a video, there are smooth regions as well as texture-rich regions, and there are static background as well as moving objects. Therefore, to obtain satisfactory overall quality, adaptive sampling strategies should be applied to different regions to address the heterogeneity of different regions. In addition, for compressed sensing hardware such as single pixel camera, there is a constraint on the sampling rate, which means that only a limited number of measurements can be collected per second. For different sequences that exhibit diverse features, different frame rates should be adopted respectively during the video acquisition to ensure the output quality of each collected frame. Thus, it is also desirable to make compressed video acquisition adaptive to the hardware sampling rate constraint as well as the perceptual quality constraint.

Furthermore, to ensure that the video is acquired with a fixed time interval between neighboring frames and can be played at a constant speed, all frames should be assigned equal numbers of measurements. However, regions with different features are not evenly distributed among all frames. Therefore, another challenging issue is, given a fixed number of measurements for each frame, how to properly allocate them within each frame to achieve the best overall quality.

1.3 Thesis Outline and Contributions

This thesis takes into consideration the fact that different regions in a video sequence have different temporal redundancy and texture complexity. It addresses the challenging issues in compressed sensing for video, and proposes a block-based adaptive sampling scheme under the hardware sampling rate and the perceptual quality constraints. Specifically, the novel features of this work include:

- A block analysis module is used to classify blocks into different types and to explore temporal correlation between neighboring frames. Based on the block types and the texture complexity of the scene, adaptive compressed sensing is applied to adjust the video acquisition and reconstruction process for maximum sampling efficiency.

- An intra-frame measurement allocation algorithm for adaptive video compressed sensing is developed to achieve the best quality and to support a constant play speed of the reconstructed video.

- A frame rate selection algorithm is proposed to select the maximum achievable frame rate under the hardware sampling rate and the quality constraints.

The rest of this thesis is organized as follows. Chapter 2 introduces the basic compressed sensing theory and reviews related works of compressed sensing for images and videos. Chapter 3 discusses the proposed adaptive video acquisition framework based on compressed sensing. Chapter 4 gives a thorough performance analysis, and shows the simulation results. Finally, Chapter 5 draws conclusions and discusses future work.

Chapter 2

Literature Review

2.1 Compressed Sensing

A signal $\mathbf{x} \in \mathscr{R}^N$ can be sparsely represented under some basis $\Psi_{N\times N} = [\psi_1, \psi_2, \ldots, \psi_N]$ where ψ_i represents the ith basic column vector in the sparse basis matrix Ψ, if $\mathbf{x} = \Psi\alpha$ and the transform coefficients $\alpha \in \mathscr{R}^N$ has only $S \ll N$ non-zero elements. For example, images can be considered as approximately sparse signals in the DCT or DWT domain, and Ψ is the corresponding DCT or DWT transform matrix. Given a signal $\mathbf{x}$ that can be sparsely represented under the basis Ψ, compressed sensing explores the sparsity of the signal and takes only $M \ll N$ of measurements during the sampling process. The sampling process can be expressed as

$$\mathbf{y} = \Phi\mathbf{x} = \Phi\Psi\alpha, \tag{2.1}$$

where Φ is an $M \times N$ measurement matrix, and $\mathbf{y} \in \mathscr{R}^M$ is the resulting measurement vector. Desirable properties of the measurement matrix Φ are [25]:

- Near optimal performance: the number of measurements required for exact reconstruction is close to the theoretical bound;
- Universality: Φ should be incoherent with a variety of sparse basis matrices

Ψ.

Definition 1 *[26] The mutual coherence of the N-dimensional orthonormal bases Φ and Ψ is the maximum absolute value for the inner product between elements of the two bases:*

$$\mu(\Phi, \Psi) = \max_{1 \leq i,j \leq N} \mid \langle \phi_i, \psi_j \rangle \mid . \tag{2.2}$$

The mutual coherence measures the largest correlation between any two elements of Φ and Ψ. Compressed sensing prefers low coherence pairs [27], because fewer measurements are required when the coherence is smaller;

- Practical application: Φ should be fast to compute and memory efficient; and
- Hardware friendly: Φ should be easy to implement on sensing devices, such as binary matrices.

In the literature, matrices with random appearance such as Noiselets [28], Scrambled Fourier Ensemble (SFE) [29], and Scrambled Block Hadamard Ensemble (SBHE) [25] have shown to be good choices for the measurement matrix Φ.

Since $M \ll N$, compressed sampling becomes a dimension reduction process, which helps reduce the number of collected data from N to M. However, it also makes the recovery of the signal $\mathbf{x}$ from the measurements $\mathbf{y}$ an ill-posed problem, because given $\mathbf{y}$ there are infinitely many $\mathbf{x}'$ such that $\Phi\mathbf{x}' = \mathbf{y}$. To ensure a good estimate of α from the compressed measurements, the matrix $A = \Phi\Psi$ should satisfy the restricted isometry property (RIP) [30].

Definition 2 *For each integer $S = 1, 2, \ldots$, define the isometry constant θ_S of a matrix A as the smallest number such that*

$$(1 - \theta_S) \parallel \alpha \parallel_2^2 \leq \parallel A\alpha \parallel_2^2 \leq (1 + \theta_S) \parallel \alpha \parallel_2^2 \tag{2.3}$$

holds for all S-sparse vectors α that only have S non-zero entries. The matrix A is said to obey the RIP of order S if θ_S is not too close to one [27].

When this property is satisfied, all subsets of S columns taken from A are nearly orthogonal, and all pairwise distances between S-sparse signals can be well preserved in the measurements space. That is, $(1-\theta_{2S}) \parallel \alpha_1 - \alpha_2 \parallel_2^2 \leq \parallel A\alpha_1 - A\alpha_2 \parallel_2^2 \leq (1+\theta_{2S}) \parallel \alpha_1 - \alpha_2 \parallel_2^2$ holds for all S-sparse vectors α_1, α_2. This fact is demonstrated to guarantee the existence of efficient algorithms for reconstruction of sparse signals [27]. For measurement matrices Φ with random appearance, given any orthonormal basis Ψ, RIP holds with high probability.

The reconstruction can be formulated as an l_1 minimization problem [7] by solving

$$\hat{\alpha} = \arg\min \parallel \alpha \parallel_1 \qquad s.t. \;\; \Phi\Psi\alpha = \mathbf{y}, \tag{2.4}$$

where $\parallel \alpha \parallel_1$ is the l_1 norm of α. To solve the above optimization problem, many techniques have been proposed in the literature. Orthogonal matching pursuit (OMP) [31, 32] is a typical approach for sparse signal reconstruction, and recently, a regularized OMP algorithm [33] is proposed to achieve fast and stable reconstruction of sparse signals. There are also other options such as Bregman iterative algorithm [34], iterative hard thresholding algorithm [35], and compressive sampling matching pursuit (CoSaMP) [36], etc. The gradient projection for sparse reconstruction (GPSR) [37] is one of the most efficient and fast algorithms that significantly reduce the computation cost. Given the solution to (2.4), the original signal is reconstructed as $\hat{\mathbf{x}} = \Psi\hat{\alpha}$.

If the signal is an image, there is an alternative way for reconstruction. For an image $\mathbf{X}$ of size $n_r \times n_c$, let $\mathbf{x} \in \mathscr{R}^{n_r \times n_c}$ be its vector representation. Given the compressed measurements $\mathbf{y}$, the image can be recovered by minimizing the total variation (min-TV):

$$\hat{\mathbf{x}} = \arg\min \parallel \mathbf{x} \parallel_{TV} \qquad s.t. \; \Phi\mathbf{x} = \mathbf{y}, \tag{2.5}$$

where $\| \mathbf{x} \|_{TV} = \sum_{i,j} \sqrt{(X_{i+1,j} - X_{i,j})^2 + (X_{i,j+1} - X_{i,j})^2}$ is the approximated l_1 norm of the image gradient [29, 38]. Compared with (2.4), it has been shown in [38, 39] that minimizing the total variation helps reconstruct images of better quality at a cost of higher computation complexity.

2.2 Compressed Sensing for Images

2.2.1 Single-pixel Camera

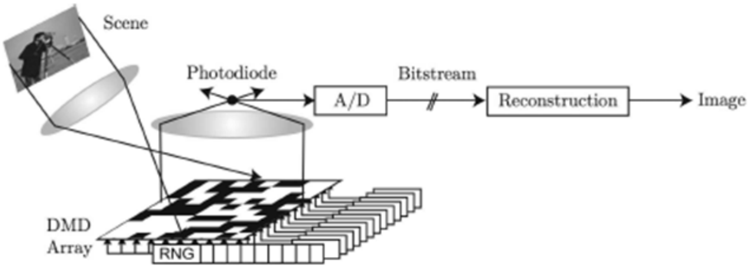


Fig. 2.1. The single-pixel camera diagram [1].

In [1], a prototype imaging system that employs compressed sensing principles has been proposed. The hardware realization is a *single-pixel* camera, which consists of a digital micromirror device (DMD), two lenses, a single photon detector, and an analog-to-digital (A/D) converter, as shown in Figure 2.1. During the compressed imaging, the light-field from the scene is focused onto the DMD. The DMD is made of an array of micromirrors, where each mirror corresponds to a particular pixel in the image $\mathbf{x}$ and one column in the measurement matrix Φ. Every setting of the micromirror array corresponds to one row in the measurement matrix Φ to collect a single compressed measurement. For example, to collect the mth measurement $\mathbf{y}(m)$, the orientation of each mirror in the DMD should be set according to ϕ_m, which is the mth row of the measurement matrix Φ. A mirror can be independently oriented either toward the lens (corresponding to a “1” at that pixel in ϕ_m) or away from the lens (corresponding to a “0” at that pixel in ϕ_m). The reflected light is then collected by the lens and focused onto a single photon detector to compute the measurement $\mathbf{y}(m) = \langle \mathbf{x}, \phi_m \rangle$ as its output. Then, it is digitized by an A/D converter. This process is repeated M times to obtain the measurement vector $\mathbf{y}$.

2.2.2 Compressed Imaging Applications

One successful application of compressed imaging technique is the compressed sensing magnetic resonance imaging (MRI). MRI is a medical imaging tool with an inherently slow data acquisition process. Its sensing speed is usually limited by the large number of samples needed along the phase encoding direction. Applying compressed sensing to MRI can significantly reduce the number of required data and shorten the scan time. MRI meets two key requirements for successful application of compressed sensing [40]: 1) medical images are naturally compressible by sparse coding in an appropriate transform domain, and 2) MRI scanners naturally acquire encoded samples, rather than direct pixel samples (e.g., in spatial-frequency encoding).

Typical sparse transform used for compressed sensing MRI is wavelet. However, wavelet will not preserve the smoothness along contours and separable wavelets can only capture limited directional information [41]. To overcome this problem, contourlet is introduced to compressed sensing for MRI, and a redundant form of contourlet, nonsubsampled contourlet transform (NSCT) [42], was employed in [43]. NSCT can suppress aliasing and improve the visual appearance of magnetic resonance images. Another improvement in the selection of the sparsity basis Ψ for MRI is the use of combined sparsifying transform. Most current compressed sensing MRI techniques only enforce the sparsity of images in a single transform, e.g. wavelet and contourlet. A single sparsifying transform limits the reconstruction quality because it cannot sparsely represent all types of image features. To improve the performance, the work in [44] proposed a new framework for compressed sensing MRI that combines different sparsifying transforms to efficiently represent different image features. Reconstruction algorithms for MRI are also studied in the literature to improve the reconstruction quality. For example, the work in [45] extended the recent Fourier-based algorithms for convex optimization to the non-convex setting. It proposed a reconstruction method that reconstructs good-quality MR images similar to those non-convex approaches [46, 47], and it also has low computational complexity that is comparable to the state-of-the-art convex methods [37, 48]. In [49], a reconstruction algorithm that uses a joint total variation and l_1 minimization model was proposed for compressed MR imaging. The algorithm gives a faithful recovery of the MR image from a small number of measurements.

Compressed sensing has also been applied to computed tomography (CT) imaging [50, 51] and terahertz imaging [12, 13] to reduce the number of samples to collect.

2.3 Compressed Sensing for Videos

Applying compressed sensing technique to video aims to explore the spatial and temporal redundancy in videos to develop efficient sampling schemes. A straightforward solution for compressed video sampling is to apply 2-D compressed sensing to each frame independently, as shown in [11]. However, this simple extension fails to explore the inter-frame correlations in the video sequence. Therefore, a challenging issue in video compressed sensing is how to explore the temporal redundancy to further improve the sampling efficiency. To address this issue, different methods have been developed, which will be discussed in the following. Also, some works have taken the different texture features of videos into consideration and developed adaptive sampling schemes.

2.3.1 3-D Wavelet Methods

To explore the inter-frame correlation, in the literature, one approach was to consider a series of consecutive frames as a 3-D signal and apply the typical 3-D wavelet transform [52] to jointly explore the sparsity in the spatial and the temporal domains.

In [53], the single pixel camera was applied to video acquisition. A pseudorandom binary matrix is used as the sensing matrix to take streaming measurements of a video signal. In this streaming setting, however, each measurement will act on a different snapshot because the scene changes from time to time. To overcome this problem, the recovery of a video sequence from these measurements are based on the assumption that the scene changes slowly across a group of measurements. Under this assumption, the acquired measurements can be divided into non-overlapping groups, and each group approximately corresponds to one single frame. In [53], given the collected measurements, the whole video sequence can be reconstructed simultaneously using 3-D wavelets as the sparsity basis. The main drawback with this 3-D method is that it reconstructs the 3-D volume at once, which will incur high computation cost and large memory requirement.

2.3.2 Frame-difference Methods

To explore temporal correlation in videos and to improve the sampling efficiency, another category of approaches is to consider the sparse inter-frame difference. In these schemes, frames are measured and reconstructed indirectly based on the intensity changes between neighboring frames.

In [39], the video sequence was divided into several non-overlapping groups of frames, and each group has the same number of frames. The first frame in each group is used as a reference frame, which is uniformly sampled using traditional pixel-by-pixel raster scan. Meanwhile, compressed sensing is applied to each frame in the group (including the reference frame) using the same measurement matrix, and the differences between measurements of neighboring frames are recorded. In this way, each group in the video sequence is represented by a reference frame, followed by the difference of measurement results between each pair of neighboring frames. For slow-motion scenes, the intensity changes between two frames are small, and the frame difference is a sparse signal in the spatial domain, which is much sparser than the frame itself. Therefore, the frame difference can be represented by and reconstructed from a smaller number of measurements, which improves the sampling efficiency. In this scheme, the video sequence is reconstructed frame by frame, and each frame is calculated by adding the previous reconstructed frame and the reconstructed inter-frame difference.

The above work depends on the sparsity of the frame difference in the intensity domain and reconstructs the video frame by frame. In [54], the compressed measurements are collected in the same way as [39]. However, the inter-frame difference is considered as a sparse signal in the transform domain instead, and a group of frames are reconstructed simultaneously. This work uses two methods for joint reconstruction of multiple frames.

- In the first method, the reconstruction is based on the difference between the current frame and the previous reference frame. However, in this situation, the assumption that the inter-frame difference is small may not hold when the distance between the current frame and the reference frame is large.

- An alternative method is to evaluate frame changes between each pair of neighboring frames, since consecutive frames tend to have a larger correlation than two frames that are far apart. Compared to the first method, the frame difference is more sparse in the wavelet transform domain than the intensity domain, therefore, this method reconstructs videos with better quality given the same number of measurements.

Different from [39], the schemes in [54] jointly reconstruct multiple frames at the same time and, therefore, give better reconstruction quality at higher computation cost.

In conclusion, the advantage of this frame-difference method over independent 2-D or 3-D wavelet methods [53] is that static background pixels can be canceled out by comparing highly-correlated frames in a sequence. The only signals to be captured and recovered are the intensity changes caused by moving objects in the scene, which are much sparser than the frame itself for slowly-changing scenes. With the temporal redundancy removed, fewer measurements are required and the sampling efficiency is improved. However, there are two problems with these frame-difference methods.

- They depend on the sparsity of the inter-frame difference. Therefore, these methods are only effective on video sequences with small scene changes and are not suitable for sequences with large inter-frame difference, for example, videos with fast motions.

- When each frame is reconstructed based on its previous frame, there will be alias accumulation during reconstruction. That is, the reconstruction error in the previous frame may propagate to the current and the following frames. For fast-motion videos, the reconstruction quality provided by frame-difference methods may be even lower than 2-D or 3-D methods due to this alias accumulation.

2.3.3 Signal-dependent Methods

Compressed sensing is a non-adaptive and signal-independent sampling scheme. All compressed video acquisition schemes introduced above treat all regions in a frame and all frames in a sequence in the same way, and do not consider the fact that different regions may move independently at different speeds and have different texture complexity. In the literature, some signal-dependent methods for compressed sensing have been proposed to address this issue.

In [55], different sparsity levels of blocks were considered and a selective video sampling scheme was developed. A video sequence is divided into reference and non-reference frames. Each reference frame is uniformly sampled using traditional pixel-by-pixel raster scan. To exploit local sparsity within a frame, each frame (including the reference frame) is split into non-overlapping blocks of equal size. After that, a compressive sampling test is carried out to decide which blocks in a non-reference frame are sparse. Note that the sparsity of the block can only be evaluated when the original signal or its reconstructed version is available. However, neither can be obtained during the sampling stage. Therefore, the sparsity of blocks in a non-reference frame is estimated based on the nearest reference frame. During compressive sampling test, DCT is applied to each block in a reference frame. For a block, the total number of DCT coefficients whose absolute values are smaller than a constant C is calculated. If this number is larger than a pre-determined threshold T, the block is determined as a sparse block. Otherwise, it is determined as a non-sparse block. Compressed sensing is applied to sparse blocks, and conventional pixel-by-pixel raster scan is applied to non-sparse blocks.

By applying a compressive sampling test, this selective sampling scheme takes sparsity of regions into consideration, and adopts compressive or conventional sampling accordingly. However, it does not explore the temporal redundancies in video. For all sparse blocks, a fixed number of measurements are collected regardless of their sparsity. Also, since it still uses pixel-by-pixel raster scan on non-sparse blocks, a relatively large number of measurements are required when there are a large number of non-sparse blocks. In addition, it requires two sets of sampling hardware, one for compressed sensing and one for uniform sampling, and wastes resources.

Using a non-adaptive sampling strategy that collects a fixed number of measurements, the work in [56] proposed a framework of Model-based Adaptive Recovery of Compressive Sensing (MARX) where the adaptivity lies in the reconstruction process. Based on the collected random measurements, the recovery process uses a piecewise autoregressive (PAR) model to learn and exploit varying local structures of an image. Through the adjustment of its parameters, the PAR model can fit nonstationary images far better than a fixed set of bases (e.g., wavelets, DCT, and gradient spaces). The performance of MARX depends on the accuracy of the estimation of of PAR model parameters. However, a good initialization of the PAR model requires knowledge of the image to be recovered, which is not available during the recovery process.

To solve this problem, a compressive-uniform hybrid sensing system was proposed in [14] for image acquisition and it can be extended to video sampling. This hybrid sensing system takes two sets of observations of an image. One set consists random compressed sensing measurements, and the other is intensities of pixels obtained by raster scan uniform sampling. To reduce the total number of measurements to be collected, the uniformly sampled image may be a down-sampled version. In this way, compressed random sampling and conventional uniform down-sampling complement each other. From the uniformly down-sampled image, enough information required for the MARX algorithm can be obtained to learn local spatial structures of the image signal and to accurately estimate PAR model parameters accordingly.

However, the improvement on the reconstruction quality of this hybrid sensing scheme is at the cost of more measurement data. The information redundancy is introduced when collecting two sets of observations of an image. Also, since it is a frame-by-frame method when applied to video, it fails to explore the temporal correlation. Finally, in this method, the adaptivity is only considered during the signal recovery process, and a fixed sampling strategy is applied regardless of the frame features.

2.3.4 Motion Compensation Method

In the literature of standard video compression, a variety of methods have been proposed to remove spatial and temporal redundancy to produce sparse representation that are easier to compress. One typical approach is to apply motion compensation and estimation algorithms [57] along with image compression techniques. To apply these ideas to compressed sensing video acquisition, a major challenge to address is that in the standard video compression problem, the ground truth of the video is available for motion estimation. However, in compressed sensing, only random measurements are available at the sampling stage. There is a chicken-and-egg dilemma: given the motion information, the reconstruction quality of frames can be improved because inter-frame correlation can be better explored; however, in order to have a better estimation of the motion, the ground truth or the reconstructed version of video frames are required.

To resolve this dilemma, the work in [58] proposed an iterative multiscale framework for compressed sensing recovery, where different scales correspond to different decimated frame sizes. In the sampling stage, for each frame and for each scale, the encoder collects one set of random measurements. Each measurement is represented as a linear function of a single frame, which can be collected using compressed imaging hardware. Given the collected measurements, this framework uses a coarse-to-fine reconstruction algorithm, and switches between tasks of motion estimation and motion-compensated sparsity-based frame reconstruction. At each scale j, the reconstruction process can be divided into two stages. In the first stage, the scale-j version of the video is reconstructed based on motion vectors estimated at coarser scales. In the second stage, the reconstructed video obtained from the first stage is used to estimate and update motion vectors. The final reconstructed video is the output of the finest scale of this iterative algorithm.

By directly compensating for motion between frames, this multiscale framework explores the temporal redundancy in the video and improves reconstruction quality. However, this coarse-to-fine reconstruction algorithm is also very computationally demanding, which limits its applications.

Chapter 3

Maximum Frame Rate Video Acquisition by Adaptive Compressed Sensing

3.1 Block-based Adaptive Compressed Sensing For Video

Given the maximum number of measurements that can be collected by compressed sensing hardware per second, this thesis proposes an adaptive sampling framework that uses different sampling strategies in different regions to maximize the frame rate and to ensure satisfactory output quality. Each frame in the video sequence is divided into non-overlapping blocks of size $n \times n$, and all blocks in a frame are processed independently. In this work, the same SBHE matrix is used as the measurement matrix Φ for all blocks, which has been proven to be memory efficient, hardware friendly, and fast to compute. The whole framework is illustrated in Figure 3.1, and it contains three stages: frame rate selection, adaptive compressed sampling, and reconstruction.

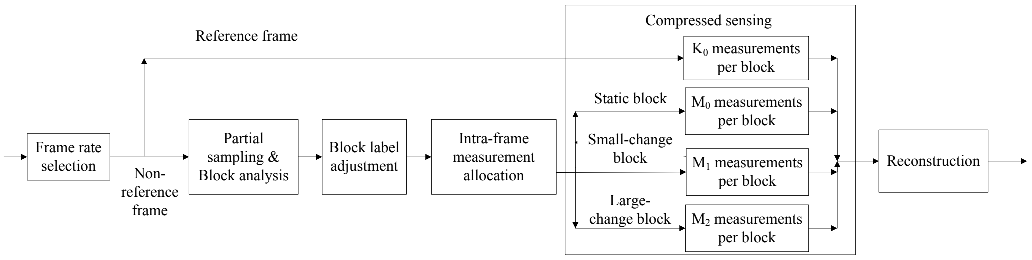

Compressed sensing
Reference frame
K0 measurements per block
Static block
M0 measurements per block
Frame rate selection
Non-reference frame
Partial sampling & Block analysis
Block label adjustment
Intra-frame measurement allocation
Small-change block
M1 measurements per block
Reconstruction
Large-change block
M2 measurements per block

Fig. 3.1. The proposed adaptive compressed video sampling framework.

Given a selected frame rate, this framework applies adaptive sampling to each frame. The first frame in the sequence is considered as a reference frame, and each block in the reference frame is acquired and reconstructed using the regular compressed image sensing technique. A fixed K_0 measurements are collected for each block in a reference frame. For each block in a non-reference frame, in the *partial sampling and block analysis* module, a very small number of measurements are first collected, and then compared with the measurements collected for the block at the same position in the immediately previous frame. Based on the comparison results, the correlation between these two blocks is estimated and used to classify the current block into different categories. Next, proper *block label adjustments* are applied to ensure that the first few frames in the video sequence have good quality and to address the alias accumulation problem [39,54]. In the *intra-frame measurement allocation* module, to enable a constant play of the reconstructed video, the same number of measurements are collected for all frames, and they are strategically allocated to all blocks in a frame to achieve the best perceptual quality. For each block, the *compressed sensing* and *reconstruction* strategies are then adjusted according to its block type and texture complexity. In this framework, if the block analysis module detects that a large portion of the blocks in a frame have little correlation with the previous frame, it is considered as a potential scene change. In such a scenario, the current frame is treated as a reference frame and a fixed K_0 measurements are collected for each block.

In this framework, given the hardware sampling rate constraint, the maximum achievable frame rate is also determined to satisfy the quality constraint on the reconstructed frames. In the *frame rate selection* module, an estimation process is applied before video acquisition, during which a list of candidate frame rates are examined. For each candidate frame rate, it collects a small amount of measurements to analyze the temporal correlation between neighboring frames and the scene complexity, estimates the average number of measurements required per second under the quality constraint, and compares it with the hardware sampling constraint to determine if the corresponding frame rate is achievable. It then selects the maximum achievable frame rate as the final frame rate of the video sequence.

Compared to the traditional raster scan method that provides the ground truth of the video sequence at a low frame rate, the proposed framework addresses the tradeoff between the frame rate and the perceptual quality in video acquisition, and maximizes the frame rate while maintaining satisfactory perceptual quality. This work uses the luminance component of a video sequence as an example, and it can be easily extended to color video acquisition.

3.2 The Physical Acquisition Device

The proposed adaptive compressed sensing framework can be realized using a single-pixel camera [11] with a DMD and a single photon detector, which sequentially takes measurements of a video signal. In this streaming setting, since the scene changes from time to time and each measurement acts on a different snapshot, same as in [53], the proposed framework assumes that the scene changes slowly across a group of measurements. Under this assumption, the acquired measurements can be divided into non-overlapping groups, and each group approximately corresponds to one single frame. For each frame, this framework acquires measurements block by block. When collecting measurements for a particular block, only the corresponding micromirrors are activated, and all others are set to ‘0’.

3.3 Performance Criteria

To evaluate the improvement in sampling efficiency of this framework, the frame rate enhancement ratio $C_{fr} = f_{cs}/f_{rs}$ is introduced, where f_{cs} and f_{rs} are the frame rate of the proposed adaptive compressed sensing scheme and that of the traditional raster scan, respectively, under the same hardware sampling rate constraint. A larger C_{fr} indicates that the proposed framework achieves higher sampling efficiency.

3.4 Block Analysis and Classification

To apply adaptive sampling, the temporal redundancy in video is explored to classify blocks into different types according to their inter-frame correlation.

3.4.1 Partial Sampling and Block Analysis

For block k of size $n \times n$ in the current frame t, it is represented as a spatial-domain signal $\mathbf{x}_t^k \in \mathscr{R}^{n^2}$. In this work, two blocks are called *co-located* if they are at the same location of different frames. The difference between the current block $\mathbf{x}_t^k$ and its co-located block in the previous frame $\mathbf{x}_{t-1}^k$ reflects the correlation between the two blocks in neighboring frames, and can be used to classify the current block $\mathbf{x}_t^k$. To address the issue that $\mathbf{x}_t^k - \mathbf{x}_{t-1}^k$ is not available at the sampling stage, the RIP [30] implies that the distances between the sparse signals can be well preserved in the measurement space, and also it follows that $\mathbf{y}_t^k - \mathbf{y}_{t-1}^k = \Phi\mathbf{x}_t^k - \Phi\mathbf{x}_{t-1}^k = \Phi(\mathbf{x}_t^k - \mathbf{x}_{t-1}^k)$, where $\mathbf{y}_t^k$ and $\mathbf{y}_{t-1}^k$ are the measurement vectors of $\mathbf{x}_t^k$ and $\mathbf{x}_{t-1}^k$, respectively [39]. From the above, the differences between the two measurement vectors also reflect the intensity changes in the two blocks and can be used to classify the block $\mathbf{x}_t^k$. Also, each measurement in $\mathbf{y}_t^k - \mathbf{y}_{t-1}^k$ is a linear combination of $\mathbf{x}_t^k - \mathbf{x}_{t-1}^k$, the pixel-wise difference of the two blocks. Therefore, the amount of intensity changes in the two blocks can be estimated by using only a small number of measurements.

Let Φ_{M_0} be a matrix containing the first $M_0 \leq K_0$ rows of the SBHE measurement matrix Φ. For block k in the current frame t, the block analysis module first uses Φ_{M_0} to collect M_0 measurements $\mathbf{y}_{M_0,t}^k = \Phi_{M_0}\mathbf{x}_t^k$ in the partial sampling stage. Then, it compares $\mathbf{y}_{M_0,t}^k$ with the first M_0 measurements in $\mathbf{y}_{t-1}^k$ and calculates the difference $\mathbf{y}_d^k = \mathbf{y}_{M_0,t}^k - \mathbf{y}_{M_0,t-1}^k$. Given $\mathbf{y}_d^k$, the block analysis module first calculates its l_1 norm normalized by M_0 and compares it with two thresholds T_1 and T_2 $(T_1 < T_2)$.

• If $\| \mathbf{y}_d^k \|_1 / M_0 \leq T_1$, the current block $\mathbf{x}_t^k$ is almost the same as block $\mathbf{x}_{t-1}^k$, and it is labeled as a static block. Due to high correlation between the current block and its co-located block in the previous frame, the M_0 measurements $\mathbf{y}_{M_0,t}^k$ collected in the partial sampling stage are sufficient to reconstruct $\mathbf{x}_t^k$ and there is no need to collect additional measurements for this block. Therefore, the highest sampling efficiency is achieved on static blocks by exploring the temporal correlation between a static block and its co-located block in the previous frame.

• If $T_1 < \| \mathbf{y}_d^k \|_1 / M_0 \leq T_2$, it indicates that the block undergoes small changes between these two neighboring frames. Compared to its co-located block in the previous frame, the current block contains some new information, which requires more measurements to be collected. For a block that undergoes small changes, this adaptive sampling framework collects a fixed number of $M_1 > M_0$ measurements in total. It uses the $(M_0 + 1)$th to the M_1th rows in the SBHE matrix Φ to collect additional $M_1 - M_0$ measurements, and combines with $\mathbf{y}_{M_0,t}^k$ to generate the final measurement vector for block $\mathbf{x}_t^k$. That is, $\mathbf{y}_{M_1,t}^k = \Phi_{M_1}\mathbf{x}_t^k$, where Φ_{M_1} contains the first M_1 rows in Φ. Still, exploration of the temporal correlation helps achieve

TABLE 3.1

Block types and their properties

Block type	Measurement #	Block properties
static	M_0	almost the same as its co-located block, achieves the highest sampling efficiency
small-change	$M_1 > M_0$	contains some new information, achieves reasonably high sampling efficiency
large-change	$M_2 > M_1$	significantly different from its co-located block, independently processed based on its texture complexity

reasonably high sampling efficiency on such small-change blocks.

• If $\| \mathbf{y}_d^k \|_1 / M_0 > T_2$, the two blocks are significantly different, which is most likely due to objects' movement, it is labeled as a large-change block. A large-change block is considered as independent from its co-located block, and will be processed based on its spatial characteristics only. For a large-change block, a total of $M_2 > M_1 > M_0$ measurements are collected during the sampling process, and the number M_2 depends on the texture complexity and the spatial characteristics of the block. In this case, a relatively large number of measurements are assigned and temporal correlation is not used to reduce the number of measurements collected.

Different block types and their properties are summarized in Table 3.1.[1]

3.4.2 Block Label Adjustment

After every block in a non-reference frame is classified based on the M_0 partial measurements, to ensure satisfactory output quality of the whole sequence, further adjustments on the resulting block labels are necessary for two purposes: to obtain

[1] Static blocks are similar to skipped macroblocks in video coding, and both explore temporal redundancy to achieve high efficiency. Large-change blocks are in the same spirit as intra macroblocks in video coding, and both are processed independently based on their spatial characteristics only.

a good initiation on the frame quality, and to prevent alias accumulation.

A Good Initiation on the Frame Quality

In this framework, the reference frame is assigned a fixed number of measurements. Due to the diverse texture complexity of different scenes, there is no guarantee that the quality of the reconstructed reference frame can meet the requirement. For example, in simulations, with $K_0 = 0.4n^2$ measurements for each block, the PSNR of the reference frame in Foreman can achieve 30.8dB, while for Tempete, it is only 23.8dB. Therefore, for scenes with slow motion and complex textures where there are a large number of static blocks in the first few non-reference frames, the quality of the reference frame may be below the quality constraint, and PSNR of the following frames will follow that of the reference frame. To address this issue, label adjustment is applied to static blocks in the first two non-reference frames.

For each block in the first non-reference frame (frame 2) that is initially determined to be static, this scheme first uses Equation (3.2) that will be introduced in Section 3.5.1 to estimate the number of measurements M_2 required, if it is labeled as a large-change block and if it will be sampled independently from its co-located block in the reference frame. If $M_2 > K_0$, it is necessary to collect more measurements to ensure the quality of the reconstructed frame, and relabel it as a large-change block. Note that this relabeling may insert many large-change blocks in one frame, which may cause large fluctuation in the required number of measurements per frame, and may increase the complexity of the intra-frame measurement allocation module. To address this issue, this adjustment process only relabels up to half of the static blocks in the first non-reference frame (frame 2) as large-change blocks, and applies the rest of label adjustment in the next non-reference frame (frame 3). For a static block $\mathbf{x}_3^k$ in the second non-reference frame, if its co-located block in the 2nd frame $\mathbf{x}_2^k$ is also labeled as static, that is, $\mathbf{x}_2^k$ is labeled as static and has not been adjusted, the number of measurements M_2 required is estimated if $\mathbf{x}_3^k$ is to be sampled independently. If $M_2 > K_0$, $\mathbf{x}_3^k$ is labeled as a large-change block. This label adjustment ensures that the first few frames in the sequence satisfy the quality constraint, and will not be affected by the fixed sampling strategy for the reference frame.

Prevention of Alias Accumulation

Another issue to be addressed is the alias accumulation (error propagation) [39]: the reconstruction error in the previous frame may propagate to the current and even future frames. This is because in this framework, as will be discussed in Section 3.7, small-change blocks and part of static blocks are reconstructed based on their co-located blocks in the previous frame. To prevent alias accumulation, large-change blocks are periodically inserted using the following block label adjustment scheme.

First, for all blocks in frame t, the ones considered are those who are labeled either as a static or a small-change block and whose co-located blocks in frame $t-1$ are small-change blocks. Assume there are a total of N_p such blocks in frame t, and let $Q_t = \{k_1, k_2, \ldots, k_{N_p}\}$ be the set containing their indices. At least half of them are relabeled, and this block relabeling process contains two steps.

- For a block $\mathbf{x}_t^k$ where $k \in Q_t$, if $\mathbf{x}_{t-2}^k$ is also a small-change block, that is, there are two small-change blocks at the same location k in frame $t-2$ and frame $t-1$, block $\mathbf{x}_t^k$ will be forced to be labeled as a large-change one. This label adjustment ensures that two consecutive small-change blocks are followed by a large-change block to prevent further alias accumulation. Assume that Step 1 modifies N_q such blocks in Q_t at frame t, and A_t is the set containing their indices.

- Next, if $N_q < \lceil 0.5N_p \rceil$, that is, less than half of the blocks in Q_t are relabeled in the previous step, This step will continue to force $\lceil 0.5N_p \rceil - N_q$ of the remaining ones in Q_t to be relabeled as large-change blocks. This will ensure that at least a total of $\lceil 0.5N_p \rceil$ blocks in Q_t are labeled as large-change blocks. Among the $N_p - N_q$ remaining blocks in Q_t whose labels are not modified in the previous step, $\lceil 0.5N_p \rceil - N_q$ of them with the largest partial measurement difference $\| \mathbf{y}_d^k \|_1$ are selected and relabeled as large-change blocks.

From the above, the total number of relabeled blocks is limited to $\mathbf{max}\{\lceil 0.5N_p \rceil, N_q\}$, in order to avoid having too many large-change blocks in one frame.

Second, in the current frame t, the relabeling module will find static blocks $\mathbf{x}_t^k$ whose co-located blocks in the previous 5 frames $(\mathbf{x}_{t-5}^k, \dots, \mathbf{x}_{t-1}^k)$ are either static or small-change blocks, and assume there are a total of N_s such blocks. This label adjustment will relabel $\lceil 0.5N_s \rceil$ of them as large-change blocks to address alias accumulation while avoiding large increase in number of measurements. Let B_t be the set containing the indices of these blocks re-labeled. The adjustment is applied to consecutive static blocks, or, static and small-change blocks that appear alternately. Compared to small-change blocks, the reconstruction error of static blocks is smaller, and the impact of error propagation is less serious. Therefore, for consecutive static blocks, or alternate static and small-change blocks, large-change blocks are inserted less frequently than the scenario when there are consecutive small-change blocks.

3.5 Adaptive Sampling and Intra-frame Measurement Allocation

After block classification, different sampling strategies are used for different types of blocks. In particular, for each large-change block, its texture complexity is considered to adaptively select the number of measurements collected for that block. Note that in this framework, each frame is assigned a fixed number of measurements to ensure that the video can be played at a constant speed. To address this issue, for each frame, this work first estimates the number of measurements that each block needs to satisfy the quality constraint, and then it allocates the measurements among all blocks in the frame to ensure the best overall quality.

3.5.1 Estimation of the Required Number of Measurements for Each Block

As discussed before, static and small-change blocks explore the temporal correlation to achieve high sampling efficiency, and they carry little new information when compared to their co-located blocks in the previous frame. Therefore, a fixed small number of measurements are assigned to each static block and each small-change block, which is M_0 for a static block and M_1 for a small-change block.

For large-change blocks, no previous information can help reduce the number of measurements, and the sampling process is based on the spatial features. Thus, for a large-change block, this framework uses the regular compressed sensing technique for images, and collects a total of M_2 measurements (including the initial M_0 measurements) during the sampling process. In the following, for each large-change block, how to estimate the number of measurements required to satisfy the quality constraint will be discussed.

Estimation of the Required Number of Measurements

Large-change blocks are processed independently from their co-located blocks. Different blocks may have different texture complexity (sparsity), and may require different number of measurements M_2 to ensure satisfactory quality of the reconstructed frame. Therefore, the sampling processing for large-change blocks should be adjusted according to their spatial characteristics. It has been shown in [7] that the solution to (2.4) satisfies

$$\| \hat{\mathbf{x}} - \mathbf{x} \|_2 \leq C \cdot R \cdot (M/\log(N))^{0.5-1/p}, \tag{3.1}$$

where R is the l_p norm of the transform coefficients α, N is the length of the original signal $\mathbf{x}$, M is the number of collected measurements $\mathbf{y}$, and $p < 2$ and C are constants. Given the constraint that the MSE between the reconstructed and the original blocks does not exceed ε, equation (3.1) can be used to find the lower bound on the number of measurements for each large-change block. This work uses $p = 2/3$ and replaces R in (3.1) with the block sparsity S, that is, the l_0 norm of the transform coefficients α, and the lower bound of M is

$$M_{lb} = \log(N) \left[\varepsilon/(C \cdot S)\right]^{1/(0.5-1/p)}. \tag{3.2}$$

Here, the l_0 norm is used since it is easier to compute than other l_p ($p < 2$) norm of the transform coefficients, which makes it suitable for real-time processing of signals and practical acquisition applications. Also, from simulation results, for different sequences with different textures and spatial characteristics, with $p = 2/3$ and the block sparsity S, a universal constant C can be found that makes (3.2) a good estimator of the required number of measurements for large-change blocks. The work in [27] also used the sparsity level to calculate the sufficient number of measurements for sparse images.

In (3.2), M_{lb} depends on the sparsity of the block, which can only be evaluated when the original signal or its reconstructed version is available. However, neither can be obtained during the sampling stage. To address this issue, during the sampling stage, this framework periodically uses fast algorithms (e.g., GPSR) to solve (2.4) and reconstruct a few frames, which are called the *indicator frames*. In this work, the nearest reconstructed indicator frame is used to estimate the sparsity of the current block. It is assumed that these indicator frames can be successfully recovered, and the reconstructed version can faithfully reflect the scene complexity.

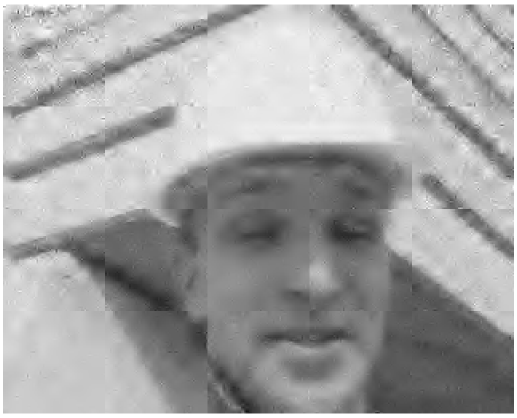
Note that if the indicator frame is not a reference frame, to reconstruct a static or a small-change block in the indicator frame, this scheme should also reconstruct its co-located blocks in the previous frames, until it finds a co-located block that is large-change and can be reconstructed independently. From the block label adjustment process in Section 3.4.2, for such a purpose, it may be necessary to trace back up to five previous frames.

For frame t, let $i_t < t$ be its nearest indicator frame. For the current block k in frame t, this work compares its first M_0 measurements $\mathbf{y}^k_{M_0,t}$ with that of blocks in the indicator frame i_t, and searches for the *indicator block* k' in frame i_t that minimizes $\| \mathbf{y}^k_{M_0,t} - \mathbf{y}^{k'}_{M_0,i_t} \|_1$. To increase the searching speed and accuracy, the block k' is searched within a 3×3 neighbor centered around location k, since it is assumed that the corresponding object would not move out of this neighbor during the time interval between the two frames. Let $\hat{\mathbf{x}}^{k'}_{i_t}$ be the reconstructed indicator block k' in frame i_t and its corresponding transform coefficients be $\hat{\alpha}^{k'}_{i_t}$. In this paper, the estimated sparsity of $\mathbf{x}^k_t$ is the number of coefficients in $\hat{\alpha}^{k'}_{i_t}$ whose magnitudes are larger than a predetermined threshold l. Given the estimated sparsity $\tilde{S}$ and the upper bound on MSE ε (or equivalently, the lower bound on PSNR), this framework uses (3.2) with $p = 2/3$ and $C = 220$ to determine M_2, the total number of measurements to be collected for $\mathbf{x}^k_t$ to satisfy the quality constraint $\| \hat{\mathbf{x}} - \mathbf{x} \|_2 \leq \varepsilon$.

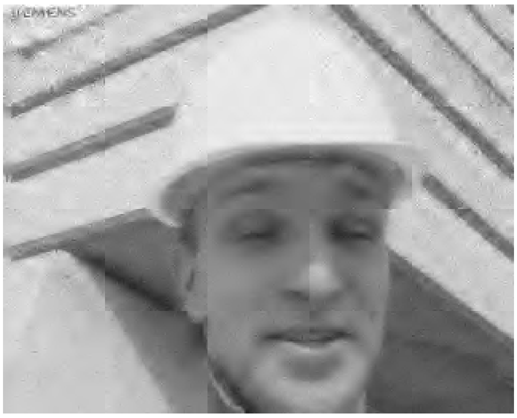
Lower and Upper Bounds of M_2

To ensure a reasonable choice of M_2, this scheme requires that M_2 is in the range $[0.3n^2, n^2]$, so it follows that:

$$M_2 = \begin{cases} 0.3n^2, & M_{lb} < 0.3n^2, \\ \underline{M}, & 0.3n^2 \leq M_{lb} < n^2, \\ n^2, & M_{lb} \geq n^2. \end{cases} \tag{3.3}$$



(a) the reconstructed frame with $K_0 = 0.2n^2$



(b) the reconstructed frame with $K_0 = 0.3n^2$

Fig. 3.2. The reconstructed reference frame of Foreman with different sampling ratio.

Here a lower bound of $0.3n^2$ is imposed on M_2 because it is observed that a small number of measurements may lead to blurring and artifacts in the reconstructed frames even if it gives a moderate PSNR. To demonstrate this and make the blurring effect more noticeable, a small value of M_2 is used on all blocks within a frame. Figure 3.2 shows the reconstructed frame 1 of Foreman. With $M_2 = 0.2n^2$ measurements assigned to each block, which is much lower than the lower threshold $0.3n^2$, the reconstructed frame has a moderate PSNR of 26.1dB. However, it shows obvious blurring effect in the face regions (month, eyes, etc.) and artifacts in the background. When M_2 is increased to $0.3n^2$, the PSNR increases to 28.8dB, and the details can be seen more clearly. Therefore $0.3n^2$ is used as a reasonable lower bound for the number of block measurements if the block is to be sampled independently.

Also, this work makes M_2 be upper bounded by n^2. This is because if $M_2 > n^2$, compressed sensing loses its advantage over pixel-by-pixel raster scan.

3.5.2 Intra-frame Measurement Allocation

In the previous section, the minimum number of measurements required for each block to satisfy the quality constraint is estimated, and different blocks in a frame are processed independently. To ensure a reconstructed video can be played at a constant speed, the same number of measurements should be assigned to all frames. Given the hardware sampling rate constraint that no more than M_{max} measurements can be collected per second, for a given frame rate f_{cs}, there are a total of $M_f = M_{max}/f_{cs}$ measurements for each non-reference frame, which should be distributed within a frame with joint consideration of all blocks to maximize the overall perceptual quality.

Assume there are N_0 static blocks, N_1 small-change blocks, and N_2 large-change blocks in the current frame. The procedure of intra-frame measurements allocation is as follows:

⧫ Step 1 Allocate measurements to static and small-change blocks: For each static block, M_0 measurements are sufficient, and for each block with small changes, M_1 measurements are assigned. Therefore, there are a total of $M_{f2} = M_f - M_0 \cdot N_0 - M_1 \cdot N_1$ measurements left for large-change blocks.

⧫ Step 2 Allocate measurements to large-change blocks: M_{f2} measurements should be allocated to the N_2 large-change blocks to achieve the best quality. According to Section 3.5.1, the number of measurements for each large-change block to achieve certain quality constraint is determined by (3.2), and with $p = 2/3$, it is proportional to block sparsity S. However, when there is a constraint on the number of measurements per frame, there is no guarantee that all large-change blocks in the current frame t can get enough measurements that they need to satisfy the quality requirement. Therefore, allocation of the limited measurements to all blocks to achieve the best overall quality becomes a constrained optimization problem. According to (3.2), for a large-change block i, M_2^i measurements gives a reconstruction error of $\varepsilon_i = C \cdot S_i \cdot (M_2^i / \log(N))^{0.5-1/p} = C \cdot S_i \cdot (\log(N)/M_2^i)$ with $p = 2/3$, where S_i is the sparsity of block i. Then the constrained optimization problem is formulated as

$$\{\hat{M}^i_{2,\ i=1,\ldots,N_2}\} = \arg\min \sum_{i=1}^{N_2} \varepsilon_i \qquad s.t. \ \sum_{i=1}^{N_2} M_2^i = M_{f2}. \tag{3.4}$$

It can be converted into an unconstrained one using Lagrange multipliers, and the Lagrange function is

$$\{\hat{M}^i_{2,\ i=1,\ldots,N_2},\ \lambda\} = \arg\min \sum_{i=1}^{N_2} \varepsilon_i + \lambda (\sum_{i=1}^{N_2} M_2^i - M_{f2}), \tag{3.5}$$

where λ is the Lagrange multiplier. To solve (3.5), take the first-order partial derivative of the cost function in (3.5) with respect to M_2^i $(i = 1, \ldots, N_2)$, and the optimal $\hat{M}_2^i$ should satisfy

$$-\frac{C \cdot S_i \cdot \log(N)}{(\hat{M}_2^i)^2} + \lambda = 0 \qquad i = 1, \ldots, N_2. \tag{3.6}$$

The equation array (3.6) shows that with constraint on the total number of measurements for large-change blocks, the optimal $\hat{M}_2^i$ should be proportional to $\sqrt{S_i}$, the square root of the block sparsity, that is,

$$\hat{M}_2^1 : \hat{M}_2^2 : \ldots : \hat{M}_2^{N_2} = \sqrt{S_1} : \sqrt{S_2} : \ldots : \sqrt{S_{N_2}}. \tag{3.7}$$

With a total of M_{f2} measurements available for large-change blocks in the current frame, given their estimated block sparsity $\{\tilde{S}_i\}_{i=1}^{N_2}$, the optimal solution is

$$\hat{M}_2^i = \frac{\sqrt{\tilde{S}_i}}{\sum_{i=1}^{N_2} \sqrt{\tilde{S}_i}} \cdot M_{f2} \qquad i = 1, \ldots, N_2. \tag{3.8}$$

⧫ Step 3 Set bounds to $\hat{M}_2$: Same as in (3.3), the number of measurements collected for a large-change block is forced to be in the range $[0.3n^2, n^2]$. When $\hat{M}_2^i$ in (3.8) is outside this range, further adjustment is necessary.

This step first makes sure all large-change blocks are assigned at least $0.3n^2$ measurements. If frame t has many large-change blocks and M_{f2} is small, it is possible that some large-change blocks are assigned fewer than $0.3n^2$ measurements. Step 3.1 and 3.2 should be followed to deal with this situation.

- Step 3.1: If there are some blocks that are assigned more than n^2 measurements, the measurements can be relocated as follows. First it determines the block l with the smallest number of measurements where $\hat{M}_2^l < 0.3n^2$, and the block h that has the largest number of measurements and $\hat{M}_2^h > n^2$ within the frame. If $\hat{M}_2^h - (0.3n^2 - \hat{M}_2^l) \geq 0.3n^2$, $(0.3n^2 - \hat{M}_2^l)$ measurements can be transferred from block h to block l, and $\hat{M}_2^l$, $\hat{M}_2^h$ can be updated as

$$\begin{cases} \hat{M}_2^l = 0.3n^2, \\ \hat{M}_2^h = \hat{M}_2^h - (0.3n^2 - \hat{M}_2^l). \end{cases} \tag{3.9}$$

  The condition $\hat{M}_2^h - (0.3n^2 - \hat{M}_2^l) \geq 0.3n^2$ ensures that the updated $\hat{M}_2^h$ is at least $0.3n^2$ and is above the lower bound. This process is continued until there are no more blocks that are assigned more than n^2 measurements. If $\hat{M}_2^h - (0.3n^2 - \hat{M}_2^l) < 0.3n^2$, Step 3.2 is adopted.[2]

- Step 3.2: If there are no blocks where $\hat{M}_2 > n^2$ or there are still blocks with $\hat{M}_2 < 0.3n^2$ after Step 3.1, this scheme will reduce the number of large-change blocks and redo measurement allocation in Step 2. First it considers all large-change blocks in the current frame, excluding those in set A_t and B_t defined in Section 3.4.2. Among these large-change blocks, the block with the smallest partial measurement difference $\| \mathbf{y}_d^k \|_1$ will be relabeled as small-change. As a result, there are a total of $M_{f2} - M_1$ measurements to be distributed among the remaining $N_2 - 1$ large-change blocks. Then this step repeats the measurement allocation in (3.8) and the measurement transfer process in Step 3.1, and checks if $\hat{M}_2 \geq 0.3n^2$ is satisfied for all the remaining large-change blocks. If not, it will relabel another large-change block as a small-change one and repeat the above process. This process is continued until all large-change blocks have at least $0.3n^2$ measurements.

Then this scheme considers the upper bound n^2 and checks if there are still large-change blocks that are assigned more than n^2 measurements. For block h with $\hat{M}_2^h > n^2$ measurements, it sets $\hat{M}_2^h = n^2$.

♦ Step 4 Allocate extra measurements: After Step 3, if there are $M_e > 0$ extra measurements that have not been assigned to any blocks, it is necessary to allocate them to blocks to fully utilize the assigned measurements per frame. Note that in this framework, static and small-change blocks are reconstructed using block difference, which may cause reconstruction error propagation. With extra measurements available, some of them can be relabeled as large-change blocks for better output

[2] Note that more complicated schemes can also be used, for example, combining and transferring extra measurements from several blocks to block l. In this work, to reduce the system complexity, it only allows the measurements be transferred from one block to another.

quality.

- First, all current small-change blocks in frame t are sorted in the descending order of their partial measurements differences $\| \mathbf{y}_d^k \|_1$. Then block k with the largest $\| \mathbf{y}_d^k \|_1$ is relabeled as a large-change block and assigned M_2^k measurements if $M_2^k \leq M_e$. Here, M_2^k is calculated using (3.2) and bounded by $0.3n^2$ and n^2. After that, it moves to the small-change block with the second largest partial measurement difference and repeats the above label adjustment until all the M_e extra measurements are allocated.

- Second, if there are still some extra measurements left after all small-changed blocks are relabeled as large-change ones, this label adjustment process will be further applied to static blocks until all the M_e extra measurements are allocated.

With M_f measurements properly allocated within the current frame, each block can be adaptively sampled according to its block type and the number of measurements assigned.

3.6 Frame Rate Selection

Given a frame rate, the previous two sections have introduced how to adaptively adjust the sampling strategy according to the inter-frame correlation and the scene complexity of the scene. In this section, assuming that the hardware can sample up to M_{max} measurements per second, it discusses how to select the maximum achievable frame rate for the video acquisition system under the hardware sampling rate and the quality constraints.

For video applications, a high frame rate is often desired to avoid obvious flicker effects, especially for fast moving scenes. In this framework, given a list of candidate frame rates, it determines the maximum achievable frame rate under the hardware sampling rate and the quality constraints. For example, this work considers the following levels of frame rate: 60 frames per second (fps), 30 fps, 20 fps, 15 fps, 12 fps, and 10 fps, from which a suitable frame rate will be selected to satisfy the quality constraint. These candidate frame rates are named as level 1 to level 6, where level 1 refers to $f_1 = 60$ fps and level 6 refers to $f_6 = 10$ fps. In this work, 6 frame rate levels are used as an example to show the adaptivity and performance of the proposed framework, and this framework can be easily extended to support more frame rate levels.

The first second is used to do frame rate estimation. For each candidate frame rate, this framework first estimates the average number of measurements per second to satisfy the quality constraint, compares with the hardware sampling rate constraint, and determines if it is feasible. Then is selects the maximum feasible frame rate as the selected frame rate of the video sequence. The frame rate selection process is summarized in Algorithm 1.

In the frame rate estimation module, same as in the previous section, the first frame is considered as a reference frame and K_0 measurements are collected for each block in the reference frame. Then frame 1 is reconstructed and used as the indicator frame to help estimate the number of measurements required for the following non-reference frames. For each candidate frame rate, the next step is to collect M_0 measurements for each block in a non-reference frame and estimate the number of measurements required to satisfy the quality constraint. Given six candidate frame rates, to avoid sampling six times at different frame rates, the following scheme is used to sample once and collect all measurements that are needed for all six candidate frame rates.

Given the six candidate frame rates in the example, first their least common multiple is found, which is $f_{lcm} = 60$ fps. Then the frame rate estimation module samples the non-reference frames at frame rate f_{lcm} and collects M_0 measurements for each block in a non-reference frame. Let $\mathbf{Y}_{t,\ lcm}$ be the collected measurements for frame t where $1 \leq t \leq f_{lcm}$ when sampled at frame rate f_{lcm}. Given $\{\mathbf{Y}_{t,\ lcm}\}_{1\leq t\leq f_{lcm}}$, it calculates the corresponding measurements $\mathbf{Y}_{t,j}$ for frame t when sampled at level-j frame rate f_j. In this work, $f_{lcm} = f_1$ and $\mathbf{Y}_{t,1} = \mathbf{Y}_{t,\ lcm}$. For level-$j$ frame rate where $j > 1$, a simple frame skipping can be used to obtain the measurements where

$$\mathbf{Y}_{t,j} = \mathbf{Y}_{j(t-1)+1,\ lcm}, \quad t = 1,\ldots,60/j,\ j = 1,\ldots,6. \tag{3.10}$$

That is, the level-2 (30 fps) measurements can be obtained from $\{\mathbf{Y}_{t,\ lcm}\}$ by skipping one frame in every two frames, the level-3 (20 fps) measurements can be obtained by skipping two in every three frames, etc.

Given the partial measurements $\{\mathbf{Y}_{t,j}\}$ for frame rate f_j, this frame rate estimation module estimates the total number of measurements required per second to satisfy the quality requirement, and it starts with the highest candidate frame rate $f_1 = 60$ fps. First, it follows Section 3.4 to analyze and label each block with block label adjustment. Then this module determines the number of measurements for each block based on its type: a static block is assigned M_0 measurements; a block with small change is assigned M_1 measurements; the number of measurements M_2 for a large-change block is calculated using (3.2). Note that this step is to estimate the number of measurement required to satisfy the quality constraint, which does not require that all frames have the same number of measurements. Thus, intra-frame measurement allocation is not applied when estimating the frame rate, and (3.2) rather than (3.8) is used to estimate the number of measurements. In addition, for a large-change block, given M_{lb} calculated using (3.2), its lower and upper bounds are set as $0.3n^2$ and $0.9n^2$, respectively. Note that in the frame rate estimation module, $0.9n^2$ is used instead of n^2 in (3.3) as the upper bound. This is because the simulation results show that these two bounds give similar output quality, while

$0.9n^2$ gives a relatively smaller estimated average number of measurements per second and thus a potentially higher frame rate. Also, the number of measurements for the reference frame (frame 1) is not included in the calculation of total number of required measurements. This is because during the video acquisition process, a reference frame is only inserted when scene change happens, and most frames in a video sequence are non-reference frames. Therefore, the frame rate estimation module only considers non-reference frames and ignores the reference frames that are sampled using a fixed strategy. For level 1 with frame rate 60 fps, this framework only calculates the required number of measurements for the 59 non-reference frames in the 1st second, based on which it estimates the corresponding average number of measurements per second.

Let $M_{sec,1}$ be the estimated number of measurements to satisfy the quality requirement with frame rate 60 fps. If $M_{max} \geq 0.98M_{sec,1}$, which means the average number of measurements per second is below the hardware sampling rate, then frame rate 60 fps is selected and this frame rate estimation process terminates. Here a tolerance level of 2% is introduced and the measurements budge M_{max} is compared to $0.98M_{sec,1}$ instead of $M_{sec,1}$. If $M_{max} < 0.98M_{sec,1}$, this module continues to check whether the next level frame rate (30 fps) is feasible, stop when $M_{max} \geq 0.98M_{sec,j}$ is satisfied, and select the corresponding frame rate f_j.

3.7 Video Reconstruction

After the video is adaptively acquired at the selected frame rate, the entire sequence can be reconstructed block by block and frame by frame. Different types of blocks are reconstructed in different ways.

If block k at frame t is a large-change block, given the measurement vector $\mathbf{y}^k_{M_2,t}$, $\hat{\mathbf{x}}^k_t$ can be found using the regular compressed sensing reconstruction algorithm, for example, by solving (2.4) using GPSR [37] or solving (2.5) using min-TV.

For a small-change block, the block difference $\mathbf{x}^k_{t,t-1} = \mathbf{x}^k_t - \mathbf{x}^k_{t-1}$ is a sparse signal, which is much sparser than $\mathbf{x}^k_t$ in the DCT/DWT domain. Therefore, this work first uses the regular compressed sensing reconstruction algorithm to reconstruct the block difference $\hat{\mathbf{x}}^k_{t,t-1}$, and then adds it to the reconstructed block k at frame $t-1$ ($\hat{\mathbf{x}}^k_{t-1}$) to get $\hat{\mathbf{x}}^k_t = \hat{\mathbf{x}}^k_{t-1} + \hat{\mathbf{x}}^k_{t,t-1}$.

A static block is highly correlated to its co-located block in the previous frame, from which it can be well predicted. Assume that $\mathbf{y}^k_{M,t-1}$, the measurement vector for $\mathbf{x}^k_{t-1}$, is of length M. Then for the current static block $\mathbf{x}^k_t$, the final measurement vector $\mathbf{y}^k_{M,t}$ is simply a concatenation of the initial M_0 measurements $\mathbf{y}^k_{M_0,t}$ collected in the partial sampling stage and the last $M - M_0$ elements in $\mathbf{y}^k_{M,t-1}$, that is,

$$\mathbf{y}^k_{M,t} = \left[\mathbf{y}^k_{M_0,t};\ y^k_{M,t-1}(M_0+1);\ \ldots;\ y^k_{M,t-1}(M)\right], \tag{3.11}$$

where $y^k_{M,t-1}(i)$ is the ith element in $\mathbf{y}^k_{M,t-1}$. For a static block, there are two reconstruction methods. First, when its co-located block $\mathbf{x}^k_{t-1}$ is a large-change block, the measurement vector $\mathbf{y}^k_{M_2,t-1}$ in (3.11) contains M_2 measurements and all information necessary to reconstruct the current block. In this scenario, $\hat{\mathbf{x}}^k_t$ can be directly reconstructed using the regular compressed sensing reconstruction algorithm. Second, if its co-located block $\mathbf{x}^k_{t-1}$ is not a large-change one, the measurement vector $\mathbf{y}^k_{M,t}$ in (3.11) does not contain sufficient information to directly reconstruct the current block $\mathbf{x}^k_t$. In this scenario, since the block difference $\mathbf{x}^k_{t,t-1}$ is also a very sparse signal, $\mathbf{x}^k_t$ is reconstructed in the same way as the reconstruction of a small-change block. Here, the block difference $\hat{\mathbf{x}}^k_{t,t-1}$ is first reconstructed from $\mathbf{y}^k_d = \mathbf{y}^k_{M,t} - \mathbf{y}^k_{M,t-1}$, and then added to the reconstructed co-located block $\hat{\mathbf{x}}^k_{t-1}$ to obtain $\hat{\mathbf{x}}^k_t = \hat{\mathbf{x}}^k_{t-1} + \hat{\mathbf{x}}^k_{t,t-1}$. Comparing these two methods, the former method is faster since the block is reconstructed from more measurements, which accelerates the reconstruction process. Therefore, in this work, the first method is used whenever possible to reconstruct a static block.

Algorithm 1: Frame rate selection

```
Acquire $K_0$ measurements for each block in the reference frame (frame 1) and reconstruct frame 1;
Sample the following non-reference frames at frame rate $f_{lcm}$ with $M_0$ measurements for each block;
for level $j$=1:6 (frame rate=60 fps to 10 fps) do
    generate the collected measurements $\{\mathbf{Y}_{t,j}\}$ for frame rate $f_j = 60/j$ fps; $M_{sec,j} = 0$;
    for each block in a non-reference frame do
        analyze the block type with block label adjustment;
        switch: block type
        case static block: assign $M_0$ measurements; $M_{sec,j} = M_{sec,j} + M_0$;
        case small-change block: assign $M_1$ measurements; $M_{sec,j} = M_{sec,j} + M_1$;
        case large-change block: estimate its sparsity, use (3.2) to calculate the required number of measurements $M_2$, and bound it in the range $[0.3n^2, 0.9n^2]$;
        $M_{sec,j} = M_{sec,j} + M_2$;
        end switch
    end for
    $M_{sec,j} = M_{sec,j} * \frac{60/j}{60/j-1}$;
    if $M_{max} \geq 0.98 M_{sec\_j}$ then
        $f_{cs} = 60/j$ fps; break;
    end if
end for
return $f_{cs}$
```

Chapter 4

Performance Analysis and Simulation Results

The previous chapter introduces the adaptive compressed sensing for video acquisition with maximum frame rate estimation, and this chapter will give a thorough performance analysis of the improvement in sampling efficiency, and show the simulation results for the proposed framework.

The performance of the proposed framework is tested on 3 video sequences: Foreman, Coastguard, and Tempete. The frame size is set to 256×320 that is cut from the 288×352 CIF video sequences. All frames are split into non-overlapping blocks of size 64×64 with $n = 64$. Define the sampling ratio as the number of collected measurements over the total number of pixels. The sampling ratio for the reference frames is fixed at 40%, and sampling ratios for blocks in non-reference frames are decided based on their block types and the estimated sparsity in the 9-7 wavelet domain. The reference frame is used as the indicator frame to estimate the block sparsity. $M_0 = 0.03n^2$, $M_1 = 0.1n^2$, $T_1 = 4$, $T_2 = 9.5$, and $l = 0.1$ are used in the simulations. The simulation applies two algorithms to reconstruct the sequences: the l_1 minimization using GPSR with the 9-7 wavelet as the sparse basis, and min-TV in the l_1 magic package. Since GPSR benefits from a good initial starting point, for block k in frame t, if it is to be reconstructed directly, $\hat{\mathbf{x}}_{t-1}^{k}$, the reconstructed block k in the previous frame, is used as the initial point to accelerate the reconstruction process.

4.1 Estimation of the Required Number of Measurements at a Fixed Frame Rate

In the frame rate selection module, the first step is to estimate the average number of measurements required to achieve the required video quality at a fixed frame rate. To demonstrate this process, test is conducted on 1 second's frames for each sequence with frame rate 30 fps. As an example, the simulation sets the lower bound on the PSNR of the reconstructed frames as 24dB, 27dB and 30dB, respectively, and a similar trend is observed for other values of the quality constraint. Table 4.1 shows the sampling ratios for different sequences with different quality constraints. Results of the reference frames are not included in Table 4.1, since a fixed sampling and reconstruction strategy is used there. As can be seen, this framework adaptively adjusts the sampling process to the scene. The Coastguard and the Tempete sequences have fast changing scenes and are texture rich. Thus, more measurements are collected to ensure high quality, and the sampling ratio varies from 35.6% to 57.9% for Coastguard, and from 35.6% to 52.1% for Tempete, depending on the quality constraint. On the other hand, with slow motion and relatively simple scene composition, Foreman is acquired with fewer measurements to achieve a higher sampling efficiency of 18.6% to 25.5%.

To show that this estimation process can accurately estimate the number of measurements required to achieve the required quality, Table 4.1 also lists the average PSNR of the reconstructed frames. Here in Table 4.1, for each large-change block, a

TABLE 4.1

Minimizing the number of measurements at a fixed frame rate

	GPSR: average PSNR(dB)			sampling ratio(%)		
PSNR bound	Foreman	Coastguard	Tempete	Foreman	Coastguard	Tempete
24dB	27.8	27.0	26.6	18.6	35.6	35.6
27dB	28.5	29.0	29.9	20.8	48.6	48.7
30dB	29.3	30.1	30.3	25.5	57.9	52.1

total of M_2 measurements are collected, which is calculated using (3.2) and bounded in the range $[0.3n^2, 0.9n^2]$. From Table 4.1, the sampling process can also be adaptively adjusted to the input PSNR lower bound, and the PSNR of different sequences are almost at the same level and generally follow the bounds.

Figure 4.1 plots the estimated number of measurements for each frame in Tempete and the corresponding PSNR of the reconstructed frames (without intra-frame measurement allocation). Here, the PSNR lower bound is set as 24dB. It shows that since the measurements are adjusted according to the quality constraint, the frame PSNR are generally consistent and vary in a small range of 1.3dB. However, blocks with different dynamic features are not evenly distributed among all frames, thus the number of measurements acquired changes from frame to frame. It demonstrates that intra-frame measurement allocation is required to ensure that each non-reference frame is assigned a fixed number of measurements so that the video can be played at a constant speed.

Furthermore, it is known that in raster scan or in 2-D compressed sensing mode, doubling the frame rate means a 100% increase in the number of measurements. However, using the proposed framework, enhancement of the frame rate does not necessarily result in a large increase of measurements to be collected. As an example, for the Tempete sequence, if the PSNR lower bound is set as 27dB, the estimated average number of measurement per second at 30 fps is approximately

1.03M measurements per second (mps); while that number at 60 fps is 1.23M mps and is only increased by 18.9%. This is because with a higher frame rate, the correlation between neighboring frames is higher, and there may be more static and small-change blocks in one frame, which helps us achieve higher sampling efficiency.

The simulation results in this section show that with a fixed frame rate, this work can correctly estimate the number of measurements required to satisfy the pre-determined quality constraints, which can help select the maximum achievable frame rate under the hardware sampling rate and the quality constraints.

4.2 Results for Adaptive Video Acquisition with Frame Rate Selection

This section gives simulation results of the whole video acquisition framework. To test the frame rate selection module, Foreman, Coastguard and Tempete, whose original frame rate is 30 fps, are temporally interpolated to 60 fps using the YUV Frame Rate Conversion software[1]. The newly obtained sequences are used in the simulations as the ground truth of the raster scan videos. The framework is tested on 60 fps videos to show that given a small number of measurements, the proposed framework can achieve very high frame rate (higher than the usual 30fps for most test sequences). Assuming there is no scene change, the first second is used to estimate the scene characteristics and to select a proper frame rate. If the selected frame rate is 60 fps, as an example, the proposed video acquisition framework is tested on the 2nd second of video with 60 frames (the first second is used for frame rate selection). If a lower frame rate is selected, this adaptive acquisition scheme is tested on the 2nd and the 3rd seconds of the video. The same trend is observed if

[1] Available at: http: //www.yuvsoft.com/technologies/frame_rate/

the framework is tested on longer sequences. The input lower bounds on the PSNR of the reconstructed frames are 24dB, 27dB and 30dB. The hardware sampling rate M_{max} is set as $256 \times 320 \times 30 \times 0.3$ measurements per second (mps), $256 \times 320 \times 30 \times 0.4$ mps and $256 \times 320 \times 30 \times 0.5$ mps, which correspond to frame rate 9 fps, 12 fps and 15 fps, respectively, if raster scan is used. Overall, there are 9 different scenarios to be tested for each sequence.

Table 4.2 shows the selected frame rates and the resulting average PSNR values under the quality and sampling rate constraints. It can be seen that a proper frame rate can be selected according to the scene characteristics, and the average PSNR of the reconstructed video acquired at the selected frame rate can meet the requirement. As expected, given the same quality and sampling rate constraints, sequence with slow motion and relatively simple scene composition, such as Foreman, can be acquired at a very high frame rate, and the highest frame rate enhancement ratio can be achieved is 60:9. With fast changing scenes and rich texture, sequences like Coastguard and Tempete are acquired at lower frame rates, and the frame rate enhancement ratio is also smaller. For each sequence, given a fixed PSNR lower bound, a larger M_{max} leads to either better output quality or a higher frame rate. In the simulations, the highest candidate frame rate is set as 60 fps. In some cases, for example, the acquisition of Foreman at 60 fps, the output PSNR is much higher than the pre-set lower bound, which suggests that an even higher frame rate is achievable if the range of frame rates to be searched is enlarged. When comparing the two reconstruction schemes, the min-TV provides better reconstruction quality especially for smooth sequences such as Foreman, while its running time is about 4 times higher than that of GPSR. Figure 4.2 shows an example of the reconstructed frame 3 of the Foreman sequence using GPSR and min-TV. By comparing Figure 4.2b and 4.2c, the reconstructed frame using min-TV has clearer appearance with less blocky and blurring effect, and Figure 4.2c is very closed to the original version in Figure 4.2a. In many applications, e.g., surveillance, GPSR can be used to first quickly reconstruct a low-quality version for a quick review, and min-TV can be used to obtain a high-quality version if detailed examination of the scene is necessary.

4.3 Parameter Selection

This framework involves several parameters that affect the system performance. The parameter selection for this framework comes from trial-and-error, and parameters are selected to achieve maximum frame rate under the quality constraint.

4.3.1 The Selection of C

C in (3.2) determines the number of measurements required for large-change blocks. If C is set too small or too big, an inappropriate frame rate will be selected, which may affect the reconstruction quality and the sampling efficiency of the framework. Table 4.3 shows simulation results on Coastguard and Tempete with different values of C, and similar trends are observed for other sequences. Data in bold are results with $C = 220$, the value used in this framework.

From Table 4.3, in the simulations on the Tempete sequence with PSNR lower bound 30dB, when a smaller $C = 180$ is selected, the proposed framework underestimates the number of measurements required to satisfy the quality constraint, and falsely selects a higher frame rate 30fps. Thus, the reconstructed sequence does not satisfy the perceptual quality constraint. On the contrary, when a larger $C = 260$ is selected in the simulations on the Coastguard sequence with PSNR lower bound 27dB, it overestimates the required number of measurements, and selects a lower frame rate of 30fps. This reduces the sampling efficiency of the frameworks, since a higher frame rate of 60fps is feasible (with $C = 220$), which still satisfies the quality constraint of 27dB. In addition, from Table 4.3, when the same frame rate is selected, the reconstruction quality is relatively insensitive to the values of C. From the simulation results, $C = 220$ is an appropriate choice to achieve maximum frame rate under the quality constraint.

4.3.2 The Selection of Tolerance Level

In Section 3.6, a tolerance level of 2% is introduced in the frame selection module. Same as the parameter C, the tolerance level may change the selected frame rate, which affects the reconstruction quality and the sampling efficiency of this framework. With a smaller tolerance level, a smaller frame rate may be selected, which may reduce the sampling efficiency of the framework; while a larger tolerance level may result in the selection of a higher frame rate, which may make the reconstructed sequence fail to satisfy the quality constraint. From the simulations, a tolerance level of 2% achieves maximum frame rate under the quality constraint, and is used in this work.

4.3.3 The Selection of T_1 and T_2

The performance of this framework is tested with different combinations of T_1 and T_2, and the results are shown in Table 4.4. Coastguard is used as an example, and similar trends are observed for other sequences. The PSNR lower bound is set as 27dB, and the hardware sampling rate is $256 \times 320 \times 30 \times 0.38$ measurements per second. Data in bold are the selected frame rate and the resulting PSNR with $T_1 = 4$ and $T_2 = 9.5$, which are the values used in this framework.

From Table 4.4, if T_1 and T_2 are set too small or too large, an inappropriate frame rate will be selected, which may affect the reconstruction quality and the sampling efficiency of the proposed framework. For example, with $T_1 = 2$ and $T_2 = 7$, this framework overestimates the number of measurements required to satisfy the quality constraint, and selects a lower frame rate of 15fps. This decreases the sampling efficiency of the framework, since a higher frame rate of 30fps is achievable (with $T_1 = 4$ and $T_2 = 9.5$), which still satisfies the quality constraint of 27dB. When larger $T_1 = 6$ and $T_2 = 11$ are used, this framework underestimates the required number of measurements, and falsely determines the maximum achievable frame rate to be 60fps. Thus, the reconstruction quality is below the PSNR lower bound 27dB. In addition, from Table 4.4, as long as the same frame rate is selected, the reconstruction quality is relatively insensitive to T_1 and T_2. From the simulations, $T_1 = 4$ and $T_2 = 9.5$ achieve the maximum frame rate under the quality constraint, and are used in this work.

4.3.4 The Selection of M_0 and M_1

Same as T_1 and T_2, M_0 and M_1 affect the frame rate selection, which may affect the reconstruction quality and the sampling efficiency of the proposed framework. M_0 and M_1 in this work are selected to maximize the frame rate under the quality constraint, and the simulation results show that $(M_0 = 0.03n^2, M_1 = 0.1n^2)$ are appropriate and are used in this work.

4.4 Further Discussions

This section further discusses the design and performance of this proposed framework.

4.4.1 The Analysis of GOP Size

In this framework, indicator frames are inserted periodically to estimate block sparsity for large-change blocks. They divide the whole sequence into group of pictures (GOP), where all frames between two neighboring indicator frames are in one GOP.

Here the influence of GOP size are studied, and equivalently, the frequency of indictor frames, on the output quality. Figure 4.3 plots the average PSNR of the reconstructed video sequences as the GOP size increases. The frame rate f_{cs} is fixed as 30 fps. The simulation is conducted on five sequences including Akiyo, Coastguard, Flower, Foreman and Tempete, who have different dynamics and spatial features. Similar trends are observed for other sequences. Each sequence corresponds to a single scene without scene change. As an example, the PSNR lower bounds are set as 30dB, 27dB, 27dB, 27dB, 24dB for Akiyo, Coastguard, Flower, Foreman, Tempete, respectively, and the maximum sampling rate M_{max} are set as $256 \times 320 \times 30 \times 0.1$ mps, $256 \times 320 \times 30 \times 0.4$ mps, $256 \times 320 \times 30 \times 0.4$ mps, $256 \times 320 \times 30 \times 0.2$ mps, $256 \times 320 \times 30 \times 0.35$ mps for Akiyo, Coastguard, Flower, Foreman, Tempete, respectively. The same trend is observed for other values of the PSNR lower bound and frame rates.

From Figure 4.3, the GOP size does not have much influence on the output quality, as long as there is no scene change. This is because for each large-change block, the corresponding indicator block can be correctly identified by searching in a 3×3 neighboring region in the indicator frame, which increases the estimation accuracy of block sparsity. Also, according to (3.8), the number of measurements assigned to a large-change block i is determined by the relative ratio of its estimated sparsity $\tilde{S}_i$ to the summation of the block sparsity of all large-change blocks in the frame, rather than the absolute value of $\tilde{S}_i$. This makes the measurement allocation result, and thus the performance of the proposed framework, less sensitive to the inaccurate estimation of the block sparsity caused by a large GOP size. Therefore, in this framework, for one scene, only one indicator frame is used, which is also the reference frame, to reduce the computation cost to reconstruct the indicator frames on the fly.

4.4.2 The Selection of the Block Size

In the simulations, it is observed that the choice of block size n affects the quality of the reconstructed videos given the same hardware sampling rate M_{max}. To show this effect, this test reconstructs the first frame of Coastguard using different block size, and the same trend is observed for other sequences. Since frames should be divided into an integer number of $n \times n$ blocks, different frame sizes are used accordingly: for $n = 32, 48, 56, 64, 72$, the frame sizes are as 288×352, 288×336, 280×336, 256×320 and 288×288, respectively. For completeness, compressed sensing is also applied to the whole frame of size 288×352, which is equivalent to considering the frame as a single large block. For fair comparison of different block sizes, it only considers the 256×288 common area shared by different frame sizes (corresponding to different blocks sizes) and calculates its PSNR. The sampling ratio is fixed as 40% for all blocks. From Table 4.5, given the same number of measurements, a $1.4 \sim 2.4$dB increase in PSNR is observed when the block size n increases from 56 to 64, which suggests that there is a constraint on the block size and it should not be too small. This improvement in the perceptual quality is probably related to randomness of the pixels. It is known that randomness plays an important role in compressed sensing. For example, the SBHE operator randomly permutes the signal to be sensed before applying the partial block Hadamard transform. Therefore, a large block size is expected to provide more randomness through permutation and gives better quality. The block size used in the simulations is 64×64, and the corresponding perceptual quality is in the same level as that when compressed sensing is applied to the whole frame.

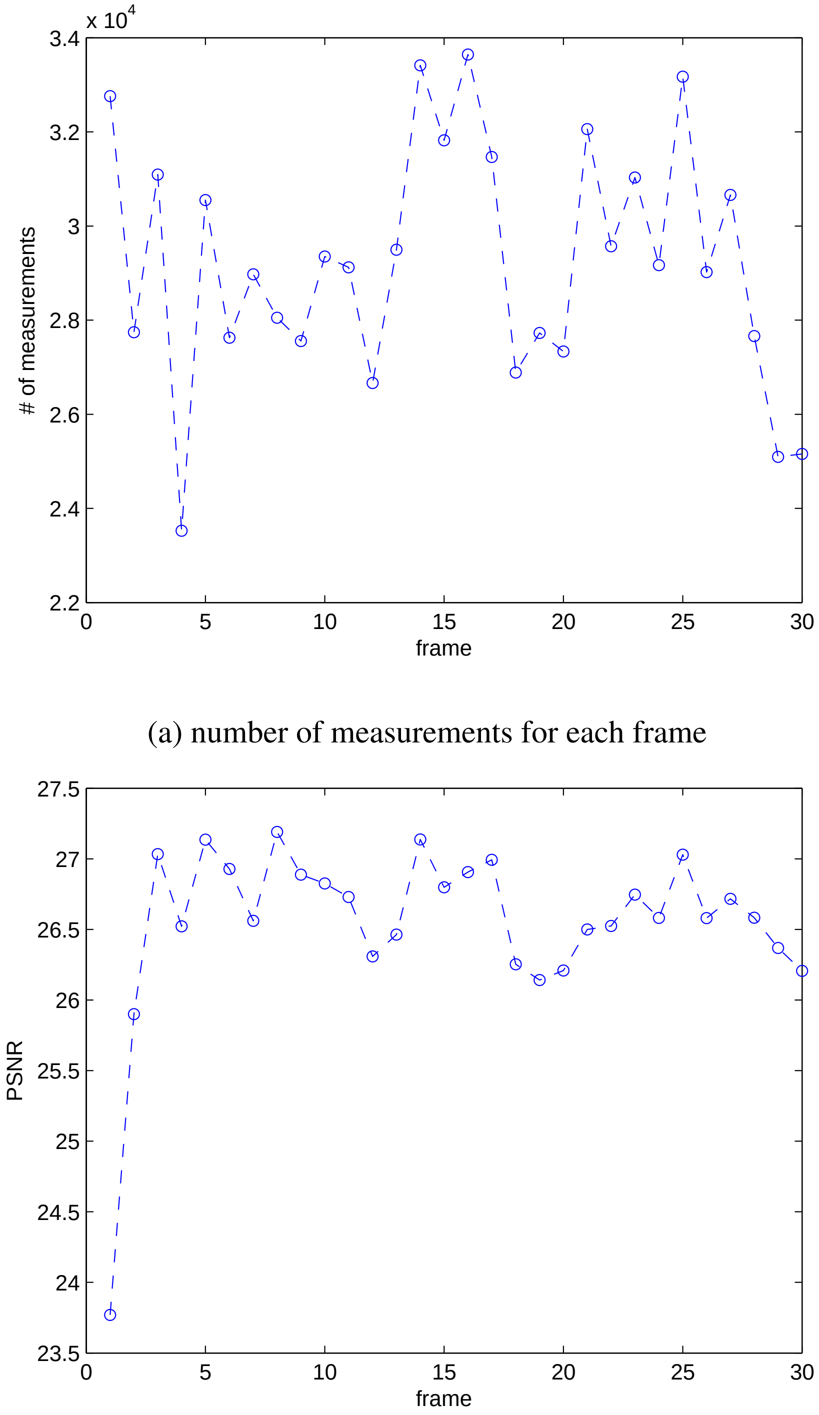
4.4.3 The Quantization Effect

For most practical applications, quantization of the collected measurements is necessary to reduce the number of bits required to represent the data. The effect of quantization of compressed sensing measurements is examined in the simulations. The steps in [59] are followed: each entry $\mathbf{y}(i)$ in the measurement vector $\mathbf{y}$ is quantized using a uniform scalar quantizer with b bits. Then the sequence is reconstructed based on the quantized measurements $\mathbf{y}^q$. This simulation starts from 8 bits and reduce 1 bit every time to examine the quantization effect on two sequences with different dynamics features and texture complexity. Figure 4.4 shows the averaged PSNR over 60 frames with respect to the bit length. Figure 4.4a shows the simulation results on Foreman with 60 fps. The PSNR lower bound set as 27dB and M_{max} is $256 \times 320 \times 30 \times 0.3$ mps. Figure 4.4b is for the Tempete sequence with frame rate 30 fps. The PSNR lower bound is 30dB and $M_{max} = 256 \times 320 \times 30 \times 0.5$ mps. The same trend is observed for other sequences and values of the parameters. The solid line in each figure represents the PSNR of the reconstructed sequences without quantization, which is used as a benchmark. The dashed curves show that the reconstruction quality degrades rapidly when $b < 6$.

4.5 Performance Comparison with Other Video Acquisition Systems using Compressed Sensing

The performance of the proposed framework is also compared with the independent 2-D compressed sensing method [53], the frame-difference method in [39] and the 3-D wavelet-based method in [53]. The same number of measurements M_{all} are used for all four algorithms and GPSR is used to reconstruct the video sequences for fair comparison. One second's sequences at 60 fps (60 frames in total) are tested. In the proposed framework, frame 1 is the reference frame and $0.4n^2$ measurements are collected for each block in frame 1. The rest measurements are evenly distributed among the remaining 59 frames. For the independent 2-D schemes, each frame is assigned $M_{all}/60$ measurements, and regular compressed sensing for image is applied to each frame independently. For the frame-difference method, longer sequence suffers more from alias accumulation effect. Thus the 60-frame sequences are divided to three 20-frame groups that are reconstructed separately, and the first frame in each groups is used as the reference frame. Here, $0.4n^2$ measurements are collected for each block in the reference frames (frame 1, 21 and 41), and the rest measurements are evenly distributed among the remaining 57 frames. For the 3-D method, $M_{all}/60$ measurements are collected for each frame. Note that this scheme treats the whole sequence as a single signal and reconstructs all frames in the sequence simultaneously, which demands a large memory and incurs high computation cost. To address this issue, the test sequence is also divided into 3 independent groups, where all 20 frames in a group are reconstructed simultaneously.

Table 4.6 lists the simulation results, and shows that the proposed method gives better reconstruction quality. The visual quality of the reconstruction results is also compared, and Figure 4.5 shows an example of the reconstructed Frame 10 in the Tempete sequence, which also demonstrates better performance of the adaptive framework. Compared to the independent 2-D compressed sensing scheme, the proposed method explores the temporal correlation between neighboring frames to improve sampling efficiency, and therefore, is able to achieve higher quality. Compared to the frame-difference scheme, this adaptive sampling attenuates the alias accumulation (error propagation), especially for fast motion sequences, and reduces the difference between the maximum and the minimum PSNR from $4.5 \sim 12.2$dB to around 3dB. The 3-D wavelet-based algorithm fails or gives poor reconstruction results when the sampling ratio is low. This is because it is difficult to reconstruct large-scale ($256 \times 320 \times 20$) signals when the sampling ration is low, e.g., in the above simulations on the Foreman sequence. What is more, compared to the 3-D wavelet-based algorithm, this block-based method reduces more than half the running time to reconstruct the sequence, since the block-by-block reconstruction is easier and faster to compute than the simultaneous reconstruction of the whole sequence. Figure 4.6 plots the PSNR (dB) versus the total number of measurements for different methods. In Figure 4.6, 1 second's Tempete sequence at 60fps is tested and GPSR is used to reconstruct the video sequences. Figure 4.6 also demonstrates that the proposed method gives better reconstruction quality than prior works.



(a) number of measurements for each frame

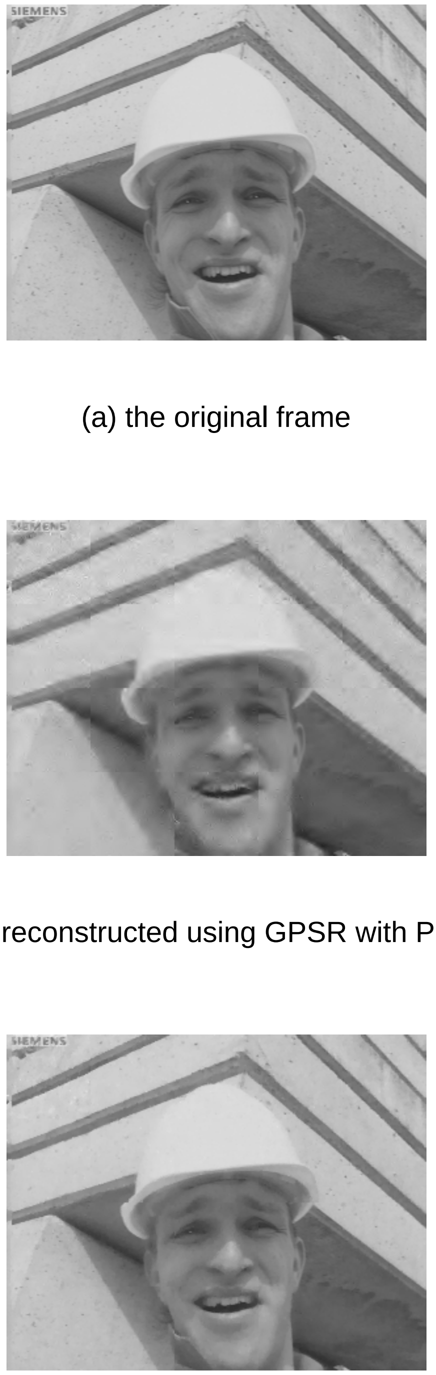
(b) PSNR of each reconstructed frame

Fig. 4.1. Variations in PSNR and the estimated number of measurements per frame for Tempete.

TABLE 4.2

Performance of adaptive compressed sensing of video with frame rate selection

PSNR bound	Measurements/sec	Sequence	GPSR: PSNR	min-TV: PSNR	f_{cs}	C_{fr}
24dB	$256 \times 320 \times 30 \times 0.3$ (f_{rs}=9fps)	Foreman	28.9	31.2	60fps	6.7
		Coastguard	26.2	29.0	30fps	3.3
		Tempete	28.8	30.2	20fps	2.2
	$256 \times 320 \times 30 \times 0.4$ (f_{rs}=12fps)	Foreman	29.5	32.1	60fps	5.0
		Coastguard	25.7	28.0	60fps	5.0
		Tempete	26.9	28.3	60fps	5.0
	$256 \times 320 \times 30 \times 0.5$ (f_{rs}=15fps)	Foreman	29.9	34.6	60fps	4.0
		Coastguard	27.2	28.9	60fps	4.0
		Tempete	27.9	29.0	60fps	4.0
27dB	$256 \times 320 \times 30 \times 0.3$ (f_{rs}=9fps)	Foreman	28.9	31.0	60fps	6.7
		Coastguard	31.0	33.6	15fps	1.7
		Tempete	30.3	31.8	15fps	1.7
	$256 \times 320 \times 30 \times 0.4$ (f_{rs}=12fps)	Foreman	29.5	31.6	60fps	5.0
		Coastguard	28.2	30.5	30fps	2.5
		Tempete	30.6	31.8	20fps	1.7
	$256 \times 320 \times 30 \times 0.5$ (f_{rs}=15fps)	Foreman	30.4	33.3	60fps	4.0
		Coastguard	27.2	28.9	60fps	4.0
		Tempete	28.1	29.0	60fps	4.0
30dB	$256 \times 320 \times 30 \times 0.3$ (f_{rs}=9fps)	Foreman	28.7	30.9	60fps	6.7
		Coastguard	33.7	36.5	12fps	1.3
		Tempete	32.5	34.2	12fps	1.3
	$256 \times 320 \times 30 \times 0.4$ (f_{rs}=12fps)	Foreman	29.3	31.1	60fps	5.0
		Coastguard	31.2	33.4	20fps	1.7
		Tempete	30.6	31.7	20fps	1.7
	$256 \times 320 \times 30 \times 0.5$ (f_{rs}=15fps)	Foreman	30.3	32.1	60fps	4.0
		Coastguard	29.7	31.7	30fps	2.0
		Tempete	30.1	30.9	30fps	2.0



(a) the original frame

(b) the frame reconstructed using GPSR with PSNR=30.0dB

(c) the frame reconstructed using min−TV with PSNR=33.8dB

Fig. 4.2. The reconstructed frame 3 of the Foreman sequence (60 fps) using 11144 (13.6% × 256 × 320) measurements.

TABLE 4.3

The selection of C in (3.2)

PSNR lower bound	Measurements/sec	Sequence	C	average PSNR (GPSR)	f_{cs}
30dB	$256 \times 320 \times 30 \times 0.45$ (f_{rs}=13.5fps)	Tempete	180	29.2dB	30fps
			200	31.6dB	20fps
			220	**31.4**dB	**20fps**
27dB	$256 \times 320 \times 30 \times 0.5$ (f_{rs}=15fps)	Coastguard	**220**	**27.2**dB	**60fps**
			240	29.7dB	30fps
			260	29.7dB	30fps

TABLE 4.4

selection of T_1 and T_2.

	T_1=2, T_2=7	$\mathbf{T_1 = 4, T_2 = 9.5}$	T_1=3, T_2=11	T_1=6, T_2=11
frame rate (fps)/PSNR (dB)	15/34.2	$\mathbf{30/27.8}$	30/27.8	60/26.0

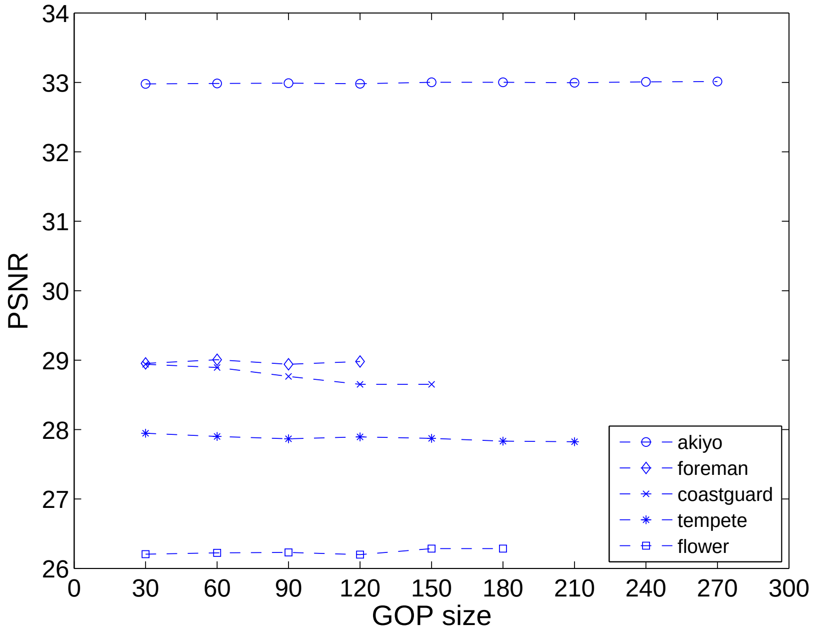


Fig. 4.3. The influence of GOP size on the video quality.

TABLE 4.5

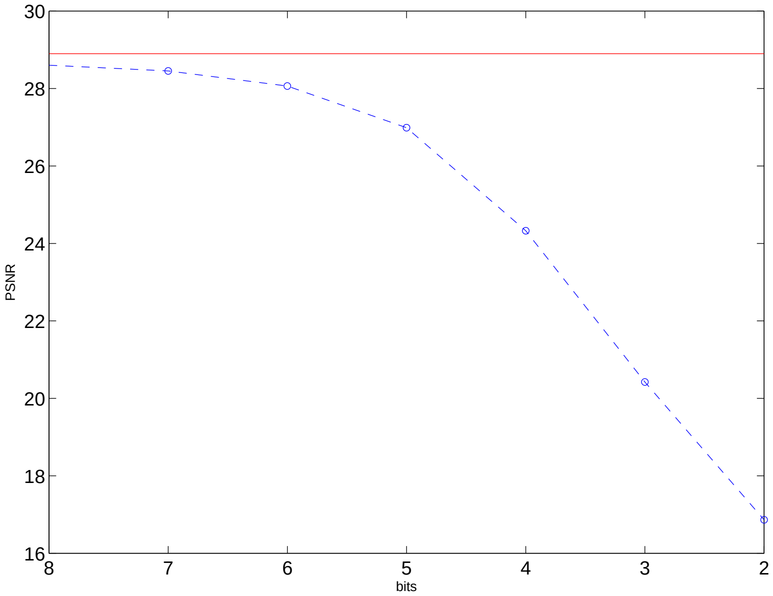
The influence of block size on the reconstruction quality

	PSNR (dB)					
Sequence	n=32	n=48	n=56	n=64	n=72	whole frame
Foreman	28.8	29.3	29.2	31.1	31.2	32.0
Coastguard	24.1	24.9	24.6	27.0	26.9	27.8
Tempete	21.6	22.1	22.1	23.5	23.3	24.8

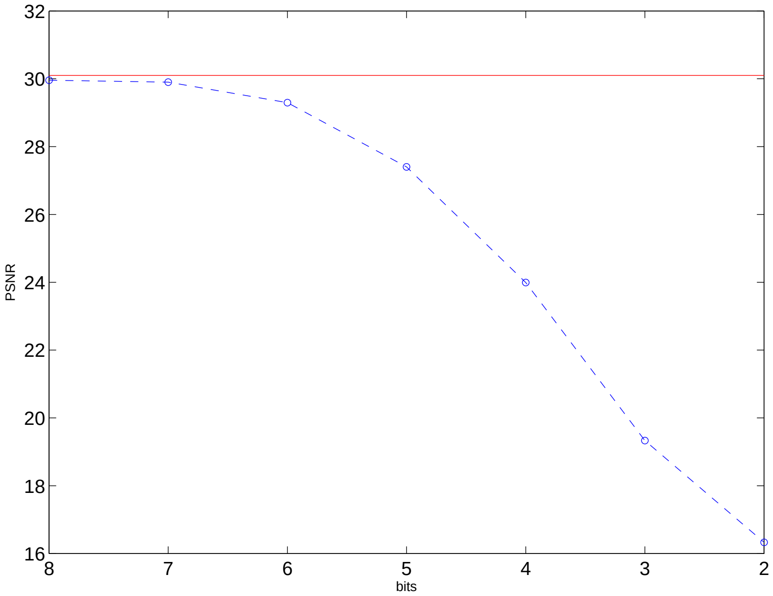
TABLE 4.6

Performance comparison

			average PSNR/(max-min) PSNR (dB)			
Sequence	M_{all}	f_{cs}	Adaptive	Independent 2D	Frame difference	3-D wavelet
Foreman	757752	60fps	28.9/4.2	27.4/1.4	25.7/12.4	6.4/0.2
Coastguard	1241080	60fps	27.2/2.5	25.5/0.8	22.8/8.5	21.3/0.9
Tempete	1241080	60fps	27.9/3.3	23.3/0.7	23.6/4.5	20.1/0.6

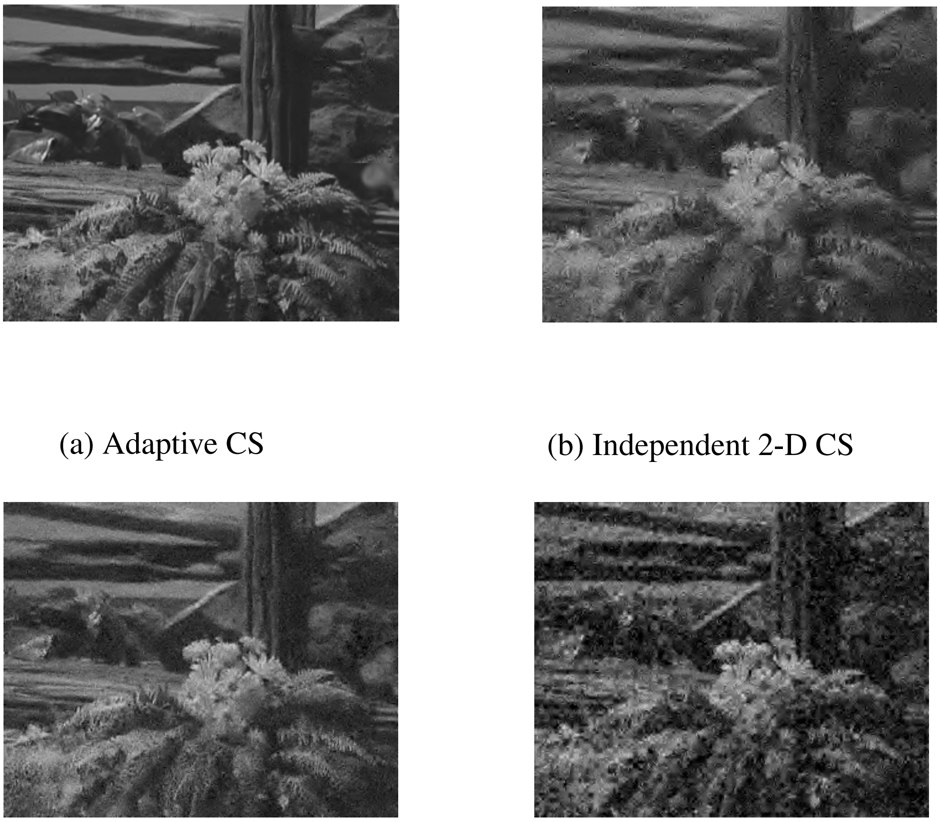


(a) The reconstruction of Foreman



(b) The reconstruction of Tempete

Fig. 4.4. The reconstruction quality with respect to bit length.



(a) Adaptive CS

(b) Independent 2-D CS

(c) Frame-difference CS

(d) 3-D wavelet-based CS

Fig. 4.5. Reconstruction results of Frame 10 in the Tempete test sequence.

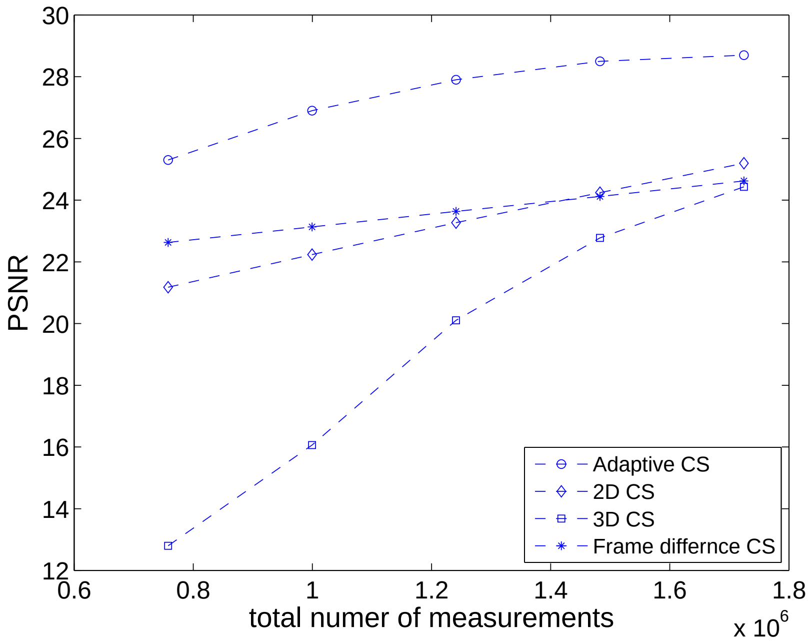


Fig. 4.6. The average PSNR versus total number of measurements for different methods.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This thesis proposes an adaptive block-based framework for compressed video sampling to maximize the frame rate under the hardware sampling rate and the quality constraints. It also analyzes the performance of the proposed framework and demonstrates its superior performance when compared to existing works.

The proposed framework estimates the inter-frame correlation between co-located blocks in neighboring frames based on partial measurements, and classifies blocks into different types. It then adjusts the sampling and reconstruction strategy for each block according to its block type and its estimated block sparsity that reflects the texture complexity of the corresponding region. During the acquisition process, for each frame, the intra-frame measurement allocation module strategically allocates measurements among blocks according to block features to achieve the best overall quality. The proposed framework also includes a frame rate selection module that selects the maximum achievable frame rate under the hardware sampling rate and the quality constraints. For each candidate frame rate, it estimates the average number of measurements per second that are required to satisfy the quality constraint, and compares with the measurement budget to determine whether the candidate frame rate is achievable. The maximum achievable frame rate is then selected.

The simulation results show that the proposed framework can effectively adjust the sampling strategy according to the motion and the complexity of the scene. It maximizes the frame rate under the hardware sampling rate and the quality constraint, and achieves a frame rate enhancement ratio of $1.3 \sim 6.7$ when compared to the traditional raster scan method. It also brings a $1.5 \sim 7.8$dB gain in the average PSNR of the reconstructed frames when compared with prior works.

5.2 Future Works

There are still some aspects can be further investigated to improve the performance or extend the framework.

Like most compressed sensing schemes, this framework assumes ideal data sampling, that is, all collected measurements are not corrupted by noise during the acquisition or transmission process. Since ideal sampling is not possible in real applications, it is desirable to conduct a careful examination of the impact of noise on the system performance.

In addition, this acquisition framework only takes the luminance component of a video as an example. Even though it can be easily extended to color video acquisition by considering each color channel independently, this direct extension is sub-optimal as it does not explore the correlation across the color bands. By jointly considering different components in the color space, the sampling efficiency for color videos would be improved.

This work has focused on compressed video sampling and reconstruction. For applications such as video surveillance and medical diagnosis, the obtained video signal is used to make a detection or classification decision. Tasks such as detection do not require a reconstruction of the original signal. Some pattern recognition algorithms directly based on compressed sensing measurements of an image have been proposed [60–62], and it is desirable to extend those approaches to videos. In video compressed sensing, measurements for neighboring frames are correlated, and can be jointly considered to increase the recognition accuracy and to track the movement of objects.

References

[1] D. Takhar, J. Laska, M. Wakin, M. Duarte, D. Baron, S. Sarvotham, K. Kelly, and R. Baraniuk, “A new compressive imaging camera architecture using optical-domain compression,” *Proc. Computational Imaging IV*, vol. 6065, pp. 43–52, Jan. 2006.

[2] H. Zettl, “Video basics,” *Cengage Learning*, 2009.

[3] Y. Wang, J. Ostermann, and Y. Zhang, “Video processing and communications,” *Prentice-Hall, New Jersey*, 2002.

[4] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the h.264/avc video coding standard,” *IEEE Trans. Circ. Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, July 2003.

[5] J. L. Mitchell, W. B. Pennebaker, C. Fogg, and D. J. LeGall, “Mpeg video compression standard,” *Chapman and Hall, New York, USA*, 1997.

[6] V. Bastani, M. S. Helfroush, and K. Kasiri, “Image compression based on spatial redundancy removal and image inpainting,” *Journal of Zhejiang University-SCIENCE C (Computers and Electronics)*, vol. 11, no. 2, pp. 92–100, Feb. 2010.

[7] D. L. Donoho, “Compressed sensing,” *IEEE Trans. Info. Theory*, vol. 52, pp. 1289–1306, Sept. 2006.

[8] E. J. Candes and T. Tao, “Near optimal signal recovery from random projections: Universal encoding strategies?” *IEEE Trans. Info. Theory*, vol. 52, pp. 5406–5425, Dec. 2006.

[9] E. J. Candes, “Compressive sampling,” *Proc. Int. Cong. Mathematicians*, vol. 3, pp. 1433–1452, 2006.

[10] R. G. Baraniuk, “Compressive sensing,” *IEEE Signal Proc. Mag.*, vol. 24, no. 4, pp. 118–121, 2007.

[11] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, K. F. K. T. Sun, and R. G. Baraniuk, “Single pixel imaging via compressive sampling,” *IEEE Signal Proc. Mag.*, vol. 25, no. 2, pp. 83–91, March 2008.

[12] W. Chan, K. Charan, D. Takhar, K. F. Kelly, R. G. Baraniuk, and D. M. Mittleman, “A single-pixel terahertz imaging system based on compressed sensing,” *Applied Physics Letters*, vol. 93, no. 12, p. 121105, March 2008.

[13] W. Chan, M. Moravec, R. Baraniuk, and D. Mittleman, “Terahertz imaging with compressed-sensing and phase retrieval,” *Applied Physics Letters*, vol. 33, no. 9, pp. 974–976, May 2008.

[14] X. Wu and X. Zhang, “Compressive-uniform hybrid sensing for image acquisition and communication,” *IEEE Int. Conf. on Image Proc. (ICIP)*, pp. 3041–3044, 2009.

[15] S. Borman and R. L. Stevenson, “Super-resolution from image sequences-a review,” *Proc 1998 Midwest Symp Circuits and Systems*, no. 5, pp. 374–378, April 1998.

[16] C. P. Sung, K. P. Min, and G. K. Moon, “Super-resolution image reconstruction: A technical overview,” *IEEE Signal Proc. Mag.*, vol. 20, no. 3, pp. 21–36, 2003.

[17] P. Maraghechi, C. Straatsma, Z. Liu, V. Zhao, and A. Y. Elezzabi, “Plasmon-assisted terahertz imaging inside metal-filled media,” *Opt. Express*, vol. 17, no. 19, pp. 16 456–16 464, 2009.

[18] W. T. Freeman, T. R. Jones, and E. C. Pasztor, “Example-based super-resolution,” *IEEE Comput. Graph. Appli.*, pp. 56–65, 2002.

[19] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, “Fast and robust multiframe super resolution,” *IEEE Trans. Image Processing*, vol. 13, no. 10, pp. 1327–1344, Oct. 2004.

[20] I. Drori, “Compressed video sensing,” *BMVA symposium:3D Video-Analysis, Display and Applications*, 2008.

[21] J. Romberg, “Imaging via compressive sampling,” *IEEE Signal Proc. Mag.*, vol. 25, no. 2, pp. 14–20, March 2008.

[22] L. Gan, “Block compressed sensing of natural images,” *Proc. Int. Conf. on Digital Signal Processing (DSP)*, July 2007.

[23] M. Lustig, D. Donoho, and J. Pauly, “Sparse mri: The application of compressed sensing for rapid mr imaging,” *Magn. Resonance Med*, vol. 58, no. 6, pp. 1182–1195, Dec 2007.

[24] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk, “An architecture for compressive imaging,” *IEEE Int. Conf. on Image Proc. (ICIP)*, pp. 1273–1276, 2006.

[25] L. Gan, T. T. Do, and T. D. Tran, “Fast compressive imaging using scrambled hadamard ensemble,” *European Signal Proc. Conf. (EUSIPCO)*, 2008.

[26] E. Candes and J. Romberg, “Sparsity and incoherence in compressive sampling,” *Inverse Prob*, vol. 23, no. 3, pp. 969–985, Sept. 2007.

[27] E. J. Candes and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Proc. Mag.*, vol. 25, no. 2, pp. 21–30, March 2008.

[28] R. Coifman, F. Geshwind, and Y. Meyer, “Noiselets,” *Appl. Comp. Harmon. Anal*, vol. 10, no. 1, pp. 27–44, 2001.

[29] E. Candes and J. Romberg, “Robust signal recovery from incomplete observations,” *IEEE Int. Conf. on Image Proc. (ICIP)*, pp. 1281–1284, 2006.

[30] E. Candes and T. Taot, “Decoding by linear programming,” *IEEE Trans. Inform. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[31] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *IEEE Trans. Info. Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[32] D. L. Donoho, Y. Tsaig, I. Drori, and J. L. Starck, “Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit,” *IEEE Trans. Info. Theory*, Submitted 2007.

[33] D. Needell and R. Vershynin, “Signal recovery from incomplete and inacurate measurements via regularized orthogonal matching pursuit,” *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 310–316, Apr. 2009.

[34] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, “Bregman iterative algorithms for l_1-minimization with applications to compressed sensing,” *SIAM Journal on Imaging Sciences*, vol. 1, no. 1, pp. 143–168, Mar. 2008.

[35] T. Blumensath and M. Davies, “Iterative hard thresholding for compressed sensing,” *Appl. Comput. Harmonic Anal.*, vol. 27, no. 3, pp. 265–274, 2009.

[36] D. Needell and J. A. Tropp., “Cosamp: Iterative signal recovery from incomplete and inaccurate samples,” *Proc. ACM Symp. Theory of Computing*, vol. 26, pp. 301–321, 2008.

[37] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, “Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems,” *IEEE J. of Sel. Topics in Signal Proc.*, vol. 1, no. 4, pp. 586–598, 2007.

[38] E. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Info. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[39] J. Zheng and E. L. Jacobs, “Video compressive sensing using spatial domain sparsity,” *Optical Engineering*, vol. 48, no. 8, pp. 087 006–1–10, August 2009.

[40] M. Lustig, D. Donoho, J. Santos, and J. Pauly, “Compressed sensing mri,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 72–82, March 2008.

[41] M. N. Do and M. Vetterli, “The contourlet transform: an efficient directional multiresolution image representation,” *IEEE Trans. Image Processing*, vol. 14, no. 12, pp. 2091–2106, 2005.

[42] D. C. Arthur, J. P. Zhou, and M. N. Do, “The contourlet transform: an efficient directional multiresolution image representation,” *IEEE Trans. Image Processing*, vol. 15, no. 10, pp. 3089–3101, 2006.

[43] X. Qu, D. Guo, Z. Chen, and C. Cai, “Compressed sensing mri based on nonsubsampled contourlet transform,” *IEEE Int. Symposium on IT in Medicine and Education*, pp. 693–696, 2008.

[44] X. Qu, X. Cao, D. Guo, C. Hu, and Z. Chen, “Compressed sensing mri with combined sparsifying transforms and smoothed l_0 norm minimization,” *IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP)*, pp. 626–629, Mar. 2010.

[45] R. Chartrand, “Fast algorithms for nonconvex compressive sensing: Mri reconstruction from very few data,” *IEEE Int. Symposium on Biomedical Imaging (ISBI)*, pp. 262–265, Jun. 2009.

[46] ——, “Exact reconstruction of sparse signals via nonconvex minimization,” *IEEE Signal Process. Lett.*, vol. 14, no. 10, pp. 707–710, Oct. 2007.

[47] R. Chartrand and V. Staneva, “Restricted isometry properties and nonconvex compressive sensing,” *Inverse Problems*, vol. 24, no. 3, pp. 1–14, June 2008.

[48] E. Candes and J. Romberg, “l_1-magic: Recovery of sparse signals via convex programming,” http://www.acm.caltech.edu/l1magic/, 2005.

[49] S. Ma, W. Yin, Y. Zhang, and A. Chakraborty, "An efficient algorithm for compressed mr imaging using total variation and wavelets," *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, Jun. 2008.

[50] G. H. Chen, J. Tang, and S. Leng, "Prior image constrained compressed sensing (piccs): a method to accurately reconstruct dynamic ct images from highly undersampled projection data sets," *Med. Phys.*, vol. 35, pp. 660–663, 2008.

[51] J. C. Ye, "Compressed sensing shape estimation of star-shaped objects in fourier imaging," *IEEE Signal Processing Letters*, vol. 14, no. 10, pp. 750–753, Oct. 2007.

[52] A. S. Lewis and G. Knowles, "Video compression using 3d wavelet transforms," *Electronics Letters*, vol. 26, no. 6, pp. 396–398, March 1990.

[53] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takahar, K. Kelly, and R. G. Baraniuk, "Compressive imaging for video representation and coding," *Proc. Picture Coding Symp (PCS)*, April 2006.

[54] R. F. Marcia and R. M. Willett, "Compressive coded aperture video reconstruction," *Proc. European Signal Proc. Conf. (EUSIPCO)*, 2008.

[55] V. Stankovic, L. Stankovic, and S. Cheng, "Compressive video sampling," *Proc. European Signal Proc. Conf. (EUSIPCO)*, August 2008.

[56] X. Wu and X. Zhang, "Model-guided adaptive recovery of compressive sensing," *IEEE Proc. Data Compression Conf.*, pp. 123–132, 2009.

[57] J. Jain and A. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun*, vol. 29, no. 12, pp. 1799–1808, Dec 1981.

[58] J. Y. Park and M. B. Wakin, "A multiscale framework for compressive sensing of video," *Proc. Picture Coding Symposium (PCS)*, May 2009.

[59] E. Candes and J. Romberg, "Encoding the l_p ball from limited measurements," *Data Compression Conference*, pp. 33–42, March 2006.

[60] M. Duarte, M. Davenport, M. Wakin, and R. Baraniuk, "Sparse signal detection from incoherent projections," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 305–308, May 2006.

[61] M. Davenport, M. Wakin, and R. Baraniuk, "Detection and estimation with compressive measurements," *Tech. Rep. TREE0610, Rice University ECE Department*, 2006.

[62] M. Duarte, M. Davenport, M.Wakin, J. Laska, D. Takhar, K. Kelly, and R. Baraniuk, "Multiscale random projections for compressive classification," *Proc. IEEE Int. Conf. Image Process. (ICIP)*, pp. 161–164, Sep. 2007.