Compressive Video Acquisition with Multiple Sensors and No Reference Frames

by

Michelle L. Parenteau

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
Signal and Image Processing

Department of Electrical and Computer Engineering

University of Alberta

Abstract

Standard large-sensor-array-based camera designs are uneconomical when imaging with exotic wavelengths that require expensive photodetectors. The single-pixel camera allows image acquisition with only one sensor; however, its compressive sampling rate is too low to reliably acquire video signals. Hence, we consider the problem of compressive video acquisition with just a few sensors. We propose a block-based framework featuring two sensor modes that does not require the periodic collection of reference frames. We use a joint-sparse signal model to exploit temporal correlations in the video signal, and we explore the behaviour of our framework under different sampling conditions.

Acronyms

DMD	digital micromirror device
RIP	restricted isometry property
SBHE	scrambled block-Hadamard ensemble
GPSR	gradient projection for sparse reconstruction
OMP	orthogonal matching pursuit
CoSaMP	compressive sampling matching pursuit
StOMP	stagewise orthogonal matching pursuit
SOMP	simultaneous orthogonal matching pursuit
MSE	mean squared error
PSNR	peak signal-to-noise ratio
SSIM	structural similarity index

List of Symbols

f_{dmd}	micromirror switching frequency of the DMD
f_r	video frame rate
S	number of sensors
F	number of frames
B	number of blocks in a frame
L	block side length
N	target signal length
ζ	the set of all sensor indices
β	the set of all block indices
ξ_b	the set of all frame group indices for block b
Γ_g	the set of frame indices for frame group g
f_g	frame index of the last frame in frame group g
$\mathbf{x}_{(f,b)}$	target image signal corresponding to the fth frame and bth block
$\boldsymbol{\alpha}_{(f,b)}$	sparse transformed target signal corresponding to the fth frame and bth block
$\boldsymbol{\alpha}_g^C$	sparse common component shared by a frame group
$\boldsymbol{\alpha}_f^Q$	sparse innovation component of frame f
K_g	sparsity of the common component shared by frame group g
K_f	sparsity of the innovation component of frame f

$\hat{K}_b$	sparsity estimate of block b
$\hat{K}_\beta$	sum of sparsity estimates for all blocks
$\mathbf{y}_{(s,f,b)}$	measurement vector associated with the sth sensor, fth frame, and bth block
$\mathbf{y}_{0(s,f,b)}$	pre-sample vector associated with the sth sensor, fth frame, and bth block
$d_{0(s,f,b)}$	inter-frame pre-sample difference of the the sth sensor, fth frame, and bth block
$\mathbf{y}_{\mathsf{stack}(f,b)}$	stack of measurements in frame f and block b across all sensors
$\mathbf{y}_{\mathsf{buff}(b)}$	buffer of measurements for block b in the central hub during acquisition
M_{frame}	total number of measurements per frame
M_{sensor}	total number of measurements per sensor per frame
M_{init}	number of measurements collected per sensor for each block in the first frame
M_0	number of measurements collected per sensor for each block while pre-sampling
M_t	number of measurements allocated to a block by a texture-focused sensor
M_m	number of measurements allocated to a block by a motion-focused sensor
$M_{(s,f,b)}$	number of measurements allocated to the sth sensor, fth frame, and bth block
M_{total}	total accumulated measurements for a block so far in the frame grouping module
M_g	total number of measurements in a frame group
Φ_s	compressive measurement matrix associated with the sth sensor
$\Phi_{s(1:M)}$	the first M rows of Φ_s
Ψ	expansion (synthesis) basis
$\tilde{\Psi}$	sparsifying (analysis) basis
N_t	number of texture-focused sensors
N_m	number of motion-focused sensors
$\mathsf{length}\,(\cdot)$	number of elements in a vector

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

The famous Nyquist-Shannon-Kotelnikov sampling theorem [1–4] states that if a signal is sampled at at least twice the rate of its highest frequency component, then it can be accurately and reliably reconstructed from its samples. This condition is sufficient, but it's not strictly necessary—in fact, engineering applications have been successfully violating the traditional sampling theorem since the 1970s [5–8]. In traditional sampling, we assume the signal is frequency-dense; i.e., it contains frequencies everywhere within its bandwidth. However, large classes of signals are actually or approximately frequency-sparse and can be recovered from fewer measurements than traditional wisdom would prescribe.

The term "compressed sensing" was first used by Donoho in 2006 [9] following a series of foundational papers by Candès and Tao in abstract harmonic analysis [10–13]. They focused on accurate recovery of sparse signals in the Fourier domain using convex optimization with only partial signal knowledge. These early papers outline the principles of incoherence, isometry, and random projections that dictate conditions on compressive sampling strategies along with the recovery guarantees and reconstruction principles that characterize the field.

Compressive sampling [14] allows subversion of traditional sampling theorems if the target signal is compressible in some basis or frame. Video signals are highly compressible—as demonstrated by the success of the MPEG-4 video compression standard [15]—but conventional video acquisition technology wastes sampling resources by collecting massive amounts of data with a huge array of sensors before extracting the important information. With compressive video acquisition, we can acquire video signals with greater sampling efficiency, hence simplifying encoding at the expense of complex decoding. Furthermore, we can do it with only a few sensors, which is economical when imaging at exotic wavelengths that require expensive photodetectors.

1.2 Motivation

The single-pixel camera [16–18] is a simple compressive imaging framework that consists mainly of a digital micromirror device (DMD) and a single photodetector. A DMD is a two-dimensional array of tiny mirrors, each of which can point either toward the sensor or away from the sensor. One row of a binary-valued compressive measurement matrix Φ is reshaped and loaded into the 2D mirror array at every sensing cycle to obtain exactly one measurement; each measurement is a linear combination of would-be pixel intensities from the target image. The temporally sequential imaging approach of the single-pixel camera is comparable to a conventional camera with an extremely slow shutter speed; hence, to avoid motion blur, early approaches to compressive video acquisition assume the target video consists only of slowly-changing scenes [19].

The measurement rate of a single-pixel camera is limited by the mirror-switching rate of the DMD. Currently, the fastest DMD on the market is Texas Instruments' DLP7000 with a switching frequency of $f_{\mathsf{dmd}} = 32{,}522$ Hz [20]. If we want to acquire a video with the DLP7000 at a frame rate of $f_r = 24$ Hz we must do it with only $f_{\mathsf{dmd}}/f_r = 1355$ mea-

surements per frame. For small frames of 128×128 pixels, this yields a sampling ratio of about 8%—a severe undersampling even for compressive sensing. Hence, to make compressive video acquisition feasible, it's sensible to use more than one sensor (but still many fewer sensors than a traditional CCD array) to increase the total number of measurements available per frame.

One possible application of compressive video acquisition is remote sensing of infrared radiation. Infrared sensors are up to ten times more expensive than visible light sensors; hence, compressive sampling makes infrared image and video acquisition affordable. Since compressive video acquisition involves simple encoding and complex decoding, it is well-suited for quickly acquiring video in hostile environments (e.g. a forest fire) and then reconstructing it offline later in safe conditions.

1.3 Proposed Framework

In the following, we outline a framework for compressive acquisition of video with multiple sensors. To our knowledge, no work has yet explicitly considered using multiple sensors for compressive video acquisition. Our framework is block-based for computational feasibility and to allow dynamic allocation of measurements within each frame. Our original contributions are as follows: we explicitly consider the feasibility of the compressive video acquisition problem using single-pixel cameras; we explicitly consider using multiple single-pixel cameras; we introduce a texture-focused acquisition mode and a motion-focused acquisition mode to address different types of perceptual salience in video signals; we do not require acquisition of reference frames, as groups of frames are jointly reconstructed to exploit temporal redundancy and prevent alias accumulation; we recover implicit reference frames and their associated innovation frames using the theory of distributed compressive sampling.

Chapter 2

Literature Review

2.1 Compressive sampling

Discrete sampling can be described in terms of a linear system of equations:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z} \tag{2.1}$$

where $\mathbf{y} \in \mathbb{R}^M$ is the measurement vector, $\mathbf{x} \in \mathbb{R}^N$ is the target signal, $\mathbf{A} \in \mathbb{R}^{M \times N}$ is the sampling operator, and $\mathbf{z} \in \mathbb{R}^M$ is noise. In traditional sampling, $M = N$, and $\mathbf{A}$ is simply the identity matrix.

In compressive sampling, $M < N$, $\mathbf{A} := \Phi$, and $\mathbf{x} := \Psi\boldsymbol{\alpha}$; i.e.,

$$\mathbf{y} = \Phi\Psi\boldsymbol{\alpha} + \mathbf{z} \tag{2.2}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^N$ is a sparse vector, $\Phi \in \mathbb{R}^{M \times N}$ is a random matrix whose rows are independently and identically drawn from a random distribution, and Ψ is a matrix whose columns contain N dictionary elements $\{\boldsymbol{\psi}_i\}_{i=1}^N$ in $\mathbb{R}^N$. The dictionary Ψ can be a basis or a frame.

It is essential that $\boldsymbol{\alpha}$ be sparse or at least compressible; this allows us to sample at a rate proportional to the signal's sparsity level $K = \|\boldsymbol{\alpha}\|_{\ell_0}$ instead of its (much larger) ambient dimension N. Note that $\|\cdot\|_{\ell_0}$ indicates the ℓ_0 "norm", defined as the number of

nonzero elements of a vector; i.e., $\|\boldsymbol{\alpha}\|_{\ell_0} := \#\{k : \alpha_k \neq 0\}$. Sparsity occurs when $K \ll N$; compressibility occurs when the sorted magnitudes of the elements of $\boldsymbol{\alpha}$ exhibit power-law decay.

Compressive sampling performs a dimension-reducing, information-preserving linear transformation on the target signal. Measurements are acquired via computing M random linear combinations of signal elements, creating an underdetermined system of linear equations. Strictly speaking, an underdetermined system of linear equations has an infinite number of possible solutions; however, if Φ and Ψ satisfy certain conditions and if M is sufficiently large, we can recover $\boldsymbol{\alpha}$ from the underdetermined system simply by finding the sparsest signal that explains our measurements. In the noiseless case, we write:

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_{\ell_0} \qquad \text{subject to } \mathbf{y} = \Phi\Psi\boldsymbol{\alpha} \tag{2.3}$$

This is a constrained combinatorial optimization problem, which is known to be NP-hard. Luckily, we can perform convex relaxation on the problem:

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_{\ell_1} \qquad \text{subject to } \mathbf{y} = \Phi\Psi\boldsymbol{\alpha} \tag{2.4}$$

where $\|\cdot\|_{\ell_p}$ indicates the ℓ_p norm; i.e., $\|\mathbf{v}\|_{\ell_p} := \left(\sum_{i=1}^{N} |v_i|^p\right)^{\frac{1}{p}}$. The formulation in (2.4) is referred to as "Basis Pursuit" [21]. It is an optimization principle, not a specific algorithm, and several algorithms for its solution exist in the literature.

If we know that the optimal solution of (2.4) is equivalent to the optimal solution of (2.3), we can guarantee successful recovery of the compressively sampled signal. Hence, we require the random sampling operator Φ to satisfy certain conditions; for instance, we often require Φ to be incoherent with Ψ, where their mutual coherence is defined as [22]:

$$\mu(\Phi, \Psi) := \max_{m,i} \frac{|\langle \boldsymbol{\phi}_m, \boldsymbol{\psi}_i \rangle|}{\|\boldsymbol{\phi}_m\|_{\ell_2} \|\boldsymbol{\psi}_i\|_{\ell_2}} \tag{2.5}$$

where $\boldsymbol{\phi}_m$ is the mth row of Φ and $\boldsymbol{\psi}_i$ is the ith column of Ψ. Note that $\langle \cdot, \cdot \rangle$ indicates the inner product. Incoherence between Ψ and Φ means that when a signal is sparse in Ψ,

it is dense in Φ; this increases the probability that all significant coefficients of the target signal are captured during sampling. Ideally, we would like a (Φ, Ψ) pair that achieves the minimal mutual coherence value of $\mu(\Phi, \Psi) = 1/\sqrt{N}$. Though intuitively appealing, the mathematics of incoherence yield weak guarantees: for successful recovery we theoretically require $K \leq \frac{1}{2}\big(1 + \frac{1}{\mu(\Phi,\Psi)}\big)$ [23]. In practice, however, we observe successful recovery with much larger values of K.

A stronger condition we might impose on Φ and Ψ is adherence to the restricted isometry property (RIP) [24]. The RIP is defined as:

$$(1 - \delta_K)\|\boldsymbol{\alpha}\|_{\ell_2}^2 \leq \|\mathbf{A}\boldsymbol{\alpha}\|_{\ell_2}^2 \leq (1 + \delta_K)\|\boldsymbol{\alpha}\|_{\ell_2}^2 \tag{2.6}$$

where $\mathbf{A} := \Phi\Psi$ and δ_K is the restricted isometry constant of order K. The RIP of order K holds if $\delta_K \in (0, 1)$ exists for all K-sparse vectors in $\mathbb{R}^N$. The RIP ensures that small perturbations in the signal do not lead to large perturbations in the measurements; i.e., it enforces a stability condition on the sampling operator. Furthermore, recovery is guaranteed when $\delta_{2K} < \sqrt{2} - 1$ [25].

Neither incoherence nor the RIP are strictly necessary conditions for guaranteed recovery, but they are sufficient [26] and popular in the literature [27]. If either holds, the optimal solution of (2.4) is guaranteed to be equivalent to the optimal solution of (2.3). Still, successful recovery is observed even when the RIP is not satisfied. Further efforts to derive recovery guarantees via concepts of isotropy [28] and a "restricted width" property [29] are underway, but these are still in their infancy. To date, the best proven approach to deriving precise conditions for guaranteed recovery uses combinatorial geometry [30,31], from which we receive a lower bound on M. With enough measurements, the target signal is guaranteed to be captured either perfectly [32] or approximately [33]. Specifically, we require:

$$M \gtrsim 2K \cdot \log(N/M)$$

for K, M, N large and $K \ll N$ [34], which is an accurate and useful result.

2.2 Distributed compressive sampling

Compressive sampling succeeds because of the known sparse structure of the target signal. In distributed compressive sampling, we collect an ensemble of target signals and impose additional structure on the ensemble by assuming the signals are jointly sparse.

Given an ensemble of J signals $\{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_J\}$, we assume that each $\boldsymbol{\alpha}_j$ for $j \in \{1, \ldots, J\}$ consists of a sparse common component and a sparse innovation component [35], i.e.

$$\boldsymbol{\alpha}_j = \boldsymbol{\alpha}^C + \boldsymbol{\alpha}_j^Q \tag{2.7}$$

where $\boldsymbol{\alpha}^C$'s support set $\Lambda^C := \{k : \alpha_k^C \neq 0\}$ is common to all $\boldsymbol{\alpha}_j$ and the support set Λ_j^Q is unique for each innovation $\boldsymbol{\alpha}_j^Q$. Other joint-sparse signal models are possible: we can assume that all $\boldsymbol{\alpha}_j$ fully share the same support with no innovations; we can assume that only the innovations are sparse while the common component is non-sparse. Such models are useful elsewhere but are unrealistic for the case of compressive video acquisition.

Intuitively, we expect that the additional structure of joint sparsity will allow us to recover the entire ensemble using fewer measurements than we'd need to recover each signal individually. This is true; analogous to the single-signal case, measurement conditions and recovery guarantees exist for distributed compressive sampling. The entire ensemble is reconstructed jointly, but each signal can be acquired independently.

Assuming each signal in the ensemble is sampled using a measurement matrix $\Phi_j \in \mathbb{R}^{M_j \times N}$ that is sufficiently incoherent with Ψ, the number of measurements required to recover the signal ensemble is subject to an additive condition [36]:

$$\sum_{j=1}^{J} M_j > c\left(K^C + \sum_{j=1}^{J} K_j^Q\right) \tag{2.8}$$

where M_j is the number of measurements collected of the jth signal, $K^C = \|\boldsymbol{\alpha}^C\|_{\ell_0}$, $K_j^Q = \|\boldsymbol{\alpha}_j^Q\|_{\ell_0}$, and $c \approx \log_2(1 + \frac{N}{K})$ is an oversampling factor that depends on the total

sparsity of the signal ensemble $K := K^C + \sum_{j=1}^{J} K_j^Q$. In the literature, the sparsity dependency is often ignored and we heuristically set $c \approx 3$ [37]. Furthermore, each individual signal in the ensemble must receive enough measurements to reconstruct its local innovation component; i.e., we require $M_j > cK_j^Q$ for all $j \in \{1, \ldots, J\}$ [38]. Failure to meet this condition along with the additive condition in (2.8) leads to a failure to reconstruct [39, 40].

When attempting compressive video acquisition with multiple sensors, one might naively assemble signal ensembles using measurement vectors drawn independently from each sensor. This is the wrong approach. Assuming all sensors have identical views of the scene, they all sample identical signals. Hence, each frame in the video is better reconstructed by simply concatenating sensor measurements. In the following framework for compressive video acquisition, we take groups of frames as signal ensembles. Neighbouring frames are highly correlated but not identical, which makes them good candidates for joint reconstruction.

2.3 Single-pixel camera

The single-pixel camera [16–18] is a simple compressive imaging framework that consists of a DMD and a single photodetector. Image acquisition with only one sensor is made possible by trading broad spatial sampling for lengthy temporal sampling. A single-pixel camera has binary-valued hardware; it consists of a two-dimensional array of tiny mirrors that can point either toward or away from the sensor. Each row of a compressive measurement matrix Φ yields exactly one measurement. The way a DMD uses a measurement matrix Φ is illustrated in Figure 2.1. Random linear measurements are acquired by projecting random linear combinations of would-be pixel intensities from the target image onto the photodetector. This is illustrated in Figure 2.2. The effect of randomness is that all compressive measurements of a certain image contain roughly the same amount of information and are naturally encrypted.

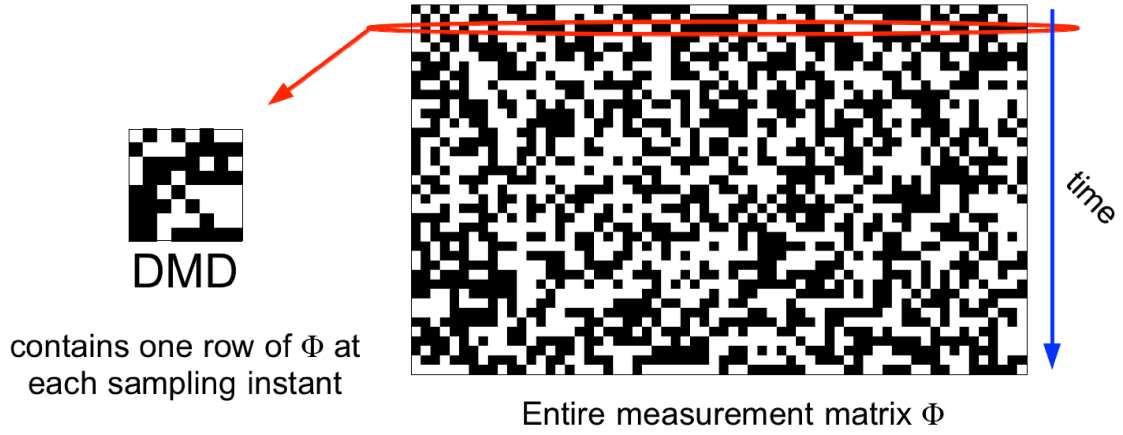

Figure 2.1: Capturing an 8×8 image with a single-pixel camera. In this example, $N = 64$, $M = 39$, and the sampling ratio $= 61\%$. The measurement matrix Φ has M rows and N columns. For each measurement, one row of Φ is reshaped and loaded into the DMD.

2.4 Compressive video acquisition

Compressive video acquisition poses unique challenges: on a limited measurement budget with no knowledge of the target signal besides compressibility, how do we sample and reconstruct the scene to obtain the best possible video quality? First, we must maximally exploit both spatial and temporal redundancies in the signal. This is usually done during reconstruction. Second, we must deploy our limited sensing resources according to where they are needed most. This is done during sampling, when measurements are allocated within the scene according to where they will have the largest impact on reconstruction quality.

2.4.1 Spatial redundancy

Spatial redundancy is easily accounted for via the proper choice of sparsifying basis Ψ. Although the theory of compressive sampling originated with analysis of Fourier-sparse signals, it quickly expanded to include signals that are sparse in arbitrary orthonormal

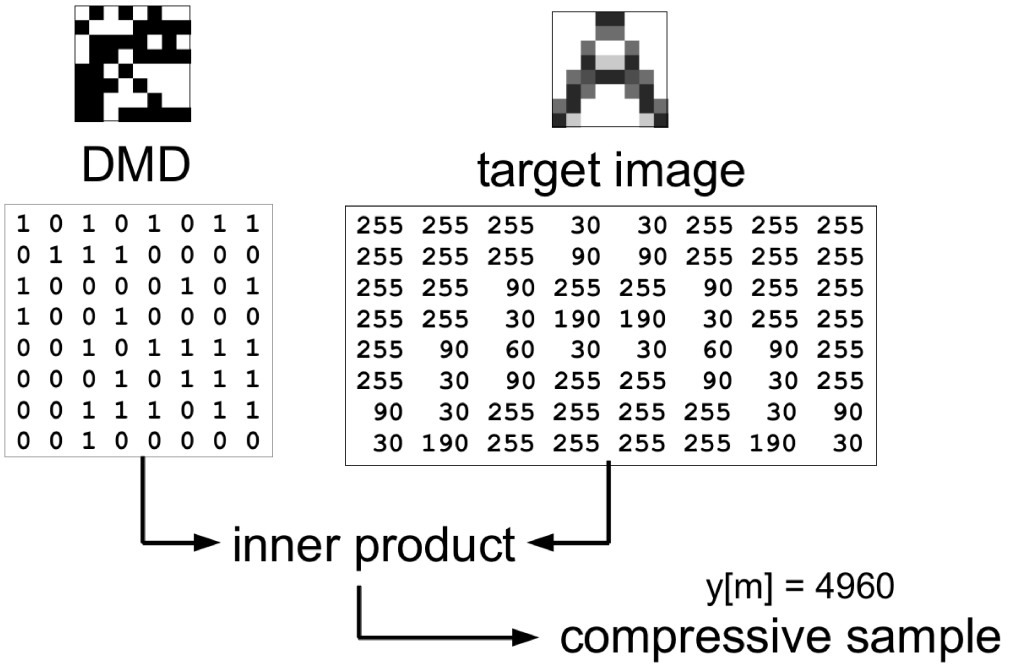

Figure 2.2: Each measurement is acquired by optically computing the inner product between the binary matrix on the DMD and the would-be pixel intensities from the target image. Note that in real applications the target image is not an array of unsigned 8-bit integers.

bases, tight frames, and overcomplete dictionaries [41]. Video signals are often composed of natural images, and most natural images are sparse in a wavelet basis; hence, wavelet bases are effective and popular choices for compressive video acquisition [42, 43]. We use wavelets in our framework. Fourier and DCT bases are also common, especially in medical applications [44, 45].

Spatial redundancy can also be exploited via multi-resolution reconstruction methods. The approach in [46] uses a multi-resolution approach combined with optical flow estimation for video reconstruction. They use a unique multi-scale measurement matrix to first reconstruct a low-resolution version of each frame and then estimate the optical flow of the scene. The optical flow estimate is used as input to a high-resolution optimization algo-

rithm. Optical flow estimation effectively exploits temporal redundancy but is extremely computationally intensive, and so we do not use it in our approach.

2.4.2 Temporal redundancy

Temporal redundancy is addressed many ways in the literature. For instance, one might simply ignore it and reconstruct all frames as if they were independent images. Obviously, this naive approach is far from optimal. At the other dimensional extreme, one might treat the video as a 3D volume and try to reconstruct the entire sequence simultaneously [47,48]. While theoretically clever, this approach is extremely computationally complex and infeasible for a video of any appreciable size.

Reference/difference-based methods are a popular reconstruction strategy [49–51]. Reference frames—either measured uncompressively or with a greater number of compressive measurements than non-reference frames—are periodically inserted during the sampling phase. Non-reference frames are then reconstructed using an inter-frame differencing approach, where the measurements from one frame are subtracted from the measurements of the subsequent frame and then reconstructed. The reconstructed difference is much sparser than an entire frame; frames are recovered by adding their differences to the previously reconstructed frame. While effective, the periodic insertion of reference frames wastes sampling resources, disrupts the frame rate, and may cause alias accumulation if not performed often. Furthermore, such methods require differenced measurement vectors to be the same length, which implies either uniform sampling throughout time or non-uniform sampling with discarded measurements. Discarding measurements is a waste of resources, and uniform sampling is not optimal; different parts of a scene require different sampling ratios due to varying sparsity levels, and dynamic measurement allocation is especially important on a limited measurement budget. In the following framework, we avoid using explicit reference frames. We instead compute implicit reference frames based on the joint reconstruction of

frame groups. We then compare each frame with its implicit reference frame and use the result to individually reconstruct the innovation components of each frame.

2.4.3 Block-based imaging

Video signals are dimensionally huge. Larger images are generally sparser in the wavelet domain thanks to extra levels of dyadic decomposition, but measuring an entire frame simultaneously is computationally infeasible: storing the measurement matrix takes too much memory; the reconstruction process takes too long. A solution to this is to use block-based imaging [52], which breaks up the frame into smaller blocks that are collected independently. This results in decreased computation time plus a lighter memory burden.

One block-based approach that exploits temporal redundancy involves motion estimation/compensation [53]. The requisite motion vector search is very computationally expensive, and since reconstruction is already so arduous we prefer to avoid any extra computation. Furthermore, the block sizes at which motion compensation is effective are much smaller than the block sizes at which block-based compressive reconstruction is effective—motion compensation requires blocks so small they are hardly sparse at all. In the following framework, we do not use motion compensation. We partition scenes using a large block size to accommodate a reasonable number of levels of wavelet decomposition while still maintaining computational feasibility.

2.5 Reconstruction strategies

Many algorithms have been developed to recover sparse signals from compressive measurements. They can be divided into two major categories: convex optimization and greedy pursuits. Bayesian probability/belief-based methods are also common [54], but they are computationally complex, require prior estimates on the coefficient distribution of the tar-

get signal, come with no guarantees, and are therefore unsuitable for serious recovery of image and video signals.

2.5.1 Convex optimization

The theory of compressive sampling was originally developed using tools from convex optimization. Indeed, the formulation in (2.4) is a convex problem, and can be solved easily using existing linear programming methods [55].

Unfortunately, (2.4) is only feasible in noiseless cases, which is extremely unrealistic for engineering applications. As a result, many approaches focus on solving formulations that can accommodate noise, e.g.:

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_{\ell_1} \qquad \text{subject to} \quad \|\mathbf{y} - \Phi\Psi\boldsymbol{\alpha}\|_{\ell_2} \leq \epsilon \tag{2.9}$$

and the closely related problem:

$$\hat{\boldsymbol{\alpha}} = \arg\min \frac{1}{2}\|\mathbf{y} - \Phi\Psi\boldsymbol{\alpha}\|_{\ell_2}^2 + \tau\|\boldsymbol{\alpha}\|_{\ell_1} \tag{2.10}$$

The problem in (2.9) is a noise-accommodating variation (where ϵ depends on the noise level) of the basis pursuit problem in (2.4) and hence is often referred to as "basis pursuit de-noising". The problem described by (2.10) is much more popular than (2.9); it is in some sense an unconstrained version of (2.9), though both approaches generally yield different results and come with different guarantees [56].

Compressible signals contain many small non-zero coefficients where a truly sparse signal would contain zeros. Noise also commonly manifests as many small non-zero coefficients; hence, to a convex optimization algorithm, the discardable part of a compressible signal looks very much like noise. In (2.10), the ℓ_1 regularization term forces small coefficients to zero while the ℓ_2 error term encourages accurate reconstructions. As a result, we are able to control the trade-off between sparsity and accuracy via the penalty parameter τ.

Ultimately, it will be (2.10) that we solve to reconstruct our signals from their compressive samples.

2.5.2 Greedy pursuits

There also exist "greedy" approaches [57] to solving (2.4), including orthogonal matching pursuit (OMP) [58], compressive sampling matching pursuit (CoSaMP) [59], and stagewise orthogonal matching pursuit (StOMP) [60]. We might even try to simultaneously reconstruct an entire signal ensemble using simultaneous orthogonal matching pursuit (SOMP) [61]. These methods are generally faster than convex optimization and perform comparably under certain conditions. They are synthesis-based; i.e., they build up a signal representation by selecting the dictionary atom that explains the largest proportion of signal energy on every iteration. However, they do not come with the same strong mathematical guarantees as convex methods. In fact, greedy approaches fail when the sparsifying basis Ψ is too coherent [62], while convex optimization methods remain successful [41].

Chapter 3

Framework Details

3.1 Signal model

We assume we are trying to capture real-time video in real life. A video is a sequence of images, and images contain high levels of spatial redundancy. Hence, we assume each frame of the video is compressible in some basis Ψ. Explicitly, we assume $\mathbf{x}_f = \Psi\boldsymbol{\alpha}_f$ and $\tilde{\boldsymbol{\alpha}}_f = \tilde{\Psi}\mathbf{x}_f$, where $\mathbf{x}_f$ is the vectorized target image, $\tilde{\Psi}$ is the dual basis of Ψ, and the magnitudes of the sorted coefficients of both $\boldsymbol{\alpha}_f$ and $\tilde{\boldsymbol{\alpha}}_f$ exhibit power-law decay.

Videos also contain high levels of temporal redundancy; i.e., each frame is usually highly correlated with both prior and subsequent frames. In the following, we employ a joint-sparse signal model [36] to describe inter-frame correlations. We assume that we can partition the set of all frames into groups such that each frame consists of a sparse common component and a sparse unique component relative to other frames in the group:

$$\boldsymbol{\alpha}_f = \boldsymbol{\alpha}_g^C + \boldsymbol{\alpha}_f^Q \tag{3.1}$$

where G is the total number of frame groups, $\xi = \{1, 2, \ldots, G\}$, and $g \in \xi$. Moreover, Γ_g is the set of all frame indices in group g, and $f \in \Gamma_g$. For each $g \in \xi$, the common component

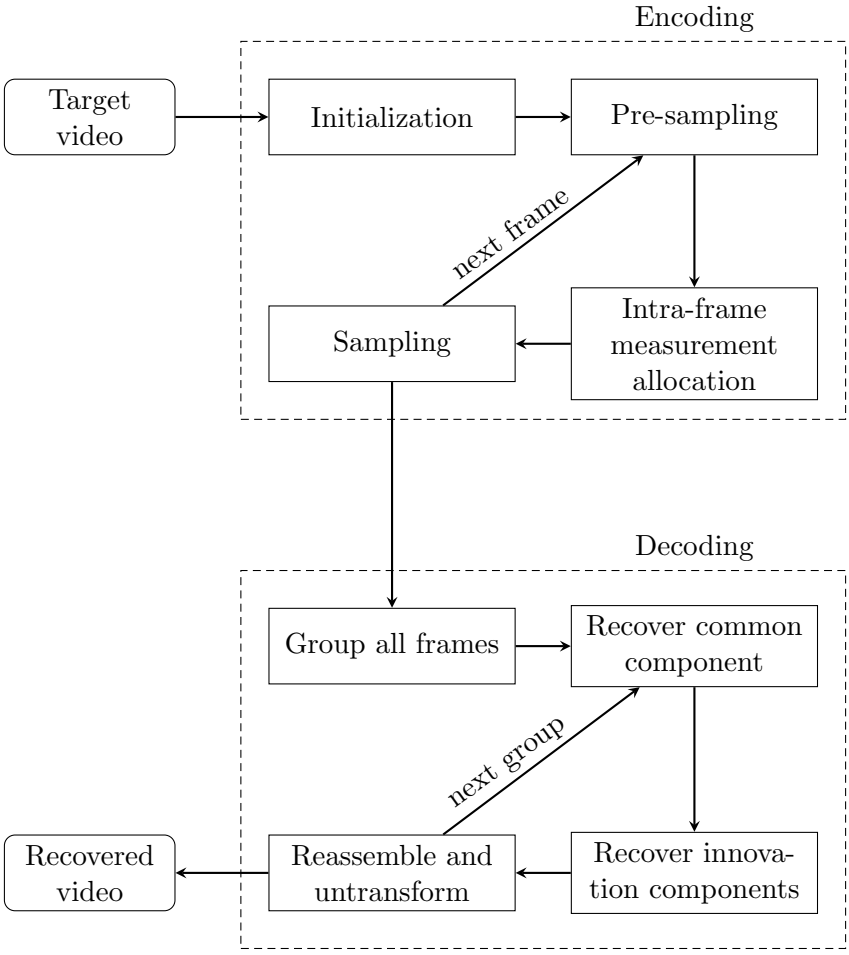

Figure 3.1: Overall block diagram of our framework.

$\boldsymbol{\alpha}_g^C$ is identical for all $\boldsymbol{\alpha}_f$ when $f \in \Gamma_g$. Furthermore, we assume that $\|\boldsymbol{\alpha}_f^Q\|_{\ell_0} \ll \|\boldsymbol{\alpha}_g^C\|_{\ell_0}$ for all $f \in \Gamma_g$ and all $g \in \xi$. Obviously, this assumption will become stronger or weaker depending on how the frames are grouped. Frame grouping is performed during reconstruction. The details of the frame grouping process are outlined in Section 3.3.3.

In our framework, we only begin video reconstruction (decoding) after all samples have been collected (encoding). A high-level block diagram is shown in Figure 3.1. The remainder of this chapter will discuss the functions of each stage in detail.

3.2 Video acquisition

3.2.1 Block-based imaging

We divide each frame into blocks of size $L \times L$. We assume the signal model defined in Section 3.1 holds for each block $b \in \beta$, where $\beta = \{1, 2, \ldots, B\}$ and B is the total number of blocks in each frame. The vectorized target image corresponding to block b and frame f is now written as $\mathbf{x}_{(f,b)} \in \mathbb{R}^{L^2}$; the entities $\boldsymbol{\alpha}_f, \boldsymbol{\alpha}_g^C, \boldsymbol{\alpha}_f^Q$, and ξ described in Section 3.1 acquire an additional index b. Block-based imaging radically reduces the amount of memory required to store Φ, eases the computational burden of reconstruction, and allows us to distribute measurements within a frame according to where they will be most useful. Under this paradigm, each block has a signal length $N = L^2$.

3.2.2 Choice of measurement matrix

Gaussian-distributed random matrices are proven to be ideal for compressive sampling. They satisfy the RIP, are incoherent with most bases Ψ, and require the fewest number of measurements in order to successfully recover any target signal. However, Gaussian matrices are real-valued and dense, which makes them incompatible with the binary-valued hardware of the single-pixel camera's DMD and computationally expensive to generate and store. [63].

In contrast, the scrambled block-Hadamard ensemble (SBHE) is sparse, can be generated quickly using a few permutation operations, and performs nearly as well as traditional Gaussian sensing matrices [64]. Importantly, the SBHE is also compatible with single-pixel camera hardware. Hence, we choose the SBHE as our measurement matrix Φ due to its technical feasibility, memory efficiency, and fast implementation. Furthermore, we can quickly compute matrix-vector multiplications since the SBHE is structured—this will be important during reconstruction.

3.2.3 Arrangement and behaviour of sensors

Using multiple sensors allows us to collect more information at one time than we'd be able to collect with only one sensor. This is especially valuable in the case of the single-pixel camera, which collects measurements sequentially over time. For simplicity, we assume each sensor has an identical view of the target scene. Each sensor is paired with its own DMD plus a processing unit that can perform simple arithmetic operations on the measurements in real time. Additionally, all sensors are connected to a central hub capable of executing fast convex optimization algorithms.

Let $\zeta = \{1, 2, \ldots, S\}$, where S is the total number of sensors. Each sensor $s \in \zeta$ is assigned a unique Φ_s that it repeatedly uses to measure every block in each frame. This is done by simply generating each Φ_s with a unique random seed. Using a unique measurement matrix for each sensor avoids redundant measurements; using the same measurement matrix repeatedly for each frame and block eliminates the need to constantly compute new matrices.

The compressive sampling rate of each sensor is limited by the pattern switching rate of the DMD, f_{dmd}. Each sensor collects the same number of measurements per frame, calculated as:

$$M_{\mathsf{sensor}} = \left\lceil \frac{f_{\mathsf{dmd}}}{f_r} \right\rceil \tag{3.2}$$

where f_r is the desired frame rate of the reconstructed video. This allows for a steady frame rate upon reconstruction. Each sensor will distribute its measurements across all blocks $b \in \beta$ in the frame. The total framewise compressive sampling rate is given by

$$M_{\mathsf{frame}} = SM_{\mathsf{sensor}} \tag{3.3}$$

Before the main sensing loop, each sensor thoroughly samples the first frame by taking an initial M_{init} measurements of each block, where $M_{\mathsf{sensor}} \leq BM_{\mathsf{init}} < BN$. All sensors then send these initial measurements to a central hub, where they are pooled together and each block is reconstructed by a fast convex optimization algorithm. This slow and

computationally expensive process is necessary in order to obtain a sparsity estimate $\hat{K}_b$ for each block; sparsity estimates are important for effective measurement allocation.

At the start of each frame acquisition cycle each sensor obtains an initial M_0 measurements of each block in the frame, where $BM_0 < M_{\mathsf{sensor}}$. These preliminary measurements $\mathbf{y}_{0(s,f,b)}$ are stored locally within each sensor for one cycle in order to efficiently compute inter-frame differences for each block. The absolute inter-frame differences are then used during measurement allocation.

Each sensor can operate in one of two modes: "texture-focused" mode or "motion-focused" mode. In texture-focused mode, sensors allocate their measurements across all blocks according to the estimated sparsity of each block, $\hat{K}_b$. The details of texture-focused mode are outlined in Section 3.2.4. In motion-focused mode, sensors allocate their measurements across all blocks according to the absolute local inter-frame difference:

$$d_{0(s,f,b)} = \|\mathbf{y}_{0(s,f,b)} - \mathbf{y}_{0(s,f-1,b)}\|_{\ell_1}$$

The details of motion-focused mode are outlined in Section 3.2.5.

At the end of each frame acquisition cycle, all sensors send their measurements to the central hub. We assume sending information to the central hub is a parallel process to other sensor operations and does not interrupt video acquisition. Once at the hub, measurements from all sensors within a block are stacked:

$$\mathbf{y}_{\mathsf{stack}(f,b)} = \begin{bmatrix} \mathbf{y}_{(1,f,b)} \\ \vdots \\ \mathbf{y}_{(S,f,b)} \end{bmatrix} = \begin{bmatrix} \Phi_{1(1:M_0+M_{(1,f,b)})} \\ \vdots \\ \Phi_{S(1:M_0+M_{(S,f,b)})} \end{bmatrix} \mathbf{x}_{(f,b)} \tag{3.4}$$

and stored in a buffer. Note that $M_{(s,f,b)}$ indicates the number of measurements allocated to sensor s, frame f, and block b, and $\Phi_{s(1:M)}$ indicates the first M rows of Φ_s. Additional buffered measurements are accumulated with every new frame. When the number of measurements in a block's buffer exceeds a threshold T, a new sparsity estimate $\hat{K}_b$ for

that block is produced and communicated to all sensors. More measurements are allocated to fast-changing blocks, so the update rate of $\hat{K}_b$ will be higher in blocks with significant motion—this lowers the risk of estimation error due to large sparsity differences. To ensure that sparsity estimates are sufficiently synchronized with the actual sparsity of the signal, we force the sparsity estimation algorithm to terminate after a small number of iterations. The sensors never communicate with each other. A diagram of the operation of an individual sensor is shown in Figure 3.2.

3.2.4 Texture-focused sensor mode

The sensing framework periodically produces sparsity estimates $\hat{K}_b$ for $b \in \beta$. Assuming the innovation signals for each frame are sufficiently small, $\hat{K}_b$ is a good predictor of the sparsity of the common component of the frame group. Hence, sampling according to $\hat{K}_b$ is approximately analogous to sampling according to the sparsity of $\boldsymbol{\alpha}^C_{(g,b)}$.

The number of measurements required for accurate reconstruction is related to signal sparsity as follows [34]:

$$M \gtrsim 2K \cdot \log(N/M) \tag{3.5}$$

for K, M, N large, $K \ll N$. Hence, we can optimally adjust the number of measurements required by solving the following equation for $\hat{M}_b$:

$$\hat{M}_b = 2\hat{K}_b \cdot \log(N/\hat{M}_b) \tag{3.6}$$

which yields:

$$\hat{M}_b = N \exp(-W\left(\frac{N}{2\hat{K}_b}\right)) \tag{3.7}$$

where $W(\cdot)$ is the Lambert W function [65]. Then, to allocate the sensor's measurements among all blocks within a frame, we normalize $\hat{M}_b$ and assign measurements to each block as follows:

$$M_{(s,f,b)} = \mathsf{round}\left((M_{\mathsf{sensor}} - BM_0)\frac{\hat{M}_b}{\hat{M}_\beta}\right) \tag{3.8}$$

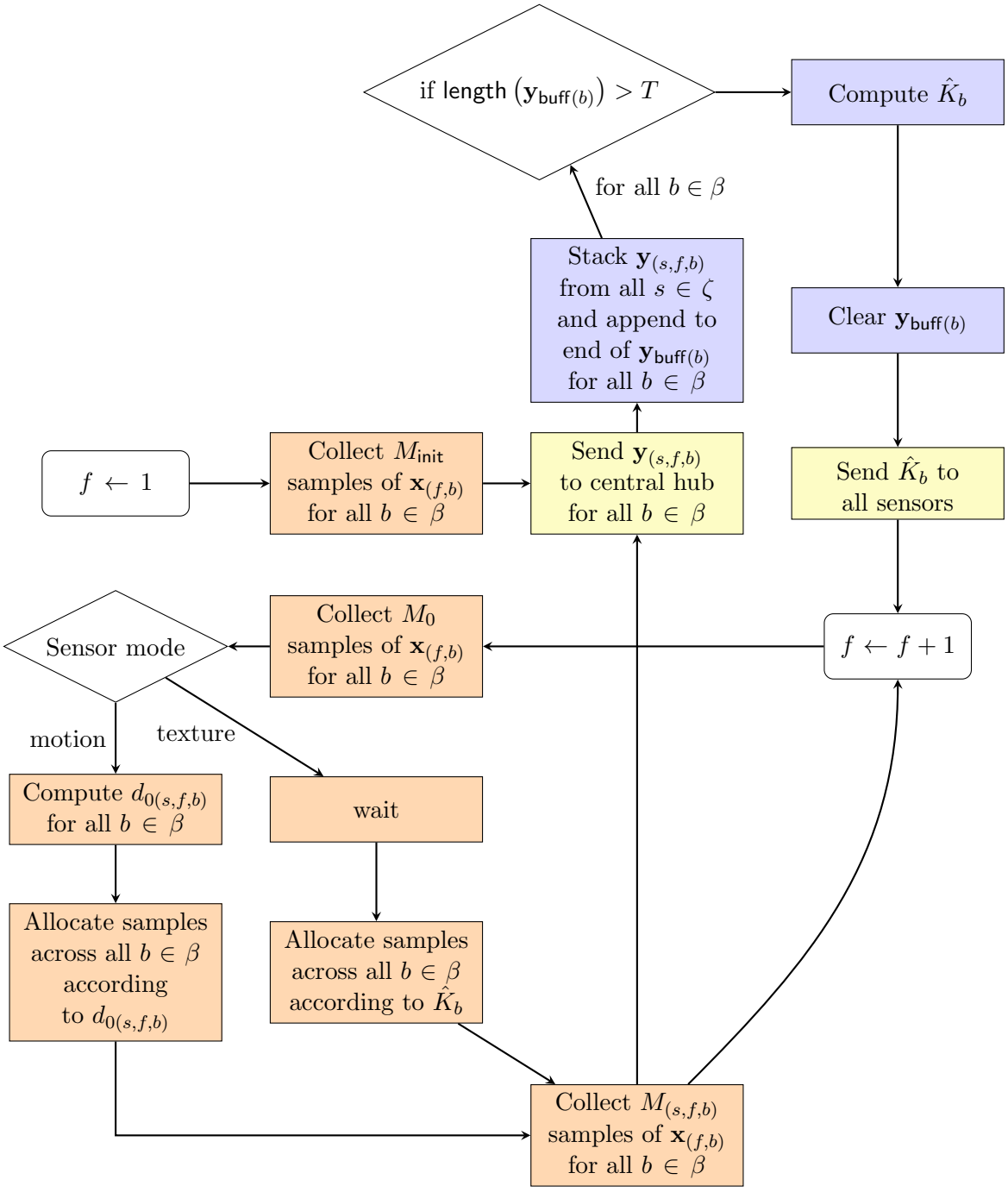

Figure 3.2: Frame acquisition process for sensor $s \in \zeta$. Sensor processes are orange, hub processes are blue, and communication processes are yellow.

where $\hat{M}_\beta = \sum_{b \in \beta} \hat{M}_b$. For computational efficiency, the ratio $\hat{M}_b / \hat{M}_\beta$ should be calculated at the hub and communicated to all sensors instead of $\hat{K}_b$. Any leftover measurements produced by $\mathsf{round}(\cdot)$ are added to the block with the most measurements, since a large number of measurements indicates perceptual importance. Similarly, any extra-budgetary measurements produced by $\mathsf{round}(\cdot)$ are subtracted from the block with the most measurements, since subtracting from a block with fewer measurements may cause reconstruction failure.

3.2.5 Motion-focused sensor mode

Texture-focused mode allocates measurements according to the sparsity of a frame group's common component $\boldsymbol{\alpha}^C_{(g,b)}$; hence, motion-focused mode attempts to allocate measurements according to each frame's innovation component $\boldsymbol{\alpha}^Q_{(f,b)}$. Since we have no direct access to the innovation component or its sparsity, we find the inter-frame innovation difference instead. We compute:

$$d_{0(s,f,b)} = \|\mathbf{y}_{0(s,f,b)} - \mathbf{y}_{0(s,f-1,b)}\|_{\ell_1} \tag{3.9}$$

which is equal to:

$$d_{0(s,f,b)} = \|\Phi_{s(1:M_0)} \Psi \boldsymbol{\alpha}_{(s,f,b)} - \Phi_{s(1:M_0)} \Psi \boldsymbol{\alpha}_{(s,f-1,b)}\|_{\ell_1} \tag{3.10}$$

which through (3.1) becomes:

$$d_{0(s,f,b)} = \|\Phi_{s(1:M_0)} \Psi \Big(\boldsymbol{\alpha}^C_{(g,b)} + \boldsymbol{\alpha}^Q_{(f,b)} - \big(\boldsymbol{\alpha}^C_{(g,b)} + \boldsymbol{\alpha}^Q_{(f-1,b)} \big) \Big) \|_{\ell_1} \tag{3.11}$$

The common components cancel, leaving:

$$d_{0(s,f,b)} = \|\Phi_{s(1:M_0)} \Psi \big(\boldsymbol{\alpha}^Q_{(f,b)} - \boldsymbol{\alpha}^Q_{(f-1,b)} \big) \|_{\ell_1} \tag{3.12}$$

□

Due to the RIP, the distance $\|\boldsymbol{\alpha}^Q_{(f,b)} - \boldsymbol{\alpha}^Q_{(f-1,b)}\|_{\ell_1}$ is well-preserved under the transformation $\Phi_{s(1:M_0)} \Psi$. Hence, we expect the ℓ_1 norm of the inter-frame measurement difference

to perform reasonably well as an estimator of innovation sparsity, provided the signal coefficients are sufficiently small and do not vary too much in magnitude. Such assumptions are reasonable for inter-frame differences.

Finally, as in texture-focused mode, we allocate each sensor's measurements among all blocks within a frame after normalizing $d_{0(s,f,b)}$:

$$M_{(s,f,b)} = \mathsf{round}\left((M_{\mathsf{sensor}} - BM_0) \frac{d_{0(s,f,b)}}{d_{0\beta}} \right) \tag{3.13}$$

where $d_{0\beta} = \sum_{b \in \beta} d_{0(s,f,b)}$. Since we wish to minimize the amount of computation undertaken by a single sensor, we do not scale $d_{0(s,f,b)}$ with the Lambert W function. Any leftover measurements are allocated to the block with the most measurements, since this indicates salience; extra-budgetary measurements are taken from the block with the most measurements to avoid aliasing blocks with fewer measurements.

3.2.6 Choice of sensor mode

How many sensors should be texture-focused, and how many should be motion-focused? The most salient (and therefore important) part of a video is its motion, but we want to ensure successful reconstruction even if there is no motion. In the following, we derive a lower bound on the number of sensors that must be in texture-focused mode.

From the frame grouping module in Section 3.3.3 (omitting the block index b for clarity) we have:

$$M_g = \sum_{f \in \Gamma_g} M_f \tag{3.14}$$

where M_f is the number of measurements obtained of $\boldsymbol{\alpha}_f$ for $f \in \Gamma_g$ and M_g is the total number of measurements in frame group g . The number of measurements allocated to each frame (within a block) is given by:

$$M_f = N_t M_t + N_m M_m \tag{3.15}$$

where N_t is the number of texture-focused sensors, N_m is the number of motion-focused sensors, M_t is the number of measurements allocated to frame f by a texture-focused sensor, and M_m is the number of measurements allocated to frame f by a motion-focused sensor. For simplicity, we assume M_t and M_m are the same for all sensors in the same mode (and the same frame and block). Obviously, $N_t + N_m = S$. From the theory of distributed compressive sampling, we require the following condition to hold for successful recovery of the frame group [36]:

$$M_g > cK_g + c \sum_{f \in \Gamma_g} K_f \tag{3.16}$$

where K_g is the sparsity of the common component $\boldsymbol{\alpha}_g^C$ shared by all frames in frame group g, K_f is the sparsity of the innovation component $\boldsymbol{\alpha}_f^Q$ of frame $f \in \Gamma_g$, and c is an oversampling factor that depends on the sparsity of the signal but is heuristically regarded as $c \approx 3$ [37].

Assuming M_f for all $f \in \Gamma_g$ are identical and combining (3.14), (3.15), and (3.16), we obtain:

$$|\Gamma_g|(N_t M_t + N_m M_m) > cK_g + c \sum_{f \in \Gamma_g} K_f \tag{3.17}$$

where $|\cdot|$ indicates the cardinality of a set.

Letting $M_m \to \dfrac{M_{\mathsf{sensor}}}{B}$ while all innovation sparsities $K_f \to 0$:

$$|\Gamma_g|\Big(N_t M_t + N_m \frac{M_{\mathsf{sensor}}}{B}\Big) > cK_g \tag{3.18}$$

Without loss of generality, we assume each frame has uniform texture and let $M_t \to \dfrac{M_{\mathsf{sensor}}}{B}$ as well:

$$\frac{|\Gamma_g| M_{\mathsf{sensor}}}{B}(N_t + N_m) > cK_g \tag{3.19}$$

rearranging to isolate $N_t + N_m = S$ and recalling that $M_{\mathsf{sensor}} = \dfrac{f_{\mathsf{dmd}}}{f_r}$, we obtain:

$$N_t + N_m > \frac{cBK_g f_r}{|\Gamma_g| f_{\mathsf{dmd}}} \tag{3.20}$$

We see from (3.20) that the minimum number of sensors required to capture a scene with no motion depends on four things: the hardware sampling rate f_{dmd}, the desired frame rate f_r, the total sparsity of the scene BK_g, and the size of the frame groups used during reconstruction $|\Gamma_g|$. Here we have assumed $|\Gamma_g|$ and K_g are the same for all blocks, but generally they will vary. The result in (3.20) makes sense intuitively: more measurements per sensor necessitates fewer total sensors, a higher desired frame rate requires more measurements and hence more sensors, and a more complex frame texture (indicated by a large BK_g) requires more measurements to capture. Jointly reconstructing more frames at once reduces the number of sensors required since extra frames contribute additional information about the common texture component.

Our framework is flexible: using S sensors instead of just one allows either an increase in frame rate by a factor of S, a decrease in frame group size by a factor of $\frac{1}{S}$, or a simultaneous increase in frame rate by a factor of S_1 with a decrease in frame group size by a factor of $\frac{1}{S_2}$. Here, $S = S_1 S_2$. Smaller frame group sizes generally correspond to higher individual frame qualities; hence, we are able to trade off frame rate and frame quality.

Since we cannot know K_g in advance, we might assume e.g. $K_g \approx (0.2)N$, $c \approx 3$, and $|\Gamma_g| \approx \dfrac{BN}{SM_{\mathsf{sensor}}}$ to obtain:

$$N_t > (0.6)S - N_m \tag{3.21}$$

and use a fixed number of $N_t = \lceil (0.6)S \rceil$ sensors in texture-focused mode. We assign the remaining $N_m = S - N_t$ sensors to motion-focused mode. The number of sensors required to be in each mode will vary between applications. If there is extreme motion in the video, we should assign more sensors to motion-focused mode.

3.3 Video reconstruction

3.3.1 Choice of sparsifying basis

We choose biorthogonal CDF 9/7 wavelets as our basis Ψ. CDF 9/7 wavelets were used in the development of the lossy JPEG2000 image compression standard [66]; they are a very good choice for lossy compression of natural images. Wavelets also have a fast transform, which is useful during reconstruction.

3.3.2 Choice of optimization algorithm

Video signals are dimensionally huge; hence, we require an optimization algorithm that is fast and computationally feasible. We might consider a greedy pursuit for speed and simplicity; however, the synthesis frame for our chosen CDF 9/7 wavelet basis is too coherent for greedy recovery approaches to succeed [62]. Hence, we must use convex optimization.

Convex solvers capable of handling problems (2.4), (2.9), and (2.10) are available online [67, 68], but the interior-point methods they employ are slow and contain several nested loops. Moreover, we need an algorithm that does not require explicit computation and storage of $\mathbf{A} = \Phi\Psi$, as $\mathbf{A}$ is dense and very large.

The gradient projection for sparse reconstruction (GPSR) algorithm [69] has no nested loops and requires only matrix-vector products of the form $\mathbf{A}\boldsymbol{\alpha}$ and $\mathbf{A}^T\mathbf{y}$ to solve (2.10). This allows us to represent $\mathbf{A}$ and $\mathbf{A}^T$ as functions that take $\boldsymbol{\alpha}$ or $\mathbf{y}$ as input and quickly return $\mathbf{A}\boldsymbol{\alpha}$ or $\mathbf{A}^T\mathbf{y}$, respectively. This is much more memory-efficient than explicitly storing the entirety of $\mathbf{A}$, and especially useful when coupled with fast wavelet transforms as our sparsifying basis. With the correct choice of τ, the GPSR algorithm is both fast and accurate. We use GPSR for all convex optimization tasks in our framework.

3.3.3 Joint reconstruction of frame groups

Before we can jointly reconstruct groups of frames, we must first divide the frames into groups. For each block, we group frames via accumulation: we start at the first frame and keep adding frames to the group until the total number of measurements contained in the group is no greater than N. This is to prevent our linear system of equations from becoming overdetermined and to ensure that the scene doesn't change too much within a frame group. Accumulating frames this way forms larger groups when the average number of measurements in a frame is small and smaller groups when the average number of measurements in a frame is large. This naturally aligns with intuition about joint reconstruction: if a frame group has a smaller number of average measurements, it means there is very little texture complexity and/or very little motion and hence the common component will be more strongly shared amongst the group. The details of the frame grouping procedure are illustrated in Figure 3.3.

Omitting the block index b for clarity, the set of frame indices for frame group $g \in \xi$ is given by:

$$\Gamma_g = \{f_{(g-1)} + 1, \ldots, f_g\} \tag{3.22}$$

where f_g is the final frame in frame group g and $f_0 = 0$. Frame groups completely partition the set of all frames and do not overlap.

We remove the need for explicit reference frames by defining innovation components relative to an implicit reference frame; implicit reference frames are obtained by reconstructing the common component of all frames in a frame group. Inspired by [70], we reconstruct the common component by stacking measurements from all frames in the group:

$$\mathbf{y}_{(g,b)} = \begin{bmatrix} \mathbf{y}_{\mathsf{stack}(f_{(g-1)}+1,b)} \\ \vdots \\ \mathbf{y}_{\mathsf{stack}(f_g,b)} \end{bmatrix} \tag{3.23}$$

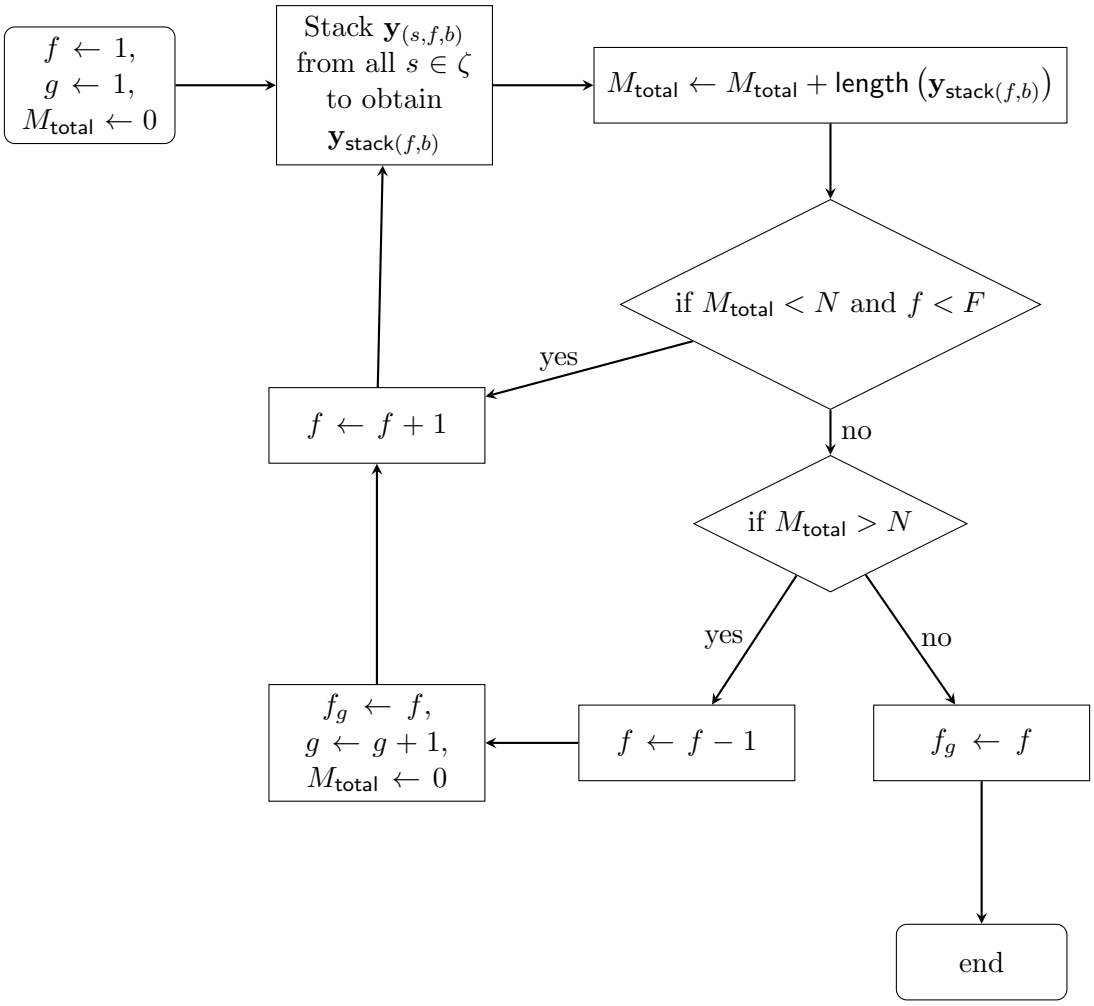

Figure 3.3: Frame grouping process for block $b \in \beta$.

and reconstructing the stack as if all the measurements were of the same frame. Note that $\mathbf{y}_{\mathsf{stack}(f,b)}$ was defined in (3.4). Reconstructing $\mathbf{y}_{(g,b)}$ effectively averages away the innovation components from all frames in the group stack, leaving only the common component; i.e.:

$$\hat{\boldsymbol{\alpha}}^{C}_{(g,b)} \leftarrow \mathsf{Recover}\Big(\mathbf{y}_{(g,b)}, \Phi_{(g,b)}\Big) \tag{3.24}$$

where:

$$\Phi_{(g,b)} = \begin{bmatrix} \Phi_{(f_{(g-1)},b)+1} \\ \vdots \\ \Phi_{(f_g,b))} \end{bmatrix} \tag{3.25}$$

and:

$$\Phi_{(f,b)} = \begin{bmatrix} \Phi_{1(1:M_0+M_{(1,f,b)})} \\ \vdots \\ \Phi_{S(1:M_0+M_{(S,f,b)})} \end{bmatrix} \tag{3.26}$$

Note that the compressive measurement matrices $\{\Phi_{1(1:M_0+M_{(1,f,b)})}, \ldots, \Phi_{S(1:M_0+M_{(S,f,b)})}\}$ will have different numbers of rows depending on the number of measurements allocated to their respective sensors, frames, and blocks. The common component is subtracted from each individual frame's measurements:

$$\tilde{\mathbf{y}}_{\mathsf{stack}(f,b)} \leftarrow \mathbf{y}_{\mathsf{stack}(f,b)} - \Phi_{(f,b)}\Psi\hat{\boldsymbol{\alpha}}^{C}_{(g,b)} \tag{3.27}$$

and the difference is reconstructed as the innovation component for that frame:

$$\hat{\boldsymbol{\alpha}}^{Q}_{(f,b)} \leftarrow \mathsf{Recover}\Big(\tilde{\mathbf{y}}_{\mathsf{stack}(f,b)}, \Phi_{(f,b)}\Big) \tag{3.28}$$

finally, the entire frame is reassembled:

$$\hat{\boldsymbol{\alpha}}_{(f,b)} = \hat{\boldsymbol{\alpha}}^{C}_{(g,b)} + \hat{\boldsymbol{\alpha}}^{Q}_{(f,b)} \tag{3.29}$$

and untransformed:

$$\hat{\mathbf{x}}_{(f,b)} = \Psi\hat{\boldsymbol{\alpha}}_{(f,b)} \tag{3.30}$$

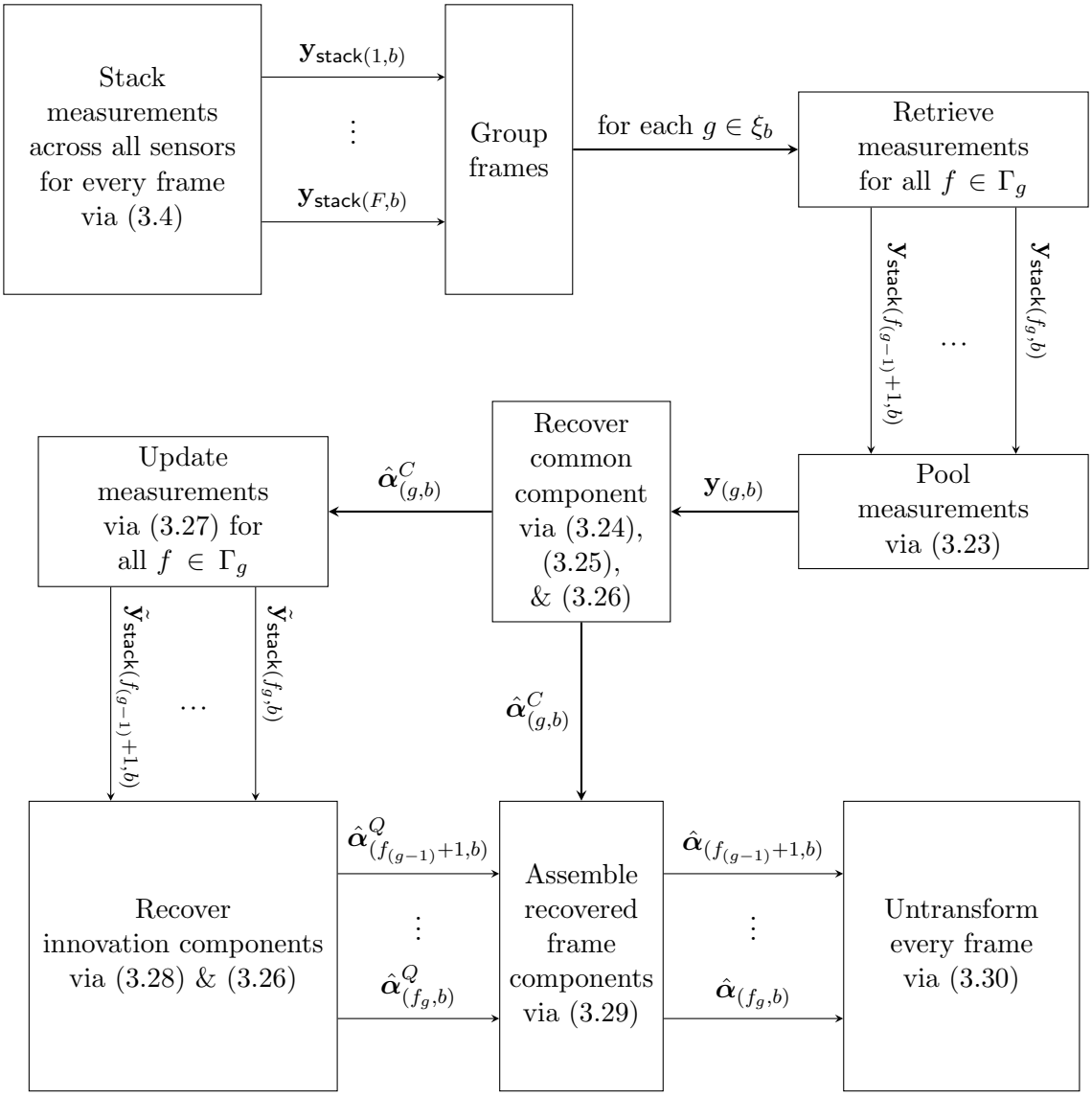

Figure 3.4: Frame reconstruction process for block $b \in \beta$.

for all $f \in \Gamma_g$, all $g \in \xi_b$, and all $b \in \beta$. A block diagram of the reconstruction process is shown in Figure 3.4. Since innovation components are not added to explicit reference frames, aliasing is confined within frame groups and alias accumulation is prevented. Once all blocks in all frames are reconstructed, we re-assemble them in the correct configuration and perform a simple blur across block borders to mitigate the perceptual effects of block-based sampling and reconstruction.

Chapter 4

Simulation Results

4.1 Quality evaluation metrics

The peak signal-to-noise ratio (PSNR) is an industry standard for evaluating image quality. It is based on the mean squared error (MSE) between an image and its approximation, and does not conform well to human visual perception. A better approach is to use the structural similarity index (SSIM) [71] which combines sophisticated judgements of luminance, contrast, and structure to produce a score that more accurately represents the way human subjects perceive image quality. In our simulations, we calculate both PSNR and SSIM and demonstrate that SSIM is indeed a better metric.

For all of the following simulations, we assume one frame-sensing cycle takes the same amount of time as 50 GPSR iterations. We force GPSR to terminate during sparsity estimation when a stopping tolerance of 1e–6 is reached or after 200 iterations, yielding a maximum sparsity estimate update delay of 4 frames. We use $\tau = \|\mathbf{y}\|_{\ell_2}/16\sqrt{M}$ for all GPSR calls, where $\mathbf{y}$ is the input vector and M is its length. During reconstruction, we set GPSR's stopping tolerance at 1e–7 and do not set an iteration limit. We use CDF 9/7 wavelets with three levels of dyadic decomposition.

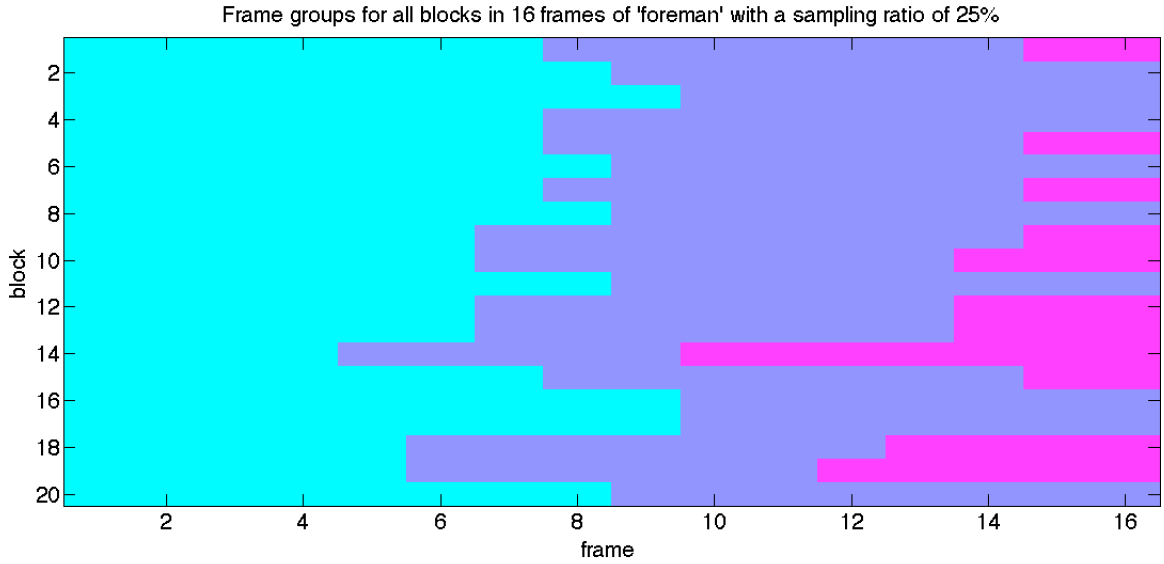

Figure 4.1: Frame groups for 16 frames of 'foreman' with a sampling ratio of 25%. The same color indicates the same frame group. Note that frame groups only exist within blocks; there are no inter-block frame groups. Smaller frame groups correspond to greater motion.

4.2 Implicit reference frames

In our first simulation, we show the results of jointly reconstructing groups of frames. We sample $F = 16$ frames of 'foreman' using $S = 8$ sensors with a sampling ratio of 25%. We use 4 texture-focused sensors and 4 motion-focused sensors. Each frame has dimensions 256×320 divided into $B = 20$ blocks of size 64×64; i.e., $N = 4096$. We set our measurement parameters to be $M_{\text{init}} = \lceil 0.4N/S \rceil = 205$, $M_0 = \lceil 0.04 \lfloor 25\% \times 256 \times 320 \rfloor / B \rceil = 41$, $M_{\text{sensor}} = \lfloor 25\% \times 256 \times 320/S \rfloor = 2560$, and $M_{\text{frame}} = SM_{\text{sensor}} = 20480$. These convoluted definitions ensure that we actually achieve a sampling ratio of exactly 25%; in a real application, we would simply sample at the maximum rate allowed by the DMD. For sparsity estimation at the central hub, we set $T = SM_{\text{init}} = 1640$.

Figure 4.1 shows frame groups for all blocks. The maximum number of frame groups for each block is 3, and the minimum is 2. Smaller frame groups correspond to faster motion in the video, while larger ones correspond to static areas.

Figure 4.2: An implicit reference frame and two innovation frames. The innovation frames have been luminance-inverted to show detail. The innovation frames are very sparse; the implicit reference frame resembles an average of all frames in the frame group. Note that each block in the implicit reference frame was reconstructed independently using different frame group sizes.

Figure 4.2 shows an implicit reference frame and two innovation frames. The implicit reference frame was reconstructed using the first frame group for each block; it appears blurred where motion occurs within a frame group. The innovation frames are very sparse.

4.3 Comparison of sensor modes

In our second simulation, we demonstrate the behaviour of the two sensor modes. We severely undersample $F = 16$ frames of 'foreman' using $S = 8$ sensors with a sampling ratio of 12%. Each frame has dimensions 256×320 divided into $B = 20$ blocks of size 64×64; i.e., $N = 4096$. We use values of $M_{\mathsf{init}} = \lceil N/4S \rceil = 128$, $M_0 = \lceil 0.04 \lfloor 12\% \times 256 \times 320 \rfloor / B \rceil = 20$, $M_{\mathsf{sensor}} = \lfloor 12\% \times 256 \times 320/S \rfloor = 1228$, and $M_{\mathsf{frame}} = SM_{\mathsf{sensor}} = 9824$. For sparsity estimation at the central hub, we set $T = SM_{\mathsf{init}} = 1024$.

Figure 4.3 illustrates the difference between texture-based versus motion-based sensing approaches. There is no hope of successfully recovering the entire 16 frames with a sampling ratio of only 12%, so we must trade-off diffuse texture quality with targeted motion quality. We see that a purely texture-based approach yields a higher PSNR, but a purely motion-based approach yields a higher SSIM. The texture-based approach recovers

Figure 4.3: Severe undersampling with different sensor modes. Left: original frame 16 of 'foreman'. Middle: reconstructed frame with all texture-focused sensors and a sampling ratio of 12%; PSNR = 13.7 dB, SSIM = 0.327. Right: reconstructed frame with all motion-focused sensors and a sampling ratio of 12%; PSNR = 11.4 dB, SSIM = 0.375. At this point in the video, the foreman is moving his head toward the upper right corner and there is a slight upward global camera motion.

significant portions of most blocks but leaves the foreman's face blurry. The motion-based approach recovers a clear picture of the foreman's face but fails to reconstruct a significant number of blocks in the frame.

4.4 Extended video sampling and reconstruction

In our final simulation, we sample and reconstruct all 150 frames of 'tennis' using $S = 8$ sensors with a sampling ratio of 40%. We use 6 texture-focused sensors and 2 motion-focused sensors. Each frame has dimensions 192×320 divided into $B = 15$ blocks of size 64×64; i.e., $N = 4096$. We take $M_{\mathsf{init}} = \lceil 0.4N/S \rceil = 205$, $M_0 = \lceil 0.04 \lfloor 40\% \times 192 \times 320 \rfloor / B \rceil = 66$, $M_{\mathsf{sensor}} = \lfloor 40\% \times 192 \times 320/S \rfloor = 3072$, and $M_{\mathsf{frame}} = SM_{\mathsf{sensor}} = 24576$. For sparsity estimation at the central hub, we set $T = SM_{\mathsf{init}} = 1640$.

Figure 4.4 illustrates the actual sparsity ratio K/N for each block in each frame of 'tennis'. All blocks were wavelet-transformed and coefficients with magnitudes below $\mathsf{Th} = 5$ were set to zero. The beginning of the video has a higher overall texture complexity than the rest of the video; this is caused by a high-frequency background texture. At frame 90

there is an abrupt scene change, after which the average sparsity becomes low except for a few extremely non-sparse blocks. The 'tennis' sequence features global camera movement in the first half and fast sports movement in the second half.

Figure 4.5 shows measurement allocations assigned to each block in each frame. The red areas in blocks 3 and 8 from frame 1–25 correspond to a ball bouncing up and down. The red streaks after frame 90 correspond to the fast motion of a tennis player. The motion-focused sensors react quickly to the scene change at frame 90, but the texture-focused sensors don't notice until they receive updated sparsity estimates a few frames later.

Figure 4.6 shows blockwise PSNR values for the reconstructed video after sampling according to the measurement allocations in Figure 4.5. PSNR is generally uniform within frames, but changes significantly from frame to frame. After frame 90 some reconstruction failures become obvious. Quality drops noticeably just before and right after frame 90; this is primarily due to the joint reconstruction of disparate frames, but it is also partially an effect of the sparsity estimate update delay. The PSNR trace is reminiscent of the actual sparsity traces in Figure 4.4.

Figure 4.7 shows the blockwise SSIM for the reconstructed video after sampling according to the measurement allocations in Figure 4.5. The SSIM is generally uniform within frames and between frames, especially in the first half of the video. Immediately prior to frame 90, blocks 5 and 10 experience extreme aliasing effects that are very obviously out of place in the scene. This phenomenon is shown in Figure 4.10. After frame 90, blocks 5 and 10 remain poorly reconstructed; however, the degradation is not as perceptually shocking since it coincides with a high complexity area of the scene. This can be observed in Figure 4.11. The PSNR and SSIM traces tell different stories about what happens around frame 90: the PSNR trace suggests that the quality in blocks 5 and 10 is better just before the scene change, which is not perceptually true. Clearly, SSIM is a more useful and intuitive metric for describing reconstructed video.

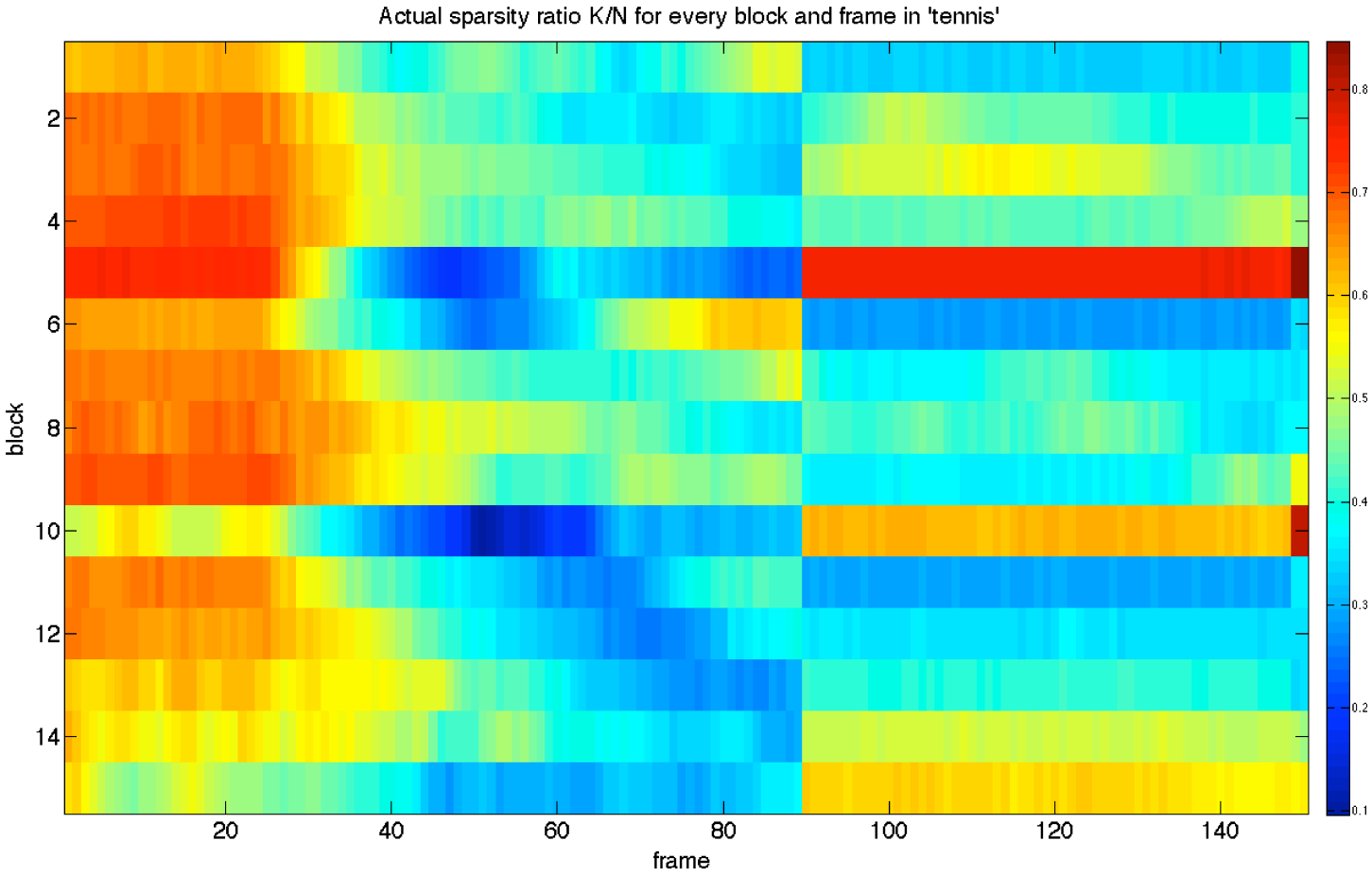

Figure 4.4: Actual sparsity ratio K/N for every block and frame in 'tennis'. Each frame is 192×320 pixels and each block is of size 64×64. All blocks were analyzed using CDF 9/7 wavelets with three levels of dyadic decomposition. Wavelet coefficients with magnitudes below $\mathsf{Th} = 5$ were set to zero.

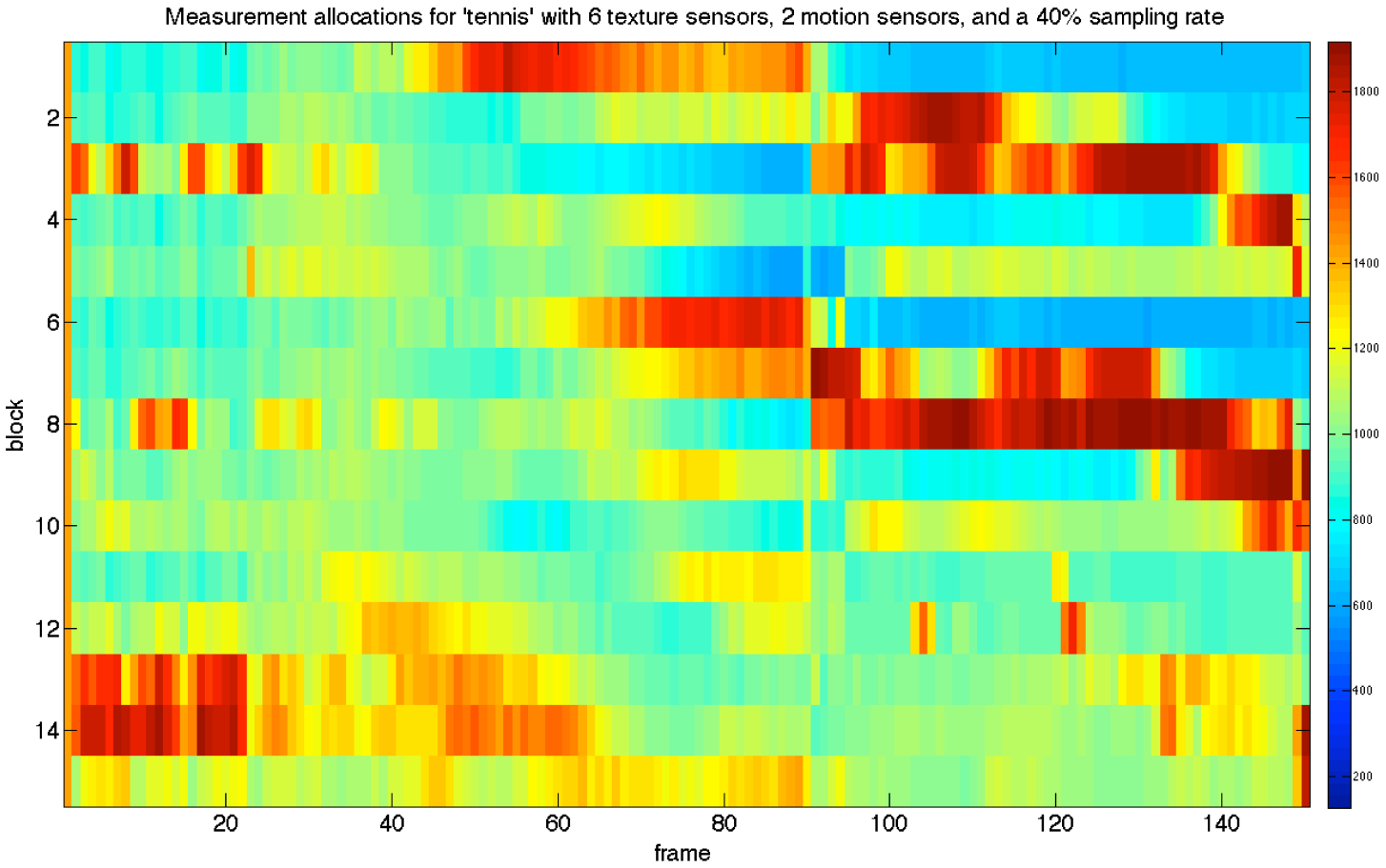

Figure 4.5: Measurement allocations for 'tennis' with six texture-focused sensors and two motion-focused sensors. The total sampling ratio for each frame is 40%. The prominent red streaks generally correspond to motion-intense sequences.

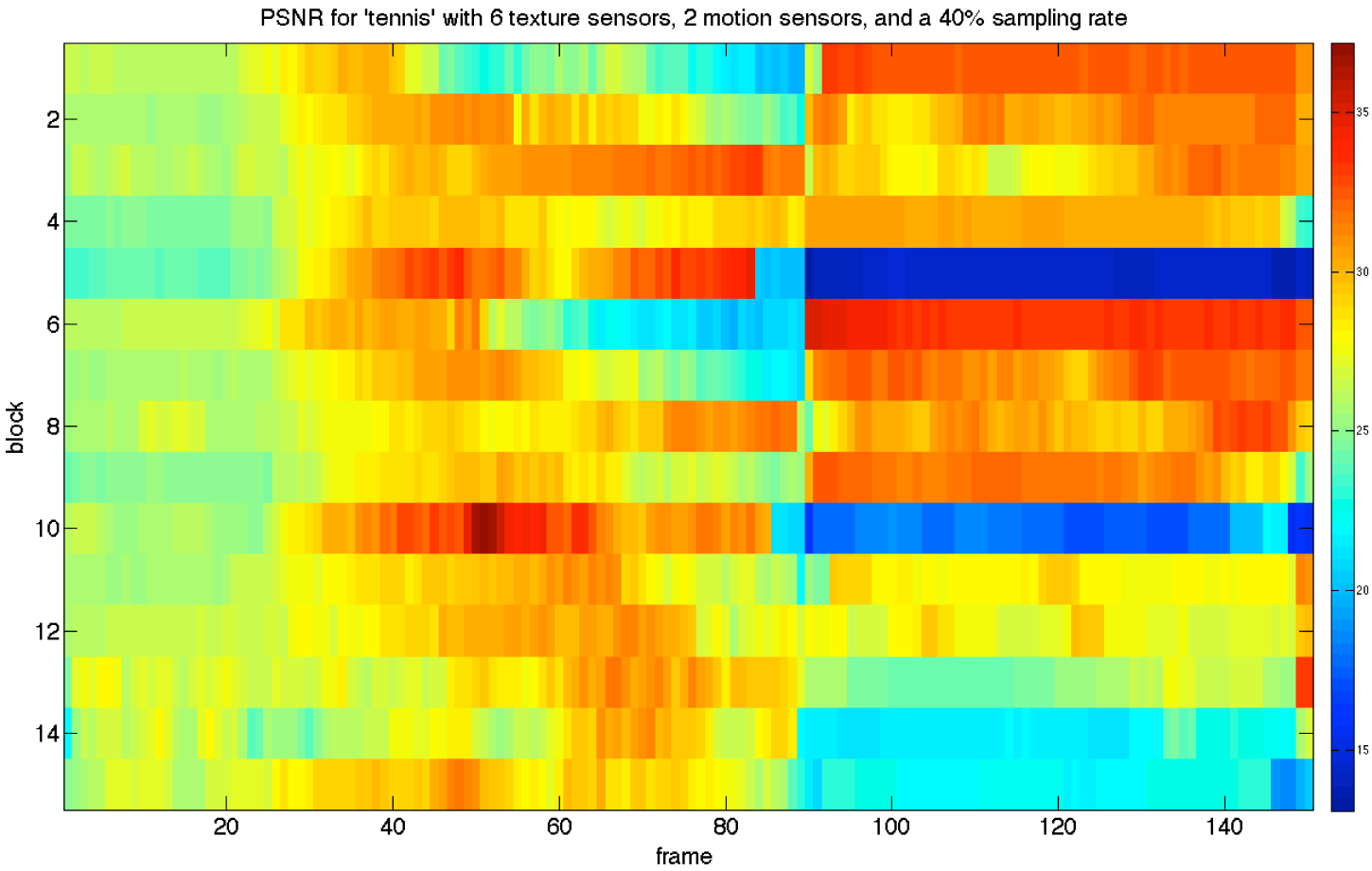

Figure 4.6: PSNR for 'tennis' sampled with measurement allocations from Figure 4.5 and jointly reconstructed using GPSR. The dark blue streaks are from a static region with high texture complexity that did not receive sufficient measurements.

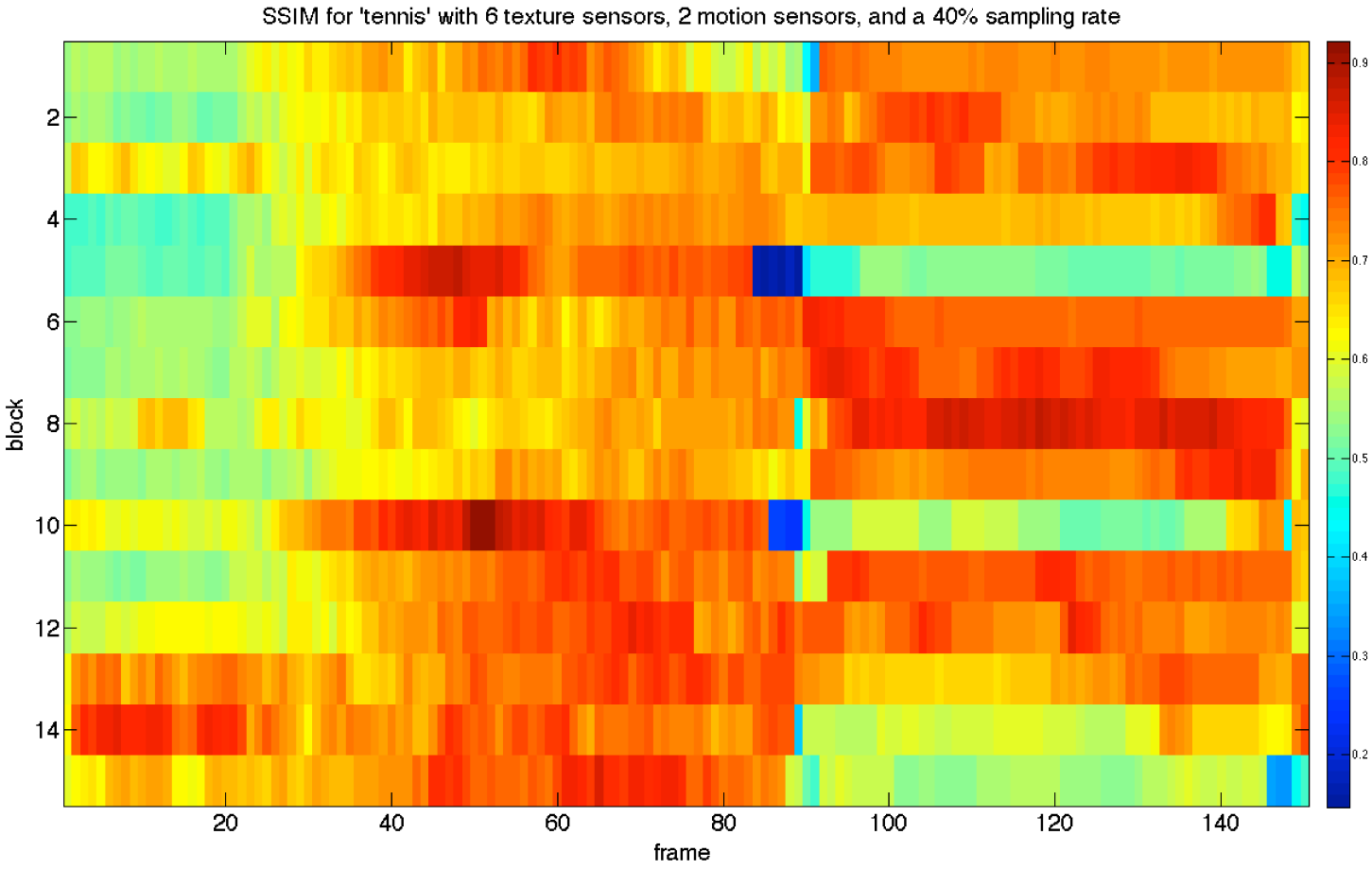

Figure 4.7: SSIM for 'tennis' sampled with measurement allocations from Figure 4.5 and jointly reconstructed using GPSR. The SSIM results conform more closely than PSNR to true perceptual quality. The quality is uniform throughout most frames in the video, with the exception of a few blocks that failed to completely reconstruct in the latter half.

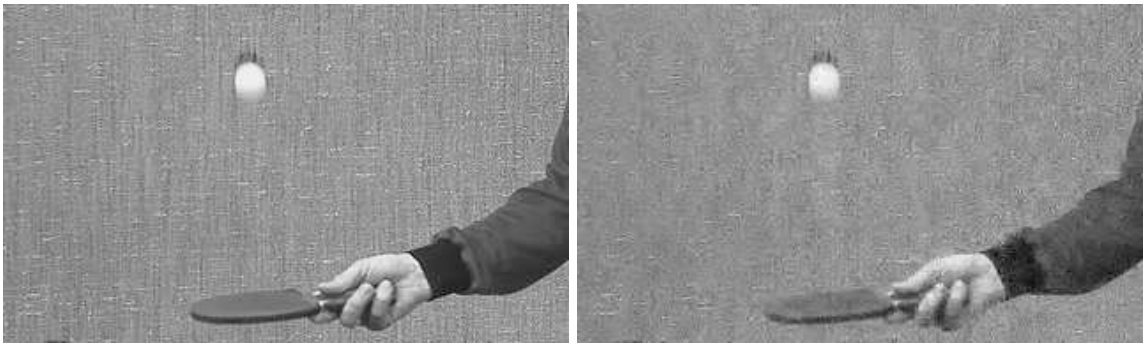

Figure 4.8: Original vs. reconstructed frame 23 of 'tennis'.
PSNR = 25.8 dB, SSIM = 0.604. The quality is uniform throughout all blocks. The sampling ratio is not quite adequate to recover the fine texture of the wall.

Figure 4.8 shows the reconstructed frame 23 of 'tennis'. The fine texture of the background is blurred due to a slight measurement insufficiency. The movement of the tennis player's hand and the bouncing ball are captured adequately.

Figure 4.9 shows frame 53 of 'tennis'. It is a good quality frame reconstruction; both PSNR and SSIM are high, though fine textures are still blurred.

Figure 4.10 shows frame 87 of 'tennis'. In this frame, there is a perceptually prominent reconstruction artifact caused by joint reconstruction of disparate frames.

Figure 4.11 shows frame 90 of 'tennis'. This frame is the first frame of a new scene. It is of generally poor quality due to joint reconstruction of disparate frames coupled with mis-allocated measurements from outdated sparsity estimates.

Figure 4.12 shows frame 140 of 'tennis'. The SSIM for frame 140 is higher than that of frame 23, even though frame 23 has a better PSNR; frame 23 is generally blurry while frame 140 has localized quality degradations. In frame 140, the sparsity estimates have been thoroughly updated but there are insufficient measurements available to fully reconstruct the high complexity static blocks in the upper right corner.

Figure 4.9: Original vs. reconstructed frame 53 of 'tennis'.
PSNR = 28.5 dB, SSIM = 0.738. The quality is uniform throughout all blocks and the sampling ratio is adequate.

Figure 4.10: Original vs. reconstructed frame 87 of 'tennis'.
PSNR = 24.0 dB, SSIM = 0.644. The two blocks in the upper right-hand corner are of poor quality because they were jointly reconstructed in a group with significantly different video frames.

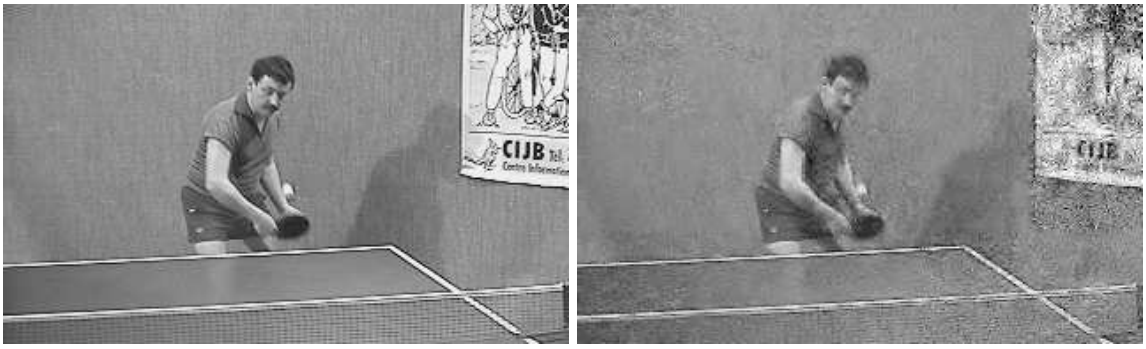

Figure 4.11: Original vs. reconstructed frame 90 of 'tennis'.
PSNR = 21.4 dB, SSIM = 0.610. The two blocks in the upper right-hand corner are of poor quality because their texture complexity is too high for the sampling ratio. The frame is generally of poor quality because the texture-focused sensors are still allocating measurements according to sparsity estimates from the previous scene.

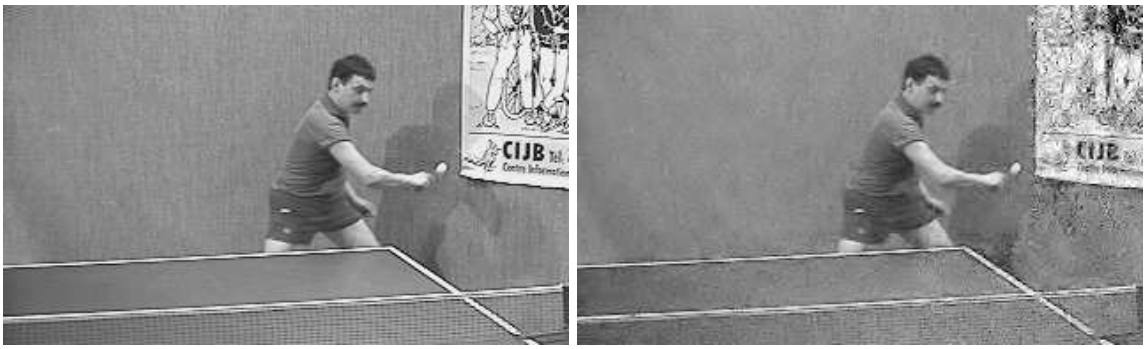

Figure 4.12: Original vs. reconstructed frame 140 of 'tennis'.
PSNR = 23.0 dB, SSIM = 0.704. The two blocks in the upper right-hand corner are of poor quality because their texture complexity is too high for the sampling ratio. The rest of the frame is of good quality because sparsity estimates have been refreshed and the tennis player is being adequately covered by the motion-focused sensors.

Chapter 5

Conclusion

5.1 Conclusion

Compressive video acquisition seeks to collect as many measurements of the most perceptually salient scene components as fast as possible—an impossible task using existing single-sensor technology. Many works on single-sensor compressive video acquisition indicate that video quality can be improved by simply adding more sensors in parallel, but ours is the only one that explicitly considers using multiple sensors to perform different acquisition tasks. Using separate sensors for texture-focused and motion-focused video acquisition minimizes the number of computations required at each sensor on each sensing cycle; if we had only one sensor, we would need to blend the modes and incur extra computation time during intra-frame measurement allocation. Hence, our framework minimizes wasted acquisition time while making the compressive video acquisition problem feasible.

In this work, we proposed a framework for compressive video acquisition with multiple sensors. We introduced two novel and distinct sensor modes: texture-focused mode and motion-focused mode. We saw that texture-focused sensors tended to increase the PSNR of the recovered signal while motion-focused sensors increased the SSIM. Motion is extremely

significant in human perception; this is reflected in the higher SSIM values observed when focusing on motion. Our framework was block-based for computational feasibility and to allow intra-frame measurement allocation. Although no explicit reference frames were obtained, a joint-sparse signal model allowed us to reconstruct implicit reference frames using the theory of distributed compressive sampling. For each group of frames, we recovered an implicit reference frame along with unique innovation components for each frame in the group—a novel approach to compressive video reconstruction. We allocated measurements according to where they were most needed in the video and we used every measurement during reconstruction.

5.2 Future directions

In our framework, the number of sensors in each mode is fixed for each video acquisition session. However, this is unlikely to be optimal; more research into adaptive sensor mode switching could improve the accuracy of the measurement allocation scheme. Furthermore, in reality, it is impossible for every sensor to have an identical view of the scene. Hence, one might investigate the consequences of each sensor having a different viewpoint; e.g., if each sensor's view differed by only a sub-pixel shift, super-resolution techniques might be used to increase the dimensions of the recovered frames. Finally, single-pixel cameras introduce large amounts of noise; this noise—often indistinguishable from small wavelet coefficients—is hard to remove, and could be a valuable research topic.

References

[1] H. Nyquist, "Certain topics in telegraph transmission theory," *Transactions of the AIEE*, vol. 47, pp. 363–390, 1928.

[2] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, July, October 1948.

[3] V. A. Kotelnikov, "On the transmission capacity of the 'ether' and of cables in electrical communications," in *Proceedings of the first All-Union Conference on the technological reconstruction of the communications sector and the development of low-current engineering*, (Moscow), 1933. Translated by C. C. Bissell and V. E. Katsnelson.

[4] C. Bissell, "Vladimir Aleksandrovich Kotelnikov: Pioneer of the sampling theorem, cryptography, optimal detection, planetary mapping," *IEEE Communications Magazine*, vol. 47, no. 10, pp. 24–32, 2009.

[5] J. F. Claerbout and F. Muir, "Robust modeling with erratic data," *Geophysics*, vol. 38, pp. 826–844, October 1973.

[6] H. L. Taylor, S. C. Banks, and J. F. McCoy, "Deconvolution with the ℓ_1 norm," *Geophysics*, vol. 44, pp. 39–52, January 1979.

[7] S. Levy and P. K. Fullagar, "Reconstruction of a sparse spike train from a portion of its spectrum and application to high-resolution deconvolution," *Geophysics*, vol. 46, pp. 1235–1243, September 1981.

[8] F. Santosa and W. W. Symes, "Linear inversion of band-limited reflection seismograms," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, pp. 1307–1330, October 1986.

[9] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, pp. 1289–1306, April 2006.

[10] E. J. Candès and J. Romberg, "Quantitative robust uncertainty principles and optimally sparse decompositions," *Foundations of Computational Mathematics*, vol. 6, pp. 227–254, April 2006.

[11] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, pp. 489–509, February 2006.

[12] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, pp. 4203–4215, December 2005.

[13] E. J. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Transactions on Information Theory*, vol. 52, pp. 5406–5425, December 2006.

[14] E. J. Candès, "Compressive sampling," in *Proceedings of the International Congress of Mathematicians*, (Madrid, Spain), 2006.

[15] "Information technology—Coding of audio-visual objects—Part 2: Visual," *ISO/IEC 14496-2:2004 (MPEG-4 Visual, Third edition)*, June 2004.

[16] D. Takhar, J. N. Laska, M. B. Wakin, M. F. Duarte, D. Baron, S. Sarvotham, K. F. Kelly, and R. G. Baraniuk, "A new compressive imaging camera architecture using optical-domain compression," in *Proceedings of Computational Imaging IV at SPIE Electronic Imaging*, vol. 6065, (San Jose, CA), pp. 43–52, January 2006.

[17] M. B. Wakin, J. N. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, "An architecture for compressive imaging," in *Proceedings of IEEE International Conference on Image Processing*, (Atlanta, GA), pp. 1273–1276, October 2006.

[18] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, pp. 83–91, March 2008.

[19] J. Zheng and E. L. Jacobs, "Video compressive sensing using spatial domain sparsity," *Optical Engineering*, vol. 48, no. 8, 2009.

[20] Texas Instruments, "DLP® 0.7 XGA 2xLVDS Type A DMD," *DLP7000 datasheet*, August 2012. Revised June 2013.

[21] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[22] E. J. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling." arXiv:math/0611957 [math.ST], November 2006.

[23] D. L. Donoho and M. Elad, "Optimally sparse representation in general (non-orthogonal) dictionaries via l1 minimization," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, pp. 2197–2202, March 2003.

[24] R. G. Baraniuk, M. A. Davenport, R. DeVore, and M. B. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, pp. 253–263, February 2007.

[25] E. J. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, pp. 589–592, March 2008.

[26] Y. C. Eldar and G. Kutyniok, eds., *Compressed sensing: Theory and applications*. Cambridge University Press, 2012.

[27] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, pp. 21–30, March 2008.

[28] E. J. Candès and Y. Plan, "A probabilistic and RIPless theory of compressed sensing," *IEEE Transactions on Information Theory*, vol. 57, pp. 7235–7254, August 2011.

[29] J. Cahill and D. G. Mixon, "Robust width: A characterization of uniformly stable and robust compressed sensing." arXiv:1408.4409 [cs.IT], August 2014.

[30] D. L. Donoho and J. Tanner, "Counting faces of randomly-projected polytopes when the projection radically lowers dimension," *Journal of the American Mathematical Society*, vol. 22, pp. 1–53, January 2009.

[31] D. L. Donoho and J. Tanner, "Counting the faces of randomly-projected hypercubes and orthants, with applications," *Discrete and Computational Geometry*, vol. 43, pp. 522–541, April 2010.

[32] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics*, vol. 59, pp. 797–829, June 2006.

[33] D. L. Donoho, "For most large underdetermined systems of equations, the minimal l1-norm near-solution approximates the sparsest near-solution," *Communications on Pure and Applied Mathematics*, vol. 59, pp. 907–934, March 2006.

[34] D. L. Donoho and J. Tanner, "Precise undersampling theorems," *Proceedings of the IEEE*, vol. 98, pp. 913–924, June 2010.

[35] M. F. Duarte, M. B. Wakin, D. Baron, and R. G. Baraniuk, "Universal distributed sensing via random projections," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, (New York, NY), pp. 177–185, 2006.

[36] D. Baron, M. F. Duarte, M. B. Wakin, S. Sarvotham, and R. G. Baraniuk, "Distributed compressive sensing," Tech. Rep. TREE-0612, Rice University, November 2006.

[37] M. F. Duarte, S. Sarvotham, D. Baron, M. B. Wakin, and R. G. Baraniuk, "Distributed compressed sensing of jointly sparse signals," in *Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems, and Computers*, pp. 1537–1541, October 2005.

[38] D. Baron, M. F. Duarte, S. Sarvotham, M. B. Wakin, and R. G. Baraniuk, "An information-theoretic approach to distributed compressed sensing," in *Proceedings of the 43rd Allerton Conference on Communication, Control, and Computing*, 2005.

[39] M. F. Duarte, M. B. Wakin, D. Baron, and S. Sarvotham, "Measurement bounds for sparse signal ensembles via graphical models," *IEEE Transactions on Information Theory*, vol. 59, pp. 4280–4289, April 2013.

[40] M. F. Duarte, S. Sarvotham, D. Baron, M. B. Wakin, and R. G. Baraniuk, "Theoretical performance limits for jointly sparse signals via graphical models," Tech. Rep. TREE-0802, Rice University, July 2008.

[41] E. J. Candès, Y. C. Eldar, D. Needell, and P. Randall, "Compressed sensing with coherent and redundant dictionaries," *Applied and Computational Harmonic Analysis*, vol. 31, pp. 59–73, July 2011.

[42] R. F. Marcia and R. M. Willett, "Compressive coded aperture video reconstruction," in *Proceedings of the European Conference on Signal Processing (EUPISCO)*, 2008.

[43] A. C. Sankaranarayanan, P. K. Turaga, R. G. Baraniuk, and R. Chellappa, "Compressive acquisition of dynamic scenes," in *Proceedings of the 11th European Conference on Computer Vision*, vol. 6311, pp. 129–142, September 2010.

[44] J. V. Shi, W. Yin, A. C. Sankaranarayanan, and R. G. Baraniuk, "Video compressive sensing for dynamic MRI." arXiv:1401.7715 [cs.CV], February 2014.

[45] M. L. Moravec, J. K. Romberg, and R. G. Baraniuk, "Compressive phase retrieval," in *Proceedings of the SPIE: Wavelets XII*, vol. 6701, September 2007.

[46] A. C. Sankaranarayanan, C. Studer, and R. G. Baraniuk, "CS-MUVI: Video compressive sensing for spatial-multiplexing camera," in *IEEE International Conference on Computational Photography*, (Seattle, WA), pp. 1–10, April 2012.

[47] M. B. Wakin, J. N. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, "Compressive imaging for video representation and coding," in *Proceedings of Picture Coding Symposium*, (Beijing), April 2006.

[48] Y. L. Montagner, E. Angelini, and J.-C. Olivo-Marin, "Video reconstruction using compressed sensing measurements and 3D total variation regularization for bio-imaging applications," in *Proceedings of IEEE International Conference on Image Processing*, (Orlando, FL), pp. 917–920, October 2012.

[49] Z. Liu, H. V. Zhao, and A. Y. Elezzabi, "Block-based adaptive compressed sensing for video," in *Proceedings of IEEE International Conference on Image Processing*, (Hong Kong), pp. 1649–1652, September 2010.

[50] Z. Liu, A. Y. Elezzabi, and H. V. Zhao, "Maximum frame rate video acquisition using adaptive compressed sensing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, pp. 1704–1718, November 2011.

[51] I. Wahidah, A. B. Suksmono, Hendrawan, and T. L. R. Mengko, "Compressive sampling for digital video signal compression involving dynamic sparsity," in *7th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, pp. 46–50, October 2012.

[52] L. Gan, "Block compressed sensing of natural images," in *Proceedings of the International Conference on Digital Signal Processing*, (Cardiff, UK), 2007.

[53] S. Mun and J. E. Fowler, "Residual reconstruction for block-based compressed sensing of video," in *Data Compression Conference (DCC)*, pp. 183–192, March 2011.

[54] D. Baron, S. Sarvotham, and R. G. Baraniuk, "Bayesian compressive sensing via belief propagation," *IEEE Transactions on Signal Processing*, vol. 58, pp. 269–280, July 2009.

[55] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 7th ed., 2004.

[56] J. A. Tropp, "Just relax: Convex programming methods for identifying sparse signals in noise," *IEEE Transactions on Information Theory*, vol. 52, pp. 1030–1051, March 2006.

[57] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Transactions on Information Theory*, vol. 50, pp. 2231–2242, October 2004.

[58] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, pp. 4655–4666, December 2007.

[59] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, pp. 301–321, May 2009.

[60] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 58, pp. 1094–1121, February 2012.

[61] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Simultaneous sparse approximation via greedy pursuit," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, pp. v/721–v/724, March 2005.

[62] M. F. Duarte and R. G. Baraniuk, "Compressive sensing with biorthogonal wavelets via structured sparsity," in *Workshop on Signal Processing with Adaptive Sparse Representations (SPARS)*, June 2011.

[63] W. Wang, M. J. Wainwright, and K. Ramchandran, "Information-theoretic limits on sparse signal recovery: Dense versus sparse measurement matrices," *IEEE Transactions on Information Theory*, vol. 56, pp. 2967–2979, June 2010.

[64] L. Gan, T. T. Do, and T. D. Tran, "Fast compressive imaging using scrambled block Hadamard ensemble," in *European Signal Processing Conference*, (Lausanne, Switzerland), August 2008.

[65] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, "On the Lambert W function," *Advances in Computational Mathematics*, vol. 5, no. 1, pp. 329–359, 1996.

[66] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, pp. 36–58, September 2001.

[67] M. Andersen, J. Dahl, and L. Vandenberghe, "CVXOPT: A python package for convex optimization." http://abel.ee.ucla.edu/cvxopt.

[68] E. J. Candès and J. Romberg, "ℓ_1-MAGIC: Recovery of sparse signals via convex programming," tech. rep., California Institute of Technology, October 2005.

[69] 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 Journal of Selected Topics in Signal Processing*, vol. 1, pp. 586–597, December 2007.

[70] S. R. Schnelle, J. N. Laska, C. Hegde, M. F. Duarte, M. A. Davenport, and R. G. Baraniuk, "Texas Hold 'Em algorithms for distributed compressive sensing," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Dallas, TX), pp. 2886–2889, March 2010.

[71] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, April 2004.