Increasing temporal, structural, and spectral resolution in images using exemplar–based priors

by

Jason Holloway

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Master of Science

APPROVED, THESIS COMMITTEE:

Ashok Veeraraghavan, Chair
Assistant Professor
Electrical and Computer Engineering

Richard Baraniuk
Victor E. Cameron Professor
Electrical and Computer Engineering

Kevin Kelly
Associate Professor
Electrical and Computer Engineering

Houston, Texas
April 2013
ABSTRACT

Increasing temporal, structural, and spectral resolution in images using exemplar–based priors

by

Jason Holloway

In the past decade, camera manufacturers have offered smaller form factors, smaller pixel sizes (leading to higher resolution images), and faster processing chips to increase the performance of consumer cameras. However, these conventional approaches have failed to capitalize on the spatio–temporal redundancy inherent in images, nor have they adequately provided a solution for finding 3D point correspondences for cameras sampling different bands of the visible spectrum. In this thesis, we pose the following question—given the repetitious nature of image patches, and appropriate camera architectures, can statistical models be used to increase temporal, structural, or spectral resolution? While many techniques have been suggested to tackle individual aspects of this question, the proposed solutions either require prohibitively expensive hardware modifications and/or require overly simplistic assumptions about the geometry of the scene.

We propose a two-stage solution to facilitate image reconstruction; 1) design a linear camera system that optically encodes scene information and 2) recover full scene information using prior models learned from statistics of natural images. By leveraging the tendency of small regions to repeat throughout an image or video, we are able to learn prior models from patches pulled from exemplar images. The
quality of this approach will be demonstrated for two application domains, using low-speed video cameras for high-speed video acquisition and multi-spectral fusion using an array of cameras. We also investigate a conventional approach for finding 3D correspondence that enables a generalized assorted array of cameras to operate in multiple modalities, including multi-spectral, high dynamic range, and polarization imaging of dynamic scenes.
I have been the beneficiary of the generous support while at Rice and I would be remiss should I not make this known. Of course, I thank my advisor, Ashok Veeraraghavan, for his help and guidance over the past three years. It goes without saying that I owe a huge debt of gratitude to him and to the knowledgeable (and jovial) team with whom I have been collaborating: Aswin Sankaranarayanan, Kaushik Mitra, and Sanjeev Kopal. Watching the lab grow from an idea to a vacant room and finally to a fully furnished and populated workspace has been a memorable experience. I am grateful for the support of my family, the members of my lab, the new friends I made here at Rice, and the small group of us that venture outside of the hedges for lunch every payday.

Thank you all.
Contents

Abstract ii
List of Illustrations vii
List of Tables ix

1 Introduction 1
1.1 Motivation ................................................. 2
1.2 Layout of this thesis ....................................... 5

2 Flutter Shutter Video Camera 7
2.1 Introduction ............................................... 8
2.2 Related Work .............................................. 10
2.3 The Flutter Shutter Video Camera ....................... 13
   2.3.1 Notation and problem statement ................... 14
2.4 Reconstruction Algorithms ............................. 17
   2.4.1 Video Priors ........................................ 17
   2.4.2 Total Variation (TV) of XT and YT Slices ....... 19
   2.4.3 Data driven dictionary-based reconstruction ...... 20
2.5 Experiments .............................................. 24
2.6 Discussions and Conclusions ........................... 28

3 Generalized Assorted Cameras 30
3.1 Introduction .............................................. 31
   3.1.1 Prior Work ....................................... 33
3.2 From Assorted Pixels to Assorted Cameras ............ 36
3.3 Establishing Point Correspondences ........................................ 38
  3.3.1 Internal and External Calibration .................................. 39
  3.3.2 Dense 3D Reconstruction ........................................... 40
  3.3.3 View Interpolation .................................................. 44
3.4 Direct Multi-spectral Imaging ............................................. 44
3.5 Multiplexed Multi-spectral Imaging ...................................... 49
3.6 HDR & Polarization Imaging ................................................ 52
3.7 Discussion and Future work ............................................... 55

4 Dense Point Correspondence Across Color Channels 57
  4.1 Introduction .............................................................. 57
  4.2 Related Work ............................................................ 59
  4.3 Establishing Point Correspondences .................................. 60
    4.3.1 Dense 3D Depth Reconstruction .................................. 60
  4.4 Image Reconstruction .................................................. 62
    4.4.1 Recovering the Composite Image .................................. 63
  4.5 RGBY Fusion .............................................................. 64

5 Conclusions ................................................................. 71

Bibliography ................................................................. 74
Illustrations

2.1 Flutter Shutter Video Camera (FSVC) .................................................. 9
2.2 FSVC Architecture .............................................................................. 15
2.3 Total Variation Prior .......................................................................... 16
2.4 Results of Flutter Shutter Video Camera ............................................. 18
2.5 Data Dependent Algorithm ................................................................... 21
2.6 Reconstruction quality vs compression ............................................... 23
2.7 Reconstruction quality vs. noise ......................................................... 24
2.8 Experimental Results ........................................................................... 25
2.9 Comparison of the qualitative properties of compressive
   computational video cameras ................................................................. 28

3.1 Assorted Cameras .................................................................................. 31
3.2 View interpolation in video .................................................................... 37
3.3 Calibration Quality ................................................................................ 40
3.4 Dense 3D Reconstruction ...................................................................... 43
3.5 Warping to the Reference Camera ....................................................... 44
3.6 Prototype GAC ...................................................................................... 45
3.7 Spectral Measurements ......................................................................... 46
3.8 Multi-spectral quality ........................................................................... 48
3.9 Spectral Content Comparison ............................................................... 49
3.10 Color Chart Recovery .......................................................................... 52
3.11 HDR Imaging ....................................................................................... 53
3.12 Polarization and HDR ............................................. 54

4.1 RGBY Imaging ..................................................... 65
4.2 RGBY Imaging ..................................................... 66
4.3 RGBY Fusion ....................................................... 67
4.4 Digital refocusing ................................................. 69
Tables

5.1 Comparison of compression vs reconstruction PSNR in dB . . . . . . 72
Chapter 1

Introduction

Limited bandwidth is the restricting factor at the heart of many problems in camera design. For example: recording high speed video, capturing the light field video of a scene, imaging in refractive media, imaging beyond the visible spectrum, and multi-spectral imaging of dynamic scenes, are all challenging problems that are inadequately solved using current camera technology. In the case of high-speed video, the bandwidth limitation is straightforward; a video camera cannot record more information than it can transmit to a storage device for writing. For other applications the bandwidth limitation manifests in other camera elements; imaging beyond the visible spectrum often requires costly photodetectors. Each “pixel” of a non-visible image sensor costs considerably more than a silicon based sensor used in consumer cameras. Therefore the limited bandwidth of the sensor material imposes design restrictions on the manufactured camera.

In this thesis, we will focus on two of the aforementioned applications—recording high speed video and multi-spectral imaging of dynamic scenes. Both of these problems can solved using a brute force approach of adding more camera elements. Many of the current designs for high-speed video cameras follow this approach by adding more sensors [1,2] or faster processing elements. Instead we will present a method for capturing high-speed video using a low-speed camera by compressing incoming light before the data is read off of the sensor. Likewise, one possible solution for recording multi-spectral video is to use an array of cameras where a different narrowband
filter can be placed in front of each camera. To this end we will present a naive approach that highlights the strengths of using camera arrays (including the ability to record multi-spectral video) as well as more recent research efforts that seek to reduce the number of cameras required in the array. For both high-speed imaging and the second multi-spectral approach we can employ patch-based priors to help guide the reconstruction of the true scene given the observed images from our camera designs.

1.1 Motivation

The following high level motivation section serves to motivate the problems faced in this thesis on an intuitive level. Comprehensive technical motivation is offered in the chapter pertaining to each problem.

**High-speed imaging:** Images and video exhibit a high level of spatio-temporal redundancy. One of the first compression algorithms for digital video, MPEG-1, was able to achieve an impressive 26 : 1 compression ratio [3]. Modern video compression standards (i.e. MPEG-4) can easily achieve compression that is $4 \times$ better than MPEG-1 for a compression ratio greater than 100 : 1 [4]. While the ability of these algorithms to reduce the spatio-temporal redundancy in videos is impressive, this compression must be performed post-capture. That is, compression can only occur after all of the sensor data has been read off of the camera onto the processing unit. This creates a bottleneck at the readout stage which limits the overall speed of the camera. One solution is to reduce the spatial resolution of the video so that fewer pixels are transferred per frame. Halving the spatial resolution in both the $x$ and $y$ axes results in a 75% reduction in bandwidth which can be reallocated to delivering faster video recording. However, this suffers from two drawbacks. Firstly, there is
the obvious reduction in image quality due to loss of spatial resolution and secondly, recording more frames per second reduces the amount of light that is allowed to hit the sensor causing the resulting video to have a low signal-to-noise ratio (SNR).

Instead of compressing video data post-capture, what if it were possible to encode incoming light such that information about high-speed events was preserved? Given that a patch of a video (e.g. $18 \times 18 \times 24$ frames) is highly redundant, it stands to reason that the true high-speed patch lies on a lower-dimensional subspace. If this subspace could be learned, or if properties of the subspace were known, it may be possible to recover the high-speed patch given just the projection of the high-speed video onto the low-speed subspace.

Optically encoding light as it enters the camera requires a modification of the existing camera hardware. We seek to constrain this modification to the bare minimum necessary to recover a high-speed video. In practice, it suffices to have control of the shutter state during each frame. Since many modern cameras (e.g. cell phone cameras) use electronic shutters this functionality is already included in many cameras and may be able to be added to others with a firmware update. Two proposed methods for recovering the high-speed video from the encoded low-speed video are presented in Chapter 2.

**Multi-spectral imaging:** Most consumer cameras capture color images using band-pass filters corresponding to the red, green, and blue wavelengths of light. These filters are commonly placed on the sensor in a Bayer array consisting of a repeating $2 \times 2$ grid of two green filters, one red filter, and one blue filter. For everyday snapshots this arrangement works well; scenes are captured with reasonable color fidelity and the incurred loss in spatial resolution is mitigated when displaying a 2 megapixel (or
larger) image on a 1 megapixel screen. Suppose that the task at hand were not as simple as snapping a keepsake and that true color fidelity were important. Given two metameric objects, objects who appear to be the same color but have different spectral compositions, the only way to differentiate these objects requires a dense sampling of the visible spectrum. Other spectrum discriminating applications include monitoring blood flow by recording in a narrow portion of the spectrum that corresponds to green light, selecting ripe fruit based on the spectral composition, and monitoring gas pipelines for leaks. These tasks require the use of a multi-spectral imaging system that can sample narrow bands of the visible spectrum.

A simple approach to measuring many narrow bands simultaneously is to use an array of cameras where each camera in the array is fitted with a narrowband filter. Invoking such a camera array successfully solves the sampling problem but introduces the complication of aligning the views of the different cameras. Consider a rectified two camera system such as those used to generate 3D stereo movies. The two cameras are parallel and are not rotated with respect to each other. Parallax between the two cameras helps establish the depth of scene points. Points at infinity will project to the same pixel location in each camera while points that are far from the cameras (but not at infinity) will project to pixel locations that are close in each camera. That is, the disparity in pixel location in both cameras is small for points that are far away. Conversely, scene points that are close to the two cameras will project to pixel locations that are far apart, exhibiting a large disparity. This scene-dependent shift means that aligning multiple cameras in the array cannot be accomplished with a simple mapping operator.

Aligning all of the views to a central camera viewpoint requires warping the outlying images using a computed depth map of the scene. Robust methods for computing
scene depth exist for cameras that record the same color channels, but no such methods exist if the cameras record different channels. One solution is to leave a subset of the cameras in the array unfiltered, therefore the cameras in this subset can be used for depth estimation. This idea is explored in Chapter 3. Alternatively, it is possible to use a model of image patch distribution to find correspondence across color channels. Consider the case of using an RGB color camera to take pictures of natural scenes. If patches in the aligned image are treated as Gaussian random variables, it is possible to compute the statistical properties of a prototypical scene patch. These statistics can then be used to compute how likely it would be to see a given scene patch, a particularly useful trait that can be used to find the correct disparity for scene points. In Chapter 4 we explore this concept using a $2 \times 2$ camera array that consists of red, green, blue, and all pass filtered cameras.

1.2 Layout of this thesis

The remainder of this thesis is divided into three independent chapters followed by a concluding statement. Chapter 2 proposes a method for optically encoding high-speed data onto a low-speed video sensor and recovering the high-speed data in post-processing. Two methods for recovering data will be presented. For general scenes a total variation based recovery algorithm is presented and for scenes that are well approximated by locally linear motion we present a data driven reconstruction that relies on the statistics of video patches.

Chapter 3 and Chapter 4 are primarily concerned with capturing multi-spectral images and video using camera arrays. In Chapter four a case is made for the use of camera arrays in a general framework. A majority of the elements in the array have different optical filters while a subset of the cameras are left unfiltered to establish
the 3D geometry of the scene. Applications in multi-spectral imaging, high dynamic range imaging, and polarization imaging are shown.

In Chapter 4 we investigate the possibility of removing redundant elements in the camera array when collecting multi-spectral data. That is, each camera in the array is fitted with a different optical filter and 3D geometry is matched across color channels. This is a particularly challenging task as each camera not only sees the scene from a different viewpoint (introducing depth dependent shifts), but each camera records different intensities for objects in the scene. A patch-based prior model is employed to compute the 3D scene geometry across color channels.
Chapter 2

Flutter Shutter Video Camera

Video cameras are invariably bandwidth limited and this results in a trade-off between spatial and temporal resolution. Advances in sensor manufacturing technology have tremendously increased the available spatial resolution of modern cameras while simultaneously lowering the costs of these sensors. In stark contrast, hardware improvements in temporal resolution have been modest. One solution to enhance temporal resolution is to use high bandwidth imaging devices such as high speed sensors and camera arrays. Unfortunately, these solutions are expensive. An alternate solution is motivated by recent advances in computational imaging and compressive sensing. Camera designs based on these principles, typically, modulate the incoming video using spatio-temporal light modulators and capture the modulated video at a lower bandwidth. Reconstruction algorithms, motivated by compressive sensing, are subsequently used to recover the high bandwidth video at high fidelity. Though promising, these methods have been limited since they require complex and expensive light modulators that make the techniques difficult to realize in practice.

In this chapter, we show that a simple coded exposure modulation is sufficient to reconstruct high speed videos. We propose the Flutter Shutter Video Camera (FSVC) in which each exposure of the sensor is temporally coded using an independent pseudo-random sequence. Such exposure coding is easily achieved in modern sensors and is already a feature of several machine vision cameras. We also develop two algorithms for reconstructing the high speed video; the first based on minimizing the
total variation of the spatio-temporal slices of the video and the second based on a data driven dictionary based approximation. We perform evaluation on simulated videos and real data to illustrate the robustness of our system.

2.1 Introduction

Video cameras are inarguably the highest bandwidth consumer device that most of us own and recent trends are driving that bandwidth higher as manufacturers develop sensors with even more pixels and faster sampling rates. Even in a mobile phone, the instantaneous bandwidth of acquiring videos far exceeds the bandwidth requirements of text, voice and data services. The escalating demand is forcing manufactures to increase the complexity of the readout circuit to achieve a greater bandwidth. Unfortunately, since the readout circuit shares area with the light sensing element of sensors, this usually results in smaller pixel fill-factors and consequently reduced signal-to-noise ratio. Further, additional circuit elements result in increased cost. This is why even high resolution digital cameras capture videos at reduced spatial resolution so that the effective bandwidth is constrained. While this spatio-temporal resolution trade-off seems fundamental, the fact that videos have redundancies implies that this bandwidth limit is artificial and can be surpassed. In fact, it is this redundancy of videos that enables compression algorithms to routinely achieve 25–50x compression without any perceivable degradation.

Advances in computational cameras and compressive sensing have led to a series of compressive video acquisition devices that exploit this redundancy to reduce the bandwidth required at the sensor. The common principle behind all of these designs is the use of spatio-temporal light modulators and/or exposure control in the imaging system so that the captured video is a multiplexed version of the original video voxels.
Figure 2.1: **Flutter Shutter Video Camera (FSVC):** The exposure duration of each frame is modulated using an independent pseudo-random binary sequence. The captured video is a multiplexed version of the original video voxels. Priors about the video are used to then reconstruct the high speed video from FSVC observations.

If the multiplexing is suitably controlled, then appropriate reconstruction algorithms that exploit the redundancy in the videos can be used to recover the high resolution/high frame-rate video. One such technique is the single pixel camera [5] which reduced the bandwidth required for image acquisition using a random spatial modulation. More recently, there have been a series of imaging architectures [6–12] that have proposed various alternative ways to compressively sense high speed/resolution videos. While many of these techniques show promising results, they mostly suffer from the same handicap: the hardware modifications required to enable these systems is either expensive/cumbersome or is currently unavailable. In this chapter, we propose the Flutter Shutter Video Camera (FSVC), in which the only light modu-
lation is the coded control of the exposure duration in each frame of the captured video. FSVC is, in spirit, similar to many of these above-mentioned techniques, but unlike those techniques it is a simple modification to current digital sensors. In fact, not only are there many machine vision cameras that already have this ability (e.g., Point Grey Dragonfly2), almost all CMOS and CCD sensors can be adapted easily to control the exposure duration.

**Contributions:** The contributions of this chapter are

- We show that simple exposure coding in a video camera can be used to recover high speed video sequences while reducing the high bandwidth requirements of traditional high speed cameras.

- We show that data independent and data-dependent video priors can be used for recovering the high speed video from the captured FSVC frames.

- We discuss the invertibility and compression achievable by various multiplexing schemes for acquiring high speed videos.

### 2.2 Related Work

The proposed FSVC relies on numerous algorithmic and architectural modifications to existing techniques.

**High speed cameras:** Traditional high speed cameras require sensors with high light sensitivity and massive data bandwidth—both of which add significantly to the cost of the camera. The massive bandwidth, caused by the large amount of data sensed over a short duration, typically requires a dedicated bus to the sensor [13]. High-performance commercial systems such as the FastCam SA5 can reach a frame-
rate of about 100K fps at spatial resolution of $320 \times 192$, but cost about $300K$ [13]. In contrast, the FSVC significantly mitigates the dual challenges of light sensitive sensors and data bandwidth by integrating over a much longer exposure time; this naturally increases the signal-to-noise ratio and reduces the bandwidth of the sensed data.

**Motion deblurring:** The ideas in this chapter are closely related to computational cameras first developed for the motion deblurring problem. In motion deblurring [14–16], the goal is to recover a sharp image and the blurring kernel given a blurred image. Of particular interest, is the Flutter Shutter Camera [14] where the point spread function of the motion blur is shaped by coding the shutter during the exposure; this removes nulls in the point spread function and regularizes the otherwise ill-conditioned forward imaging process. An alternative architecture [15] uses parabolic motion of the sensor to achieve a well conditioned point spread function. While these approaches are only applicable to a small class of scenes that follow a motion model, there is a fundamental difference between video sensing and deblurring. Deblurring seeks to recover a *single* image and an associated blur kernel that encodes this motion. In contrast, video sensing attempts to recover multiple frames and hence, seeks a richer description of the scene and provides the ability to handle complex motion in natural scenes.

**Temporal super-resolution:** Video compressive sensing (CS) methods rely heavily on temporal super-resolution methods. Mahajan et al. [17] describe a method for plausible image interpolation using short exposure frames. But such interpolation based techniques suffer in dimly lit scenes and cannot achieve large compression factors.
Camera arrays: There have been many methods to extend ideas in temporal super-resolution to multiple cameras—wherein the spatial-temporal tradeoff is replaced by a camera-temporal tradeoff. Shechtman et al. [18] used multiple cameras with staggered exposures to perform spatio-temporal super-resolution. Similarly, Wilburn et al. [1] used a dense array of several 30 fps cameras to recover a 1000 fps video. Agrawal et al. [19] improved the performance of such staggered multi-camera systems by employing per-camera flutter shutter. While capturing high speed video using camera arrays produces high quality results (especially for scenes with little or no depth variations), such camera arrays do come with significant hardware challenges. Another related technique is that of Ben-Ezra and Nayar [20] who built a hybrid camera that uses a noisy high frame rate sensor to estimate the point spread function for deblurring a high resolution blurred image.

Compressive sensing of videos: There have been many novel imaging architectures proposed for the video CS problem. These include architectures that use coded aperture [21], a single pixel camera [5], global/flutter shutter [6, 22] and per-pixel coded exposure [8, 9].

For videos that can be modeled as a linear dynamical system, [7] uses a single pixel camera to compressively acquire videos. While this design achieves a high compression at sensing, it is limited to a rather small class of videos that can be modeled as linear dynamical. In [6], the flutter shutter (FS) architecture is extended to video sensing and is used to build a camera system to capture high-speed periodic scenes. Similar to [7], the key drawback of [6] is the use of a parametric motion model which severely limits the variety of scenes that can be captured. The video sensing architecture proposed by Harmany et al. [22], employs a coded aperture and an FS
to achieve CS “snapshots” for scenes with incoherent light and high signal-to-noise ratio. In contrast, the proposed FSVC, which also employs an FS, can be used to sense and reconstruct arbitrary videos.

Recently, algorithms that employ per-pixel shutter control have been proposed for the video CS problem. Bub et al. [12] proposed a fixed spatio-temporal trade-off for capturing videos via per-pixel modulation. Gupta et al. [11] extended the notion to flexible voxels allowing for post-capture spatio-temporal resolution trade-off. Gu et al. [10] modify CMOS sensors to achieve a coded rolling shutter that allows for adaptive spatio-temporal trade-off. Reddy et al. [8] achieve per-pixel modulation through the use of an LCOS mirror to sense high-speed scenes; a key property of the associated algorithm is the use of optical flow-based reconstruction algorithm. In a similar vein, Hitomi et al. [9] use per-pixel coded exposure, but, with an overcomplete dictionary to recover patches of the high speed scene. The use of per-pixel coded exposure leads to powerful algorithms capable of achieving high compressions even for complex scenes. Yet, hardware implementation of the per-pixel coded exposure is challenging and is a significant deviation from current commercial camera designs. In contrast, the FSVC only requires a global shutter control; this greatly reducing the hardware complexity needed as compared to systems requiring pixel-level shutter control. Such exposure coding is easily achieved in modern sensors and is already a feature of several machine vision cameras.

### 2.3 The Flutter Shutter Video Camera

Flutter shutter (FS) [14] was originally designed as a way to perform image deblurring when an object moves with constant velocity within the exposure duration of a frame. Since FS was essentially a single frame architecture there was very little
motion information that could be extracted from the captured frame. Therefore, linear motion [14] or some other restrictive parametric motion model [23] needs to be assumed in order to deblur the image. In contrast, we extend the FS camera into a video camera by acquiring a series of flutter shuttered images with changing exposure code in successive frames. The key insight is that, this captured coded exposure video satisfies two important properties,

1. Since each frame is a coded exposure image, image deblurring can be performed without loss of spatial resolution if motion information is available.

2. Multiple coded exposure frames enable motion information to be extracted locally. This allows us to handle complex and non-uniform motion.

Thus, for FSVC to work reliably, it is pertinent that both properties are satisfied and that several successive captured frames are available during the decoding process. This stands in contrast with other methods such as [11] and [9] where motion information can be encoded within a single frame by independently changing the exposure time for different pixels.

2.3.1 Notation and problem statement

Let $x$ be a high speed video of size $M \times N \times T$ and let $x_t$ be the frame captured at time $t$. A conventional high speed camera can capture $x$ directly, whereas a low speed video camera cannot capture all of the desired frames in $x$. Therefore, low speed cameras either resort to a short exposure video (in which the resultant frames are sharp, but noisy) or to a full-frame exposure (which results in blurred images). In either case, the resulting video is of size $N \times N \times (T/c)$ where $c$ is the temporal sub-sampling factor. In the FSVC, we open and close the shutter using a binary
Figure 2.2: **FSVC Architecture:** Every captured frame is a sum of a pseudo-random sampling of sub-frames.
Figure 2.3: **Total Variation Prior**: The first row shows example frames from four different videos of increasing complexity in motion. The second and third rows show the XT and the YT slices for these videos. It is clear from the XT and the YT slices that there are very few high gradients and therefore minimizing total variation on the XT and YT slices is an appropriate prior for videos. Further, our mixing matrix essentially punches holes in the temporal dimension, i.e., some rows of the XT-YT slices are completely missing in the observations (corresponding to shutter being closed). Therefore, it is important to use a long sequence of XT and YT slice in order to perform the reconstruction. Also notice that edges in the XT and YT slices encode velocity information. For small compression factors and slow moving scenes, local regions of the video can be approximated using linear motion.

pseudo-random sequence within the exposure duration of each frame. In all three cases, the observed video frames $y_{t_l}$ are related to the high speed sub-frames $x_t$ as

$$y_{t_l} = \sum_{t=(t_l-1)c+1}^{t_l c} S(t)x_t + n_{t_l}, \quad \text{(2.1)}$$

where $S(t) \in \{0,1\}$ is the binary global shutter function, $x_t$ is the sub-frame of $x$ at time $t$, and $n_{t_l}$ is observation noise modeled as additive white Gaussian noise. For a full exposure camera $S(t) = 1$, $\forall t$, while for short exposure video $S(t)$ is 1 only for one time instant within each captured frame. Our goal is to modify the global shutter function and recover all sub-frames of $x_t$ that are integrated during exposure. Since the observed pixel intensities $y$ are a linear combination of the desired voxels $x$ with weights given by $S$ corrupted by noise, equation (2.1) can be written in matrix
form as

\[ y = Sx + n, \]  \hspace{1cm} (2.2)

where \( S \) is a matrix representing the modulation by the global shutter and the observation noise \( n \) is the same size as \( y \). While the modulation of the shutter affects all pixels, the pattern of modulation need not be the same for each integration time.

Equations 2.1 and 2.2 hold for all \( m \times m \times \tau \) patches of a video, so the same notation will be used for patches and the full video. Unless otherwise mentioned, all equations refer to a patch of the video. Let \( x \) and \( y \) represent the vectorized form of the desired high-speed voxels \( x \) (e.g. \( 8 \times 8 \times 24 \)) and the observed voxels \( y \) (e.g. \( 8 \times 8 \times 3 \)) respectively. The observed video \( y \) has significantly fewer entries than the desired true video \( x \) resulting in an under-determined linear system.

### 2.4 Reconstruction Algorithms

Frames captured using the FSVC are a linear combination of sub-frames with the desired temporal resolution. Given that the number of equations (observed intensities) recorded using the FSVC architecture is significantly smaller than the desired video resolution, direct inversion of the linear system is severely underconstrained. Inspired by advances in compressive sensing, we advocate the use of video priors to enable stable reconstructions.

#### 2.4.1 Video Priors

Solving the under-determined system in equation (2.2) requires additional assumptions. These assumptions have typically taken the form of video priors. There are essentially two distinct forms of scene priors that have been used in the literature so far.
Figure 2.4: **Results of Flutter Shutter Video Camera** on a video of a toy bike translating with uniform velocity using TV reconstruction. The top row shows one frame of the reconstructed video for various compression factors. As the compression factor increases, the output degrades gracefully. The bottom two rows show the rotated XT and the YT slices corresponding to the column and row marked yellow and green in the first row. The XT and YT slices clearly show the quality of the temporal upsampling.

Data-independent Video Priors: One of the most common video priors used for solving ill-posed inverse problems in imaging is that the underlying signal is sparse in some transform basis such as the wavelet basis. This has been shown to produce effective results for several problems such as denoising and super-resolution [24]. In the case of video, apart from wavelet-sparsity one can also exploit the fact that consecutive frames of the video are related by scene or camera motion. In [8], it is assumed that (a) the video is sparse in the wavelet domain, and (b) optical flow computed via brightness constancy is satisfied between consecutive frames of the video. These two sets of constraints provide additional constraints required to regularize the problem. Another signal prior that is data-independent and is widely used in image processing is the total variation regularization. A key advantage with total variation-based methods is that they result in fast and efficient algorithms for video reconstruction. Therefore, we use total variation as one of the algorithms for reconstruction in this chapter.
Data-dependent video priors: In many instances, the results obtained using data-independent scene priors can be further improved by learning data dependent over-complete dictionaries [24]. In [9], the authors assume that patches of the reconstructed video are a sparse linear combination of elements of an overcomplete dictionary; this serves as a regularizing prior. We use a data-dependent over-complete basis as a regularizer and show performance superior to total variation-based methods especially when the compression factor is small ($\leq 8$). The problem with using data-dependent regularization for very large compression factors is that the learned patch dictionary has to be much larger than that used in [9] since, as discussed earlier, the mixing matrix for FSVC is more ill-conditioned than the mixing matrix in [9] and [8]. Handling such large dictionary sizes is computationally infeasible and therefore, we use the total variation-based prior for handling larger compression factors.

2.4.2 Total Variation (TV) of XT and YT Slices

Shown in Figure 2.3 are four different videos with increasing complexity of motion. The second and third row of the figure shows the XT and the YT slices corresponding to the four videos in the first row. In spite of the complexity of the scene and the motion involved in these videos, it is clear that the XT and the YT slices are indeed nothing but deformed versions of images—the deformation being a function of 3D structure and non-uniform velocity of the scene. It is also apparent, that just like images, the XT and YT slices of videos are predominantly flat with very few gradients. Motivated by the sparse gradient distribution in natural images, minimal total variation has been used very successfully as a prior for images [25,26] for various problems like denoising and deblurring. Similarly, we use minimal total variation in the XT and YT slices as a prior for reconstructing the XT and YT slices from the
observations. Needless to say, 3D total variation will probably work even better, but we stick to 2D total variation on XT and YT slices, since this results in a much faster reconstruction algorithm. We use Tval3 [27] to solve the ensuing optimization problem on both the XT and YT slices; the high-speed video is recovered by averaging the solutions of the two optimization problems.

Total variation generally favors sparse gradients. When the video contains smooth motion, the spatio-temporal gradients in the video are sparse, enabling TV reconstruction to successfully recover the desired high-speed video. Recovering the high speed video using spatio-temporal slices of the video cube can thus be executed quickly and efficiently. A $256 \times 256 \times 72$ video channel with a compression factor of 4x can be reconstructed in less than a minute using MATLAB and running on a 3.4GHz quad-core desktop computer. Further, the algorithm is fast and efficient and degrades smoothly as the compression rate increases as shown in Figure 2.4.

### 2.4.3 Data driven dictionary-based reconstruction

While total variation-based video recovery results in a fast and efficient algorithm, promising results from Hitomi et al. [9] indicate that significant improvement in reconstruction quality may be obtained by using data driven dictionaries as priors in the reconstruction process. Since the mixing matrix produced by FSVC is far more ill-conditioned than that in [9], we need to learn patches that are larger in both spatial and temporal extent. Motion information is recorded by consecutive observations; we use four recorded frames to reconstruct the high speed video. When the compression rate is $c$, we learn video patches that are $18 \times 18 \times 4c$ pixels. As the compression rate increases, we need to learn patches that are larger both in spatial and temporal extents, so that the spatio-temporal redundancy can be exploited. Unfortunately,
Figure 2.5: DD Algorithm. A: A simulated video of a moving Lena image captured by the FSVC with 6x compression; a captured frame is shown on the right. B: Local velocities are determined using equation 2.3. Overall, the error in the measurement space is smooth and achieves its minimum value at the closest velocity in the database to the actual motion. The error in one frame for three highlighted velocities is shown on the right. Error quickly rises for velocities not near $v^*$ yielding errors that are an order of magnitude larger. C: Four frames of the recovered high speed video are shown.
learning dictionaries is computationally infeasible as the dimension increases and so we limit the use of data-driven priors for compression factors less than 8. Using data-driven (DD) priors for such low compression factors resulted in a significant performance improvement over total variation minimization. In the future, as algorithms for dictionary learning become more robust and computationally efficient, we expect that such data-driven priors will indeed perform better than total variation.

In implementing data-driven dictionaries, we make two small adaptations to the traditional dictionary learning and sparse approximation algorithms. First, we extract the principal components for each velocity \( v \), independently. We achieve this by taking images and translating them by the appropriate velocity \( v \) to create artificial videos which contain scene elements that moving at the desired velocity. Then we extract patches of size \( 18 \times 18 \times 28 \) from these videos. For each velocity \( v \), we learn the top 324 principal components, and create a principal component matrix, \( P_v \). In practice we generated a total of 521 velocities, sampling heading direction uniformly at 9 degrees and varying the speed from 0.15 pixels/frame to 1.95 pixels/frame (resulting in blur of up to \( 1.95 \times c \) pixels/captured frame). The static motion case is also included. Thus, there are a total of 521 principal component matrices \( P_v \).

For example, for a given compression rate of \( c = 7 \), we take \( 18 \times 18 \times 4 \) patches from the captured FSVC video. For each such patch we recover a high temporal resolution patch which is \( 18 \times 18 \times 28 \) resulting in a temporal upsampling of \( c = 7 \). For each \( 18 \times 18 \times 4 \) patch from the FSVC video, we estimate the best local velocity \( v^* \) as

\[
    v^* = \arg\min_v ||y - SP_v \alpha_v||_2^2, \quad v = 1, \ldots, 521.
\]

In equation (2.3), \( y \) is the vectorized observation patch, \( S \) is the observation matrix defined by the flutter shutter code, \( P_v \) is the principal component basis for the \( v \)th
Figure 2.6: **Reconstruction quality vs compression.** Left: As the compression factor increases, the quality of reconstruction decreases. The DD reconstruction curve is limited to 7x. Both algorithms are tested using the same input video of a hairnets advertisement translating to the right; the TV reconstruction uses all 72 frames while the DD reconstruction only uses the first 24 frames (28 for 7x). Right: Frames from the reconstructed videos using both DD (top) and TV based (bottom) algorithms.

velocity, and $\alpha_v$ is the least squares estimates of the coefficients denoted by $\alpha_v = (SP_v)^\dagger y$, where $\dagger$ represents the pseudo-inverse. Figure 2.5b shows that the error is much smaller for velocities near the true patch velocity and quickly rises for other velocities. Finally, each patch is reconstructed as

$$ \hat{x} = P_{v^*} \alpha_{v^*}. $$

The recovered high speed video in Figure 2.5c demonstrates the high quality reconstruction that can be achieved using the DD algorithm. After reconstructing each patch in the observation, overlapping pixels are averaged to generate the reconstructed high speed video. Recovering a high speed video with this algorithm is considerably slower than the TV-based reconstruction, a $252 \times 252 \times 28$ video channel with a compression factor of 7x takes approximately 5 minutes using the same computation
The more pressing issue, is that such a data-driven method suffers from two disadvantages especially when handling large compression factors: (1) learning is prohibitively slow, and (2) the locally linear motion assumption is violated when the temporal extent of patches becomes longer. In practice, we notice that this algorithm results in significant improvement over total variation for small compression factors.

2.5 Experiments

We evaluate the performance of FSVC through simulations on videos with frame rates of 120 – 1000 fps. We also capture data using a Point Grey Dragonfly2 video camera and reconstruct using both algorithms. Effect of compression rate: We
Figure 2.8: **Experimental Results.** Observation images with a spatial resolution of 234 × 306 were captured at 7 frames per second with a compression factor of 7x. **Top:** Four frames of a book being pulled to the left are captured by FSVC, ghosting artifacts can be observed in the outset. The high speed video is reconstructed using the DD algorithm and one frame is shown. The outset shows that the ghosting has been removed. **Bottom:** A toy robot is moved to the right along a non-linear path, 12 observations are captured using FSVC. Ghosting of the fine details on the robot can be seen in the outset. Reconstruction was done with the TV algorithm and one frame of the output is shown. The outset shows that fine details have been preserved.

First test the robustness of our algorithms through simulation. Simulated observations are obtained by sampling a high speed ground truth video using the forward process of equation (2.2). To test the robustness of our algorithm at varying compression factors, we use a video of hairnets advertisement moving from right to left with a speed of ∼ 0.5 pixels per frame (video credit: Amit Agrawal). The input video has values in the range [0, 1], the observation noise has a fixed standard deviation, \( \sigma = 0.025 \), and the exposure code is held constant. The observed videos have a PSNR of ∼ 34 dB. The effect of increasing the compression on the quality of the recovered
video is highlighted in Figure 2.4. As the temporal super-resolution increases, FSVC retains fewer dynamic weak edges but faithfully recovers large gradients leading to a graceful degradation of video quality. Notice, in Figure 2.4, that the hub caps of the wheels of the bike are present in the reconstructed video even as the compression is increased to a factor of 18. FSVC has a high spatial resolution that allows slowly moving weak edges to be preserved in the video reconstruction. Shown in the plot in Figure 2.6, is the reconstruction PSNR of both algorithms as a function of the compression rate. Notice that for low compression factors, the data-driven algorithm performs better than total variation, since the locally linear motion assumption is not violated. As the compression factor increases, the peak signal-to-noise ratio (PSNR) of the reconstructed video decays gracefully for the TV-based algorithm. The ability to retain very high quality reconstructions for static/near-static elements of the scene and the graceful degradation of the TV algorithm, both of which are apparent from the reconstructions of the hairnet advertisement video in Figure 2.6, are very important attributes of any compressive video camera.

Robustness to observation noise: Figure 2.7 shows the effects of varying levels of noise on the fidelity of reconstruction for both the DD and TV algorithms. The same hairnets video is used as the ground truth and the standard deviation of the noise varies from 0.025 to 0.125 in increments of 0.025. This corresponds to a range of PSNRs from 20.8 – 34.8 dB. The compression factor has been fixed at 7x. In this experiment, we compare the quality of the high speed video recovered from images captured using FSVC and a standard video camera using a short exposure. The short exposure video is created by modifying the temporal code used in $S$ to have a single 1 for the same time instant $t$ during each exposure. The coded frames and short exposure frames are reconstructed using the DD and TV algorithms. As expected for
a low-speed and approximately linear scene, the DD algorithm reconstruction is of higher quality than the TV reconstruction. This experiment shows two observations: (1) degradation of the FSVC reconstruction is graceful in the presence of noise and (2) FSVC coding improves the fidelity of reconstruction by approximately 5 dB compared to simple short exposure (at a compression rate of $c = 7$).

**Experiments on real data:** The FSVC can be implemented using the built-in functionality of the Point Grey Dragonfly2 video camera. The coded exposure pattern is provided using an external trigger. Due to inherent limitations imposed by the hardware, images were captured at 7 frames per second with a compression factor of 7x. Hence, our goal is to obtain a working 49 frames per second camera by capturing 7 frames per second coded videos. Figure 2.8 shows the input coded images collected by the camera, and frames of the reconstructed video using the TV and DD algorithms. The top row of Figure 2.8 shows one observation frame recorded with FSVC of a book moving to the left. Ghosting artifacts in the observation frame demonstrate that using a traditional video camera with the same frame rate would result in a blurry video. We recovered a high-speed video using the DD methods; as expected, the recovered video correctly interpolated the motion and removes the ghosting artifacts.

A second dataset was captured of a toy robot moving with non-uniform speed. One captured frame from this dataset is shown in Figure 2.8 (bottom-left). The outset shows an enlarged view of a blurry portion of the observed frame. Figure 2.8 (bottom-right) shows one frame from the reconstructed video and the outset shows the corresponding enlarged section of the recovered frame. Notice that the blurring is

---

*Exposure mode 5 is used to control the timing and duration of multiple exposures within each integration time of the camera.*
Figure 2.9: Comparison of the qualitative properties of compressive computational video cameras.

significantly reduced but since the motion of the toy is quite large, there is still some residual blurring in the reconstructed video.

2.6 Discussions and Conclusions

Benefits: Our imaging architecture provides three advantages over conventional imaging. It significantly reduces the bandwidth requirement at the sensor. It exploits exposure coding which is already a feature of several machine vision cameras making it easy to implement. It improves light throughput of the system compared to acquiring a short exposure low frame-rate video or [9] and allows acquisition at low light levels. These are significant advantages since the prohibitive cost of high-speed imagers is due to the requirement for high bandwidth and high light sensitivity. Figure 2.9 highlights these advantages of our imaging architecture as well as places it in a larger context of computational cameras developed over the last decade.
Limitations: FSVC exploits the spatio-temporal redundancy in videos during the reconstruction phase of the algorithm. Scenes that do not have spatio-temporal redundancy such as a bursting balloons cannot be handled by the camera. Since the spatio-temporal redundancy exploited by traditional compression algorithms and our imaging architecture are very similar, as a rule of thumb one can assume that scenes that are compressed efficiently can be captured well using our method. FSVC uses a coded exposure and this causes a 33% reduction in light throughput since we use codes that are open 67% of the time.
Chapter 3

Generalized Assorted Cameras

Generalized Assorted Pixels (GAP) has been proposed for capturing high spatial resolution images with spectral detail, high dynamic range and multiple polarization states. GAP works by placing a diverse filter mosaic directly on the photodetectors. However, beyond Bayer mosaics, the GAP architecture has not become widely available, since depositing filter mosaics on the sensor array can be both cumbersome and expensive. Furthermore, the filter configuration is fixed, which reduces the sensor’s flexibility.

In this chapter, we propose to use Generalized Assorted Cameras (GAC), an imaging architecture that leverages the recent surge in camera arrays to realize the potential of GAP. In the GAC architecture, a mosaic of filters is placed in front of an array of cameras. Since the different sensors in a GAC array do not share a single viewpoint, correspondence across cameras needs to be established. Luckily, the recent success in multi-view stereo and structure from motion has shown that sub-pixel dense correspondences can be reliably obtained. We leverage these existing state-of-the-art techniques to establish dense correspondence across a GAC array and warp the data obtained from the different cameras to the viewpoint of a reference camera. The GAC camera then essentially acts like a GAP camera. The primary advantage of the GAC architecture is that the external filter mosaic can easily be customized for particular applications which require spectral selectivity, high dynamic range etc. We show several experimental results in multi-spectral imaging, high dynamic range imaging
3.1 Introduction

In computational photography, we frequently face the issue of resolution trade-off. If the full light-field of an object is obtained [28], samples of the plenoptic function [29] can be generated by image-based rendering. However, capturing all the scene’s light-rays is an inefficient strategy for many real-world scenarios. Therefore, most subsequent work has focused on tactics for either exploiting scene redundancies [30] or trading off spatial, angular, temporal and spectral resolutions [31–35] to achieve a particular goal.

In this chapter, we propose to use a Generalized Assorted Camera (GAC) array that provides a flexible way to tackle resolution trade-offs. Each GAC array sensor might have different characteristics, such as spectral response, sensor gain, video and polarization-based imaging.
frame-rate, neutral-density filters, polarization filters etc. Given scene point correspondences and assuming diffuse reflectance and no motion-blur, we can easily render an image of the scene with the combined characteristics of all the cameras. A specific instance of this idea was proposed by [36] who increase temporal resolution using a camera array. We generalize and extend that concept to any other dimension of the plenoptic function. A camera array gives a higher SNR than co-locating sensors with a beam-splitter. Furthermore, changing the camera filters is convenient and adding or removing a few cameras from the setup would involve only a small amount of post-calibration computation. This flexibility makes GAC arrays efficient since the sensor’s limited resources can be directed to those parts of the light-field that provide success at some particular vision, imaging, or graphics task.

The GAC array does not reduce spatial or temporal sampling and, depending on the filters used, increases the resolution of one reference camera view along multiple dimensions of scene information. However, the GAC array does not increase the angular resolution. This is in contrast to previous light-field camera array research that captures many angular samples to achieve goals such as refocusing. In fact, in our method, we want the cameras to be as close as possible to each other, so that the angle subtended at each scene point is small.

What allows us to go beyond previous camera array efforts is the recent surge in methods that provide high-quality 3D reconstruction from multiple views. These techniques typically perform triangulation using robust features such as SIFT [37], followed by bundle adjustment [38] and a global or semi-global [39] regularization. This produces dense pixel correspondence between all the views in the camera array. Note that multi-view reconstruction works best when all the camera characteristics are almost identical. Yet, our method requires that the characteristics of our sensors
are diverse and assorted. Therefore, some near-identical subset of the cameras is used for scene reconstruction.

We demonstrate a variety of applications of a GAC array. First we present an application that provides multi-spectral video at 15 fps. Most commercially available multi-spectral cameras use a tunable filter. Since such cameras temporally multiplex the spectral information from the scene, they suffer from motion blur. Other techniques have been proposed that compromise on the spatial resolution to obtain low-resolution multi-spectral videos [40] or require controlled illumination [41]. The main advantage of using an assorted camera array for multi-spectral data is that we are able to get video data for any set of task-specific spectral filters, which can easily be placed on the array, without reducing spatial resolution. Next we demonstrate high dynamic range imaging that is obtained by controlling exposure duration for each of the cameras. Finally, we use polarization filters to separate objects containing polarized light from the rest of the scene.

3.1.1 Prior Work

Plenoptic Imaging: Common plenoptic imaging systems use a microlens array to capture angular resolution [31, 42–46]. Such systems can be used for refocusing [31], high-dyanmic range [43], and spatial super resolution [42]. The major drawback of using a single sensor light field camera is the loss in spatial resolution. For example, the Raytrix camera captures the light field on a 10.8 megapixel sensor but has an effective resolution of at most 3 megapixels [47].

Multi-spectral Imaging: Filter-based methods [48, 49] take a sequential series of images using various narrowband filters and only work for static scenes. Multi-
spectral video is realized by adding elements along the optical path such as a dynamic mirror [50], Lyot filter [51], filter array [52], or a dispersing prism [53]. Since many of these elements are internal to the camera, the spectral sampling is fixed at manufacture. Closest to our work is [54] and [55], where narrow-band filters are also placed on cameras. In both cases the scene is assumed to be planar and the images are aligned using homographies. Alternatively, an array of cameras, each with a different narrow-band filter in front of the lens, can be used to simultaneously capture high spatial resolution images [54]. Since the images captured by each camera are no longer registered an additional processing step is required to recreate the spectral cube. This camera design is similar to our system but has the restriction of acquiring approximately planar images an applying an affine image warping to perform registration and does not capture video data.

**High Dynamic Range imaging:** Photographs of scenes with that exhibit a high dynamic range (HDR), greater than the available 256 levels, either lose details in bright regions that are washed out or in dark regions that are underexposed. If the scene and camera are static, multiple images can be recorded with varying exposure times and combined post capture [56, 57]. In [58] wide angle HDR images of planar scenes are captured by attaching a spatially varying density filter to the front of the camera lens and capturing multiple images as the camera moves, generating multiple measurements of each scene point under varying optical conditions. HDR video of dynamic scenes requires a trade-off in spatial [52, 59] or temporal resolution [60]. The work most similar to ours is the camera array constructed in [61] where multiple cameras simultaneously record images with varying exposure times; however their implementation is limited to planar scenes and the differing exposure times intro-
duces non-uniform blurring between captured images. Kanbara et al. [62] also use a ProFUSION camera array to capture HDR scenes but operate under the assumption that all scene points are greater than 5 meters away from the camera array, thus having zero disparity.

**Generalized sampling:** The generalized assorted pixel camera array [52] replaces the common Bayer color filter array with a mosaic that offers greater spectral, HDR, or polarization [63] resolution. The inclusion of additional filter types reduces the spatial resolution of the camera and the filter selection is fixed. In generalized mosaicing for static scenes [58] the camera moves and captures a sequence of images through an external spatially varying filter. For dynamic scenes, most previous work requires a trade-off in spatial [52,59] or temporal resolution [60]. [61] uses an array of cameras to simultaneously record images with varying exposure times; however, their implementation is limited to planar scenes. Our camera array expands upon the principles of these generalized sampling cameras by exploiting high-quality reconstruction algorithms that are now widely available.

**Image alignment:** Multi-camera and multi-image techniques for increasing spectral and HDR resolution require precise alignment of the captured images. Traditional methods for aligning images via stereo matching by rectifying image pairs [64] and running stereo correspondence are quite mature [65]. Most stereo algorithms either aggregate costs of local windows—sum of squared differences [66], normalized cross correlation [67] and rank and census transforms [68]—or use explicit smoothness assumptions and operate globally—simulated annealing [69,70], probabilistic diffusion [71], and graph cuts [72,73].

Robust multi-view stereo algorithms can be used to reconstruct a 3D model of
the scene provided that each camera is operating in the same mode [74]. Common algorithms include a plane-sweep generalization [75, 76] and multiple-baseline stereo [77]. However, in practice the typical approach to generating dense correspondence in camera arrays has often been to assume that the scene is sufficiently far away as to be planar and the disparity estimation reduces to fitting homographies [54,55,61].

Our GAC framework uses a combination of multi-view plane-sweep and graph cuts minimization algorithms to generate dense disparity maps between the cameras in the array.

3.2 From Assorted Pixels to Assorted Cameras

Assorted pixels [33,52,63] leverage the large pixel counts that appear with successive generations of digital cameras. These methods use a generalized filter mosaic directly on the photodetectors to obtain, after demosaicing, images containing polarization, high-dynamic range and other characteristics. These methods preserve spatial resolution by high-quality interpolation.

The main problem with a camera equipped with assorted pixels is that the pixel characteristics, and their distribution, are fixed, physically, inside the sensor. We cannot easily add or remove filters for scene properties that we would like to measure. Furthermore, the advantage of retaining spatial resolution is eroded as the number of pixel characteristics are increased. Suppose a camera sensor has $N$ pixels. If we are interested in measuring $m$ characteristics of the scene, each requiring $K_i$ channels where $i \in (1, m)$, then the number of pixels that measure a particular scene property are reduced to $\frac{N}{K_1 K_2 \ldots K_m}$. For example, a $N = 200 \times 300 = 60000$ assorted pixel camera that has three color channels $K_1 = 3$ and five HDR channels $K_2 = 5$ would have only $\frac{60000}{3 \times 5} = 4000$ pixels for each color in a particular intensity range. Since
Figure 3.2: View interpolation in video: Calibration prior to video capture: 1) compute the internal parameters (focal length and principal points) of all the cameras separately, 2) compute the external parameters (rotation and translation w.r.t. the center camera). For each frame: 3) obtain a dense 3-D point cloud using groups of camera, and 4) warp all the images to the center camera based on the 3-D point cloud.

interpolating a $40 \times 100$ image to obtain the full $200 \times 300$ resolution will result in artifacts, the assorted pixel approach may not work.

Multi-dimensional imaging with assorted cameras: Figure 3.1 depicts the GAC approach, where each camera has a different property, such as spectral sensitivity, frame-rate, polarization etc. The conventional notation of a light-field as a set of 4D rays with intensities can be denoted as $L(x, \omega)$, where $x$ is the spatial location on an arbitrary plane $\Pi$ and $\omega$ are the azimuth and elevation angles of the light ray being described. Since we wish to measure additional dimensions of information along the light-ray, we augment the light-field notation as $L(x, \omega, \Theta)$, where $\Theta$ describes scene properties being measured. This notation is inspired by the plenoptic function [29], and each camera can sample a different subset of $\Theta$. For example, a camera measuring a certain wavelength $\lambda$ would be obtaining a slice of the light-field $L(x, \omega, \lambda)$.

Our method assumes that the measurements of a scene point’s radiance from cameras with slight different viewpoints are identical. This means that, if any angle
subtended by the camera array at any point $P$ is in the range $\omega_i \in (\omega_1, \omega_n)$, and these rays intersect the plane $\Pi$ at points $x_i \in (x_1, x_n)$, then for any subset $\theta_i \in \Theta$, $L(x_i, \omega_i, \theta_i) = L_{\theta_i}$, which is a constant. As long as the cameras are close together relative to the depths of the different objects in the scene, this requirement is usually true for a broad range of scene reflectances. If we obtain dense pixel correspondence between all the cameras, then we can warp the views from the different sensors onto a canonical “central” camera, transforming it into a multi-dimensional image sensor. In our work, the central camera is usually the sensor whose location is closest to the median camera location. The set of warped images forms a cube of data, much like a hyper-spectral cube, and each pixel contains information corresponding to $\Theta$.

### 3.3 Establishing Point Correspondences

To warp all images to a central camera with sub-pixel accuracy we perform the procedure outlined in Figure 3.2, namely: 1) for all cameras calibrate the intrinsic parameters, such as focal length and principle points, 2) compute the extrinsic parameters, rotation and translation, of all the cameras with respect to the central camera, 3) perform dense 3D reconstruction using as subset of the cameras, 4) using the 3D point reconstruction warp all of the images to the viewpoint of the central camera.

Given that the position of each camera is fixed with respect to the others, accurate calibration of the array need only be performed once. Once calibrated, the projective geometry of the cameras restricts where a pixel from one camera will project to on the remaining cameras, a fact that we will use to guide the search for correspondence.
3.3.1 Internal and External Calibration

Camera calibration is performed by following the standard procedure of extracting initial estimates of internal and external camera parameters and refining those estimates using bundle adjustment. We compute the internal parameters of the camera, i.e., focal length, principle points and skew, using the Caltech calibration toolbox [78], and by imaging a planar checkerboard pattern at 24 different orientations using all of the cameras in our array. At the conclusion of internal calibration we obtain an average reprojection error of 0.6 pixels.

Next we compute the extrinsic parameters (rotation and translation) of the cameras with respect to the central camera. Following internal calibration, [78] also provides an initial estimate for the rotation and translation of each camera relative to the central camera. We obtain an average reprojection error of 1.13 pixels following initial external calibration as seen in Figure 3.3. To improve the estimates of the extrinsic parameters, we jointly refine internal and external camera parameters through bundle adjustment.

If $P_i, i = 1, 2, \ldots, M$ are the projection matrices of the $M$ cameras in the array and $X_j, j = 1, 2, \ldots, N$ are the total number of 3D calibration scene points and $x_{i,j}$ is the imaged $j$-th point in the $i$-th camera, then the cost minimized by bundle adjustment is:

$$
\min_{P_i, X_j} \left[ \left( x_{i,j}(1) - \frac{P_i(1,:)}{P_i(3,:)X_j} \right)^2 + \left( x_{i,j}(2) - \frac{P_i(2,:)}{P_i(3,:)X_j} \right)^2 \right],
$$

(3.1)

where $\tilde{X}_j = [X_j 1]$ is the homogeneous representation of the $X_j$. We use the sparse bundle adjustment toolbox [79], to refine the estimates and obtain a mean reprojection error of 0.24 pixels as shown by the second histogram in Figure 3.3.
Figure 3.3: **Calibration quality:** (Left) We first compute an initial estimate of each cameras’ external parameters using the Caltech calibration toolbox’s [78] stereo functionality. (Right) We then refine these estimates by performing bundle adjustment (BA) using the Sparse bundle adjustment toolbox [79]. Shown above are histograms of the reprojection error for each stage. The mean reprojection error is 1.13 pixels before BA and 24 pixels after BA.

### 3.3.2 Dense 3D Reconstruction

Given an array of calibrated cameras, we choose the center camera to be the reference camera. All depth calculations are performed on the grayscale intensity of the images.

We compute a dense 3D depth reconstruction, with respect to the reference camera, by sweeping a virtual plane through the 3D volume of the scene and computing the cost of each scene point being placed at the depth of the plane. For a depth $d = 1, \ldots, D$, each pixel in the reference camera is projected to a point $X_q$ on the virtual plane. Each point is then projected to the remaining cameras where the images are sampled, thus each scene point $p$ is represented by an $q$-dimensional vector where $q$ is the number of cameras in the array,

$$p = I_n(P_nX), \ n = 1, \ldots, q, \ \forall \ scene \ points$$  \hspace{1cm} (3.2)

where $P_n$ is the projection matrix that projects points in the world coordinates to the image coordinates of the $n$th camera. Should the projection $P_nX$ fall between pixel
coordinates, $I_n(P_nX)$ is computed using bicubic interpolation.

For an $M \times N$ pixel reference view, the collection of scene points form an $M \times N \times q$ cube for each hypothesized depth. Instead of working with individual pixels, overlapping $8 \times 8$ patches are used to compute the depth of scene points.

The depth map is found by using graph cuts to minimize the energy

$$E(d) = E_D(d) + \lambda E_S(d),$$

(3.3)

where $E_D(d)$ is the cost of placing a scene point $p$ at depth $d$ and $E_S(d)$ is a bilateral smoothing term. In other words, the data term, $E_D(d)$, measures the degree of fit of the depth assignment $d$ in a local region while the smoothness term, $E_S(d)$, measures the interaction between neighboring patches. The total energy is minimized using $\alpha$-expansion graph cuts method (and supplied code) described in [72,73,80].

**Data term:** Designing an appropriate data term for stereo matching is a well researched topic. A simple yet effective data term is to compute the sum of squared differences (SSD) for pixel intensities [66]. For the data term in 3.3 we extend the SSD metric using windows of size $8 \times 8$ to the multiview search. Let $y$ be an $8 \times 8$ patch, let $\mathcal{P}$ be the set of all patches in the reference image $R$, let $R_y$ be the pixel values of patch $y$ in the reference image reshaped as a column vector and let $I^y_n$ be the pixel values of patch $y$ in the $n$th image reshaped as a column vector. The data term is then

$$E_D(d) = \sum_n \sum_{y \in \mathcal{P}} (R_y - I^y_n)^2.$$  

(3.4)

The cost of placing each patch at every depth $d$ is recorded and $E_D(d)$ is the collection of costs associated with all of the patches being placed at depth $d$. The SSD metric was chosen because it strongly penalizes edge mismatch where it can be assumed that the pixel intensity on either side of the edge is large.
Smoothness term: The data term just is computed independently for each patch allowing neighbors in smooth regions to have vastly different depth assignments. The smoothness term \( E_S(d) \) is designed to remedy this shortcoming. Specifically, the smoothness term ensures that the depths assigned to neighboring patches are smooth if the observed patches are similar and simultaneously allows discontinuity at object boundaries. The smoothness term is of the form

\[
E_S(d) = \sum_{j \in \mathcal{N}_i} u_{\{i,j\}}(d),
\]

where \( \mathcal{N}_i \) is the set of all patches neighboring patch \( i \in \mathcal{P} \) and \( u_{\{i,j\}}(d) \) is a function that assigns penalties according to the attributes of the patches and depth assignments. In practice, we use the median intensity value \((\tilde{i}, \tilde{j})\) of patches taken from the reference image as a measure of similarity. Patches in close proximity that have similar median intensities are assumed to belong to the same object and should therefore have similar depth assignments. We achieve this by implementing \( u_{\{i,j\}}(d) \) with a sigmoid function to enforce smoothness for proximate patch intensities and to allow disjoint assignments for patches with disparate median values.

\[
\text{sg}(\tilde{i}, \tilde{j}, a, b) = \frac{1}{1 + e^{||\tilde{i} - \tilde{j}|| - a/b}},
\]

and \( a \) and \( b \) are the center and width of the sigmoid respectively.

The spatial weighting of the neighbors shown in equation 3.6 is coupled with a simple \( L_1 \) distance measure on the depths to favor smooth depth variation. The function \( u_{\{i,j\}}(d) \) is given by

\[
u_{\{i,j\}}(d) = \text{sg}(\tilde{i}, \tilde{j}, a, b)|d_i - d_j|
\]

where \( d_i \) and \( d_j \) are the depths assigned to patch \( i \) and \( j \) respectively. If the difference between median values is large, the spatial weighting sigmoid forces the smoothness term to zero allowing for discontinuity in depth assignment.
Figure 3.4: 3D **reconstruction**: A subset of cameras are used for computing the dense 3D point cloud of the scene. We compute the scene depth map with respect to a reference camera. A dense 3D point cloud is found by triangulating the 2D point correspondences.
Figure 3.5: **Warping to the reference camera:** We project the reconstructed 3D points to all the cameras. Using this relationship, we map pixels to the correct location in the reference camera view. (Left). In this figure, we have shown the mean of all warped images (Right). The sharp edges indicates a high quality warping.

### 3.3.3 View Interpolation

We perform view interpolation by warping all of the camera views to the reference view by projecting the 3D points $X_p$ to all the cameras with projection matrices $P_n$. Let $x_{n,p}$ be the projection of the $p$th point in the $n$th camera and let $x_{n_r,p}$ be the projection in the reference camera $C_{n_r}$. We map the pixel value at $x_{n,p}$ to the location $x_{n_r,p}$ in the reference camera for all of the 3D points. An overview of 3D correspondence matching is shown in Figure 3.4 and the resulting warped images are shown in Figure 3.5.

### 3.4 Direct Multi-spectral Imaging

Traditional cameras are unable to differentiate between metameric scene points which have the same RGB values but different spectral composition. The GAC framework allows for increased spectral resolution to capture these differences by using narrow-band filters in front of some cameras in the array. In general, the specifications of the
Figure 3.6: **Prototype GAC:** (a) The ProFUSION $5 \times 5$ color camera array records video at 15 fps. (b) The location of narrowband filters used to directly measure multi-spectral video. (c) Broadband filters used to capture multiplexed spectral images. (d) Location of pairs of linear polarization filters.

Filters such as bandwidth and the central wavelengths of the filters are application dependent. For example, suppose that we wish to capture multi-spectral information within the range 410nm-710nm. Using a $5 \times 5$ camera array, it would be possible to use 16 filters with a bandwidth of 10nm while leaving the remaining cameras vacant to compute point correspondences. The center wavelengths of the filters are chosen to span the range of interest and are shown in Fig. 3.6(b).
Figure 3.7: **Spectral measurements:** 9 of the 16 spectral bands recorded using narrowband filters in front of the camera array. Images are displayed from the viewpoint of the central camera. From left the fruits are a green apple, a lemon, an orange, and bananas.

**Hardware:** We implement our GAC using a ProFUSION $5 \times 5$ camera array distributed by Point Grey. Each camera has a resolution of $640 \times 480$ pixels, a Bayer color filter array, and a throughput of 15 frames per second. Additionally, image acquisition is synchronized but each camera offers independent control of gain and exposure duration. In our experiments the gain was held constant across all cameras. An image of the camera array and the filter configuration used to directly sample multiple spectral bands is shown in Figure 3.6 (a) and (b).
**Multi-spectral images:** We first capture multi-spectral images of an assortment of fruit in front of a multi-colored background. Since the ProFUSION sensor contains a Bayer mosaic, the captured images are demosaiced [81] and then converted into a luminance image. The nine spectral bands are shown in Fig. 3.7. Notice that the citrus fruits and bananas are dark in the blue and green wavelengths (below 580nm) and bright in the yellow, orange and red wavelengths while the green apple is brightest at 550nm.

After capturing the spectral images, false RGB images can be constructed for ease of display. Figure 3.8 shows two scenes captured using the GAC and rendered in false RGB. The first scene is of the fruit placed in front of a color checker and the second scene is a collection of household objects. For each scene, we show the average pixel intensity for both the warped images and unwarped images of the RGB cameras and the same images are shown for the false RGB representations of the spectral datacube. Plots of the reflectance of select scene points show the richness of spectral information that is obtained. In the first scene spectral responses of red, green, and blue points on the color checker are plotted while in the second scene, responses for points on the red and green marker caps are displayed. By viewing the spectral response curve of the green marker is it possible to see that it has a fairly broadband response. The filters at the edge of the spectral range have lower transmission than the other filters, therefore spectral measurements recorded in this range are subject to more noise. A ground truth experiment, similar to the experiment shown in Figure 3.10, using a color checker, was also conducted to verify the quality of our measurements. The spectral content of the squares is reconstructed with an average SNR of 23.3dB.
Figure 3.8: **Multi-spectral quality**: Two scenes captured using narrowband spectral filters. For scenes I & II: (a) the center RGB image, (b) the average of the images captured by the remaining RGB cameras without warping, (c) the average of the warped RGB images, (d) the average of unwarped spectral images [shown in false RGB], (e) the average of the warped false RGB spectral images, and (f) spectral measurements of selected scene points shown in (a).
Figure 3.9: Comparison of spectral content in a frame of a multi-spectral video of two hands with different amounts of melanin weaving in an arbitrary path: (a-e) follow from Figure 3.8, (f) shows the spectral reflectance of points on the two hands, (g) mean of the warped false RGB images for two additional frames of the video.

**Multi-spectral video:** One advantage of the GAC concept is that temporal resolution is preserved. We demonstrate multi-spectral video, using the ProFUSION array, at 15 frames per second. Each frame of the video is processed independently. In Figure 3.9, three frames are shown of a video where two people, with different skin tones, wave their hands in front of a color checker background. In areas where point correspondences are inaccurate, minor pixel artifacts may occur but, in general, the frames of the video are temporally coherent.

### 3.5 Multiplexed Multi-spectral Imaging

While simple to use, the narrowband filters discussed in Section 3.4 have low light-throughput and reduced SNR. Furthermore, the Bayer mosaic on the cameras further attenuates the signal. In the worst case, a particular narrowband filter might have mutually exclusive spectral frequencies when compared to one or more of the Bayer
filters and the SNR of the resulting measurement will be so low as to be useless.

This suggests a multiplexing strategy to maximize SNR, similar to that used by [58] and [41]. We use a set of broadband filters, each with high transmittance, and whose transmission power for a particular wavelength $\lambda$ is $F_c(\lambda)$ where $c$ is the camera index (since each camera has only one filter). Post-capture, we demultiplex the measured intensities to produce the spectral reflectance $R(\lambda)$ of a scene point.

If the spectral response of the Bayer pattern is $B_i(\lambda)$, where $i$ is the index of either red, green or blue filters, then the intensity measurement $E$ of this scene point at camera $c$ and for color channel $i$ is:

$$E_{ci} = \int J(\lambda)B_i(\lambda)F_c(\lambda)R(\lambda)d\lambda,$$

where $J$ is the (unknown) spectral response of the illumination. If we apply a demosaicing algorithm (such as [81, 82]) to the raw Bayer image data, then we can obtain all three color channels at every pixel. Therefore, if there are $C$ cameras and $I$ Bayer channels we get $CI$ measurements for a scene point.

If we discretize the above equation into $S$ spectral channels, combine the known filter and Bayer spectral responses into an effective camera filter $G_{c,i}$, and fold the illumination $J$ into an effective reflectance $R$, then,

$$E_{c,i} = \sum_{s \in S} G_{c,i}(s)R(s).$$

Which in vector notation is $E = GR$. Given linear responses for the sensors in the camera array, we propose a calibration step that allows us to recover the unknown spectral reflectance of a scene point. Our method does not require recovery of the $CI \times S$ mixing matrix $G$.

We first capture images of a color chart with known spectral reflectances $R$ and capture a $CI \times CC$ matrix $E$, where $CC$ is the number of color chart squares, such
that $\mathbf{E} = \mathbf{GR}$. Now, given a length $CI$ vector of measurements $\mathbf{X}$ for some scene point, obtained by the imaging process of equation 3.9, we learn a set of $K$ coefficients that reconstruct $\mathbf{X}$ using $\mathbf{E}$ as a dictionary,

$$
\min_K \| \mathbf{X} - \mathbf{E} \mathbf{K} \| \quad \text{where} \quad \| k_j \|_0 < \Delta,
$$

(3.10)

where $\Delta$ is a threshold on sparsity. If the unknown reflectance of $\mathbf{X}$ is $\mathbf{R}_x$, then the minimization of the above equation will also minimize (using equation 3.9 and equation 3.10),

$$
\min_K \| \mathbf{G} (\mathbf{R}_x - \mathbf{R} \mathbf{K}) \| \quad \text{where} \quad \| k_j \|_0 < \Delta.
$$

(3.11)

Therefore, given a calibration color-chart image, taken under the same unknown illumination $\mathbf{J}$ as the scene, and given the measurements $\mathbf{X}$ at a scene point, we find the sparse weights $\mathbf{K}$ to reconstruct the measurements $\mathbf{X}$ from the color chart image data. We use the same weights directly on the known spectral responses of the color chart to recover the spectral response $\mathbf{R}_x$ of the scene point.

We tested this theory by performing an experiment (Fig. 3.10) where we captured an image of the color chart seen in Fig. 3.9 with 140 squares and with 40 spectra levels. We then placed a second color chart of 24 squares and captured images using filters with a broad spectral response. The spectra of each square was recovered with an average SNR of 13.69dB. The figure shows the real and recovered spectra for some of the squares. Notice that the shapes of the spectral curves closely match the ground truth responses.

We observe that the $CI \times S$ matrix $G_{c,i}$ must be full rank if the scene point reflectance is to be recovered. While this requirement is necessary but not sufficient, we have found it an effective guiding principle for a greedy filter selection algorithm. Given a set of candidate filters $\mathbf{H}$ (for example, from a Roscolux booklet), and given
Figure 3.10: Color chart recovery: Since the responses of our sensors are linear, we use a linear system of weights to reconstruct the spectral reflectance at any point from the image of a 140 square color chart (not pictured). We test our method with a second color chart (pictured) which did not belong to our training set. We show the recovery of five squares, demonstrating good reconstruction and with an SNR of 13.69dB.

a partial set of selected filters $G_{\text{partial}}$, we implement a greedy selection algorithm inspired from [83],

$$F_{\text{next}} = \min_{F \in \mathcal{H}} \text{cond}(G_{\text{new}})$$  \hfill (3.12)

where $\text{cond}$ refers to the condition number of the matrix $G_{\text{new}}$ formed by concatenating $G_{\text{partial}}$ and $F$.

### 3.6 HDR & Polarization Imaging

**HDR Imaging**: Traditional HDR methods take a sequence of images of a scene at different exposure settings, which can result in motion blur. GACs are able to record
Figure 3.11: **HDR imaging**: In scenes with complex lighting the dynamic range necessary to correctly expose all regions of the image exceeds the capability of a single camera. Columns 1–3 show the images recorded with varying exposure times warped to the viewpoint of the center camera. For short exposure times the specularity is exposed properly but the rest of the image is underexposed while long exposure images dark regions are properly exposed and the specularity is overexposed. The image on the right is the output of computing a tone mapped HDR image using the views warped to the center camera.

HDR images of dynamic scenes [61] and minimize the presence of motion blur. When using the camera framework shown in Figure 3.6(b), nine cameras do not have filters. High dynamic range images can be captured by varying the exposure duration of these unfiltered cameras. Seven of the remaining nine cameras are used with different exposure times to maximize the captured intensity range. The smallest and largest exposure times ($t_0$ and $t_6$ respectively) are determined by the scene composition. The remaining exposure durations are chosen to span the range between $t_0$ and $t_6$.

Computing point correspondences for HDR images requires an additional processing step. Since all of the cameras have different exposure settings, we first normalize the image intensities before computing stereo disparities. Let cameras $C_0, \ldots, C_6$ have exposure times $t_0, \ldots, t_6$. Point correspondences are found by grouping all sets of three cameras with neighboring exposure durations. For example, cameras $C_1, C_2,$ and $C_3$ form the first group and the intensities of cameras $C_1$ and $C_3$ are normalized to match the intensity range of camera $C_2$. Pixels which are underexposed ($E < 10$) or overexposed ($E > 250$) in any of the three images are not used to compute cor-
Figure 3.12: **Polarization and HDR**: 4 pairs of linear polarizers are used to image an LCD display while the remaining 17 cameras are used to record HDR images. (a) The view of the scene from the central camera. The insets show that the Lego car is overexposed while the rabbit toy is underexposed. (b) The average of the 6 images from polarizers that do not block light from the display warped to the center camera. (c) Average of the 2 polarizers which block the light from the LCD warped to the center camera. (d) Tone mapped HDR image captured using the remaining 17 cameras not used for polarization imaging. As seen in the inset, the Lego car and rabbit are properly exposed.

Point correspondences in saturated regions are computed using these camera pairs. The 3-D point cloud is computed for this grouping. The process is repeated to obtain 3-D point clouds for all sets of neighboring exposures. Finally, view interpolation is done using all of the 3-D points. By design, the brightest scene points are overexposed in all cameras except $C_1$ and the darkest scene points are underexposed in all cameras except $C_6$. In order to compute point correspondences in these regions the exposure duration of the remaining two cameras is set to $t_0$ and $t_6$. Point correspondences in saturated regions are computed using these camera pairs.

Figure 3.11 shows the results of capturing a specular scene with varying illumination. Images of a globe with two specularities on its surface are recorded with exposure times that vary from $1 - 16$ms. Any single image of the globe suffers from limited dynamic range as shown in Figure 3.11(a). The final tone mapped HDR image shown in 3.11(b) effectively extends the dynamic range of the scene.
**Polarized imaging:** The GAC can be used to detect and remove polarized light from a scene. For example, most LCD monitors emit light that is polarized to 45° or 90°. In order to block light from LCD screens, 4 pairs of linear polarizers are placed in front of the camera array. Assuming that the LCD monitors are perpendicular to the optical axis it is sufficient to use polarizers with orientations of 0°, 45°, 90° and 135°. The arrangement of the polarizers is shown in Figure 3.6(d).

Stereo correspondence is computed between each pair of polarizers and the images are warped to the view of the center camera. By comparing the intensities of images captured with the polarizers it is possible to detect presence of polarized light. Figure 3.12 depicts an LCD display that polarizes light at 90°. The view from the center camera (with no polarizer) is shown in the first column. The second column is the average of the polarizers that pass the LCD display (the images are warped to the center view) while the third column shows the average intensity for the polarizers which block the LCD display content.

In this application, only 8 cameras are used to detect polarized light, therefore the remaining 17 cameras can be used to capture HDR images using a modified version of the previously described technique. In Figure 3.12(a) the effects of limited dynamic range can be seen in the Lego car which is overexposed while the jumping rabbit toy is underexposed. A tone mapped HDR image is presented in Figure 3.12(d), and, as shown in the insets, both the Lego car and toy rabbit are properly exposed.

### 3.7 Discussion and Future work

**Advantages:** Using a generalized assorted camera array offers the following advantages compared to traditional imaging systems: (1) Increased temporal resolution. Simultaneously capturing multiple views with the camera array allows for greater
temporal resolution in recorded images and video as well as reduced motion blur. (2) Flexibility in imaging modality. By changing the filter arrangement in front of each camera it is easy to configure the camera array to fit specific application domains such as multi-spectral imaging, HDR imaging, or polarization detection. (3) Adaptability to scene and illumination composition. Cameras in the array can be controlled independently or as a whole to adapt to novel lighting and scene changes without requiring laborious calibration.

**Limitations:** Camera arrays are not suitable for all imaging situations. In particular a camera array imaging setup is limited by scene complexity and proximity of scene elements. Point correspondences for highly complex scenes involving many thin objects and mirrors are not faithfully recovered with current stereo matching algorithms. Objects that are close to the camera array will not be present in the field of view of all of the cameras and thus cannot be properly imaged. The minimum distance is a function of the baseline of the camera array. Adding more elements to the array increases the effective baseline of the array, consequently increasing the minimum operating distance.

The imaging modalities presented in this chapter require a subset of cameras operating in the same mode which introduces redundant cameras into the array. As a consequence the baseline is increased which reduces the utility of the array. Removing this redundancy and reducing the array size is the focus of the next chapter.
Chapter 4

Dense Point Correspondence Across Color Channels

One class of successful strategies to mitigating resolution trade-off involves using a generalized assorted camera (GAC) array, where each camera has independent measurement characteristics. However, to align the different viewpoints, most previous efforts either make simplifying geometric assumptions or use duplicate cameras (as in the previous chapter). In this chapter, we tackle the challenge of GAC arrays in which each camera measures a totally different spectral component of the incident light field. We do not make any simplifying assumptions about scene geometry. We show accurate estimation of cross camera pixel correspondences (scene geometry) and robust estimation of the multi-channel visual information using a learned Gaussian prior on the registered multi-channel image content. We demonstrate our method by showing color image capture and light-field based re-focusing with a $2 \times 2$ array of monochrome cameras.

4.1 Introduction

Generalized Assorted Camera (GAC) arrays are recently becoming a popular and scalable technique for multi-channel imaging scenarios such as multi-spectral imaging [55], high dynamic range imaging [62], high-speed imaging [36], multi-focus imaging [84], light-field imaging and range imaging. Typically, each camera in the array acquires specialized information about one channel (e.g., one wavelength, one time
instant or a particular dynamic range) and a single, fused image is created from measurements acquired by the different cameras. The key to creating such an image is to establish high-quality scene correspondence across all the cameras in the array. To align the images from different cameras, most previous efforts make assumptions that allow for the easy application of multi-view geometry techniques. For example, some methods assume the scene is planar while others utilize two or more identical cameras in the GAC array to estimate scene correspondences. Unfortunately, traditional stereo or multi-view based methods for obtaining correspondences usually fail when each camera captures different scene properties.

In this chapter, we present a novel method for simultaneous accurate estimation of cross camera pixel correspondences (scene structure), and robust estimation of the multi-channel visual information using a generalized array of cameras. The framework relies on learning a Gaussian prior on the registered multi-channel image content. We learn the Gaussian prior using traditional methods in dictionary learning. Given the intrinsic and the extrinsic calibration of the cameras in the array, pixel correspondences are established by maximizing the posterior probability of scene geometry (depth map) given the multi-channel prior and the observed camera array data. Further, the same multi-channel prior is also used to robustly recover the registered multi-channel data enabling us to seamlessly incorporate multi-channel interpolation, smoothing, denoising, and even spatial super-resolution. While the framework is versatile and has many applications, the research behind this effort is still ongoing. We demonstrate the potential of this research by performing color image capture and light-field based refocusing using a $2 \times 2$ array of filtered monochrome cameras.
4.2 Related Work

Capturing red, green, and blue color channels that are isolated is beneficial in reducing channel cross talk and increasing spatial resolution when compared to common color filter arrays. Multi-sensor prism based cameras place a dispersing prism in the optical path and direct light to separate sensors [85]. However, costly custom hardware is required and the design fixed at the time of manufacture.

Alternately, color filters at the aperture can be used to physically separate the R, G, and B channels onto a single sensor. Robust fusion techniques are required to find disparity across color channels. Chang et al. [86] first capture an image of a projected pattern that is visible in all three channels to compute stereo correspondence and then capture a second image without the projected pattern which is fused using the computed disparity map. This approach is limited to static scenes and cannot be extended to video. Amari and Adelson [87] assume edges are invariant between the color channels and implement a simple sum of squared differences matching scheme to estimate disparity. Bando et al. [88] use a correlation based cost function which is limited to the application of RGB fusion.

The more general problem of image fusion across modalities (e.g. cameras recording in different wavelengths) is complicated by the lack of mutual information between views. Irani and Anandan [89] fuse cross-modality images of planar scenes by matching edges within a global framework. Lau and Yang [55] capture multi-spectral images by assuming scene points are sufficiently far from the cameras as to be auto-registered. Horstmeyer et al. [90] uses multi-modal aperture masks to reconstruct aligned images of scenes with multi-spectral and other characteristics.

Only the work by Frese and Gheta [91] and extended by Gheta et al. [92] offers a method for computing depth across color channels in a multi-spectral camera array.
This work requires that the center wavelengths of the filters must be close together and the filters must be arranged such that neighboring filters are placed next to one another in the array. Importantly, the reconstruction algorithm computes depth in a pairwise fashion and fuses depth maps as a post-processing step. A natural extension of the work presented in this chapter is to multi-spectral imagers where our more robust multiview geometry method would improve depth estimation.

4.3 Establishing Point Correspondences

In order to recover accurate composite images from the cameras in a rigid array we first reconstruct the 3D depth map of the scene by computing dense point correspondences between pixels in all of the views. By repeating the calibration process outlined in §3.3.1 for the $2 \times 2$ array of cameras we obtain a mean reprojection error of 0.1 pixels following bundle adjustment.

Dense 3D depth reconstruction is completed using the framework from §3.3.2. Unlike in the previous chapter where the cameras used for computing point correspondence shared the same channel, each camera in the $2 \times 2$ array is operating in a different channel. The simple SSD metric computes differences in pixel intensities which is not applicable when computing depth across channels so a new cost function must be developed.

4.3.1 Dense 3D Depth Reconstruction

Depth is found by using the same energy minimization scheme presented in equation 3.3, and repeated here

$$E(d) = E_D(d) + \lambda E_S(d).$$

(4.1)
The energy is minimized using graph cuts and a bilateral smoothing term is used to preserve discontinuity at object boundaries. The only modification is the introduction of a novel data term to compute disparity likelihood across color channels.

**Data term:** In order to compute the data term in equation 4.1, we learn a patch-based Gaussian data prior. A large number of $8 \times 8 \times 4$ patches are extracted from RGBY exemplar images of natural scenes to form a representative sample of RGBY patches. For this collection of patches, we subtract the mean and compute the covariance matrix. Using the covariance matrix we perform eigenvalue decomposition to learn the principle components matrix and corresponding singular values $m_i$ ($i = 1, \ldots, 256$). The dimensionality of the basis is restricted to the first $L$ principle components where $L$ is the largest singular value to satisfy the constraint

$$m_L \geq \frac{m_2}{100}.$$  

(4.2)

A $256 \times L$ principle component matrix $W$ serves as the reduced dimensionality basis for true RGB patches.

Using the plane sweep approach described in §3.3, at every proposed depth $8 \times 8 \times 4$ patches are extracted from the $M \times N \times 4$ datacube and rearranged into column vectors. Let $y$ be a $256 \times 1$ vectorized observation patch, $y$ is related to the unknown vectorized true patch $x$ by the relationship

$$y = Hx,$$  

(4.3)

where $H$ is the mapping from the true image space to the observed image space. In this thesis $H$ is an identity matrix, but this is not strictly required, and in future extensions the mapping will not be 1 to 1. If equation 4.3 holds, then the principle component basis of $y$ is $HW$, which is the linear mapping of the principle component
basis for $x$ using $H$. The residue of projecting $y$ onto $HW$ is

$$R_y = ||y - HW\alpha||_2,$$  \hspace{1cm} (4.4)

where $\alpha$ is the least-squares estimate of the projection coefficients,

$$\alpha = (HW)^\dagger y,$$  \hspace{1cm} (4.5)

and $\dagger$ denotes the pseudo-inverse. The energy associated with a given observed patch $y$ at a proposed depth $d$ is taken to be the residue of projecting $y$ onto $HW$. More generally, the energy is given as

$$E_D(d) = \sum_{y \in P} R_y(d),$$  \hspace{1cm} (4.6)

where $P$ is the set of all patches taken from the image. Patches that project outside of the domain of any image are excluded from $P$. Said another way, equation 4.6 takes every patch in the image to be a node. The cost of assigning each node to be at a given depth $d$ is recorded and stored in $E_D$. Graph cuts minimization incorporates a local smoothness cost and minimizes over this total cost term to assign depths.

### 4.4 Image Reconstruction

If a depth map is recovered perfectly for a given scene, outlying views may be projected onto the depth map and reprojected onto the reference camera, forming an accurate composite image. Perfect depth recovery is rarely achieved by any stereo algorithm for non-trivial scenes, and inaccurate depth maps lead to artifacts in the warped images. To minimize the effects of error in the depth map, we apply a Gaussian prior during reconstruction to improve the composite image quality.
4.4.1 Recovering the Composite Image

Composite images obtained by directly applying the warping algorithm described in 3.3.3 suffer from errors introduced by inaccuracies in the 3D depth map. These errors are common to all multi-view warping techniques; however they are more pronounced in our application as each image provides one of the color channels used to reconstruct the final image.

Color artifacts often force patches in the composite image to lie away from the subspace $W$ learned from ground truth patches. Therefore, the effect of artifacts can be suppressed by applying a linear minimum mean-square error (LMMSE) estimator to find $\hat{x}$, the estimate of the ground truth patch $x$. If we model $x$ as a mean-subtracted random Gaussian vector with normal distribution, $x \sim \mathcal{N}(0, \sigma_x^2 I)$, and add zero-mean Gaussian noise, $\nu \sim \mathcal{N}(0, \Sigma_{\nu \nu})$ to equation 4.3, then the observed patch $y$ is

$$y = Hx + \nu. \quad (4.7)$$

The observed patch is also a random Gaussian vector, $y \sim \mathcal{N}(0, \sigma_y^2 HHT + \Sigma_{\nu \nu})$. The LMMSE estimate for $\hat{x}$ is

$$\hat{x} = \mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y). \quad (4.8)$$

Since we have already performed mean substation, equation 4.8 simplifies to

$$\hat{x} = \Sigma_{xy}\Sigma_{yy}^{-1}1y. \quad (4.9)$$

The covariance matrices are defined as

$$\Sigma_{xy} = \sigma_x^2 H^T$$

$$\Sigma_{yy} = \sigma_y^2 HHT + \Sigma_{\nu \nu}$$
The covariance of \( \nu \), \( \Sigma_{\nu\nu} \), is a \( 64 \times n \times 64 \times n \) diagonal matrix where the first 64 diagonal entries correspond to the first camera, the second diagonal 64 entries to the second camera and so on. By controlling the entries of the covariance of \( \nu \), the linear estimator can be biased to favor a subset of views more than others. Warping the outlying camera views to the viewpoint of the reference camera introduces an asymmetry. While the warped images may have warping artifacts, the reference image cannot. This is expressed by setting the standard deviation of the noise differently for the reference camera, \( \sigma_{\text{ref}} < \sigma_{\text{outlier}} \). In practice, we set \( \sigma_{\text{ref}} \) to give an SNR of 60dB and set \( \sigma_{\text{outlier}} \) to give an SNR of 40dB.

### 4.5 RGBY Fusion

By using a small camera array with 4 monochrome cameras arranged in a \( 2 \times 2 \) grid, we can recover an RGB image and reconstruct the 3-D depth map of the scene. Our camera setup is to place a red, green, and blue color filter in front of three of the cameras in the array, and leaving the fourth open as a panchromatic camera. The cameras are tightly packed, reducing the baseline between camera centers and maximizing the shared field of view between the four cameras. The panchromatic (Y channel) camera is selected to be the reference view for the array.

Reconstruction of the 3D depth map follows from Section 4.3.1 by sweeping a virtual plane through 65 depths in the volume common to the four cameras. At every depth, overlapping patches spaced 4 pixels apart are extracted and their residue is calculated as in equation 4.4. The bilateral smoothness term is set so that the center of the sigmoid is placed in the range \([0.05, 0.1]\) with a fixed width of 0.005. The smoothness term is further weighted by \( \lambda \) to make the median value of \( E_D(d) \) roughly equal to one half of the number of depths to be searched over.
Figure 4.1: RGBY imaging: Four images captured with a $2 \times 2$ array of monochrome cameras with different color filters in front of each. The Y channel is panchromatic and not filtered.
Figure 4.2: **RGBY imaging:** Four images captured with a $2 \times 2$ array of monochrome cameras with different color filters in front of each. The Y channel is panchromatic and not filtered.

The experimental setup for our camera array is shown in Figure 4.1. We use four PointGrey Flea3 monochrome cameras (model: FL3-U3-13EM-C) in a $2 \times 2$ array. Three of the cameras are fitted with red, green, and blue visible spectrum filters and the fourth camera is unfiltered. As the sensitivity of the monochrome camera extends past the visible range we are implicitly incorporating some information in the near UV and near IR spectrum when recording with the Y channel. This, along with non-ideal filter responses, prevents the Y channel from being a linear combination of the other three channels. Each camera is triggered synchronously and exposes for the same duration. The analog gain for each camera is set independently to allow each camera to record images with pixel intensities in the range $[0, 255]$. 
Figure 4.3: **RGBY fusion**: A color image of the scene from Figure 4.2 is created using both SSD (left) and our prior-based model (right) as the data term in equation 4.1. The first row shows the recovered depth map with the corresponding color images in the second row. The bottom two rows offer expanded outsets comparing the quality of the two algorithms. Using a naïve SSD cost results in large errors in the depth map which manifest as color artifacts.
For the scene given in Figure 4.2, we compute the depth map with respect to the reference camera and warp the images to that view to generate a color image as shown in Figure 4.3. We compare the effects of using a naïve SSD data term and our prior-based data term. The first row of Figure 4.3 shows the depth map computed with the two data terms, the second row shows the color image generated using the depth maps, and the third row offers a zoomed-in view of the highlighted regions in the two previous rows. Since SSD computes differences in intensity between images it comes as no surprise that the depth map is of low quality when matching across color channels. For example, inspecting the orange in Figure 4.2 shows that it is high intensity (bright) in the red channel and low intensity (dark) in the green and blue channels.

While using SSD results in a poor quality image, by using our prior-based model we are able to recover a high quality color image. Close inspection of the enlarged regions in Figure 4.3 shows that the depth map using the proposed method is sharp and correctly determines the order of objects in the image—the orange is in front of the banana which is in front of the bird. The slight inaccuracies in the depth map result in warped image patches that do not fit the model. A post-processing step removes these artifacts by projecting these patches back to $W$. The end result is a high quality color image and the corresponding depth map.

**Depth based refocusing**  Given an all-in-focus scene and corresponding depth map, it is possible to perform digital refocusing [31]. Such depth-based refocusing can be either applied blindly when the camera characteristics such as the focal length, focus distance, aperture, and circle of confusion are unknown, or given this information a more accurate refocusing is possible. From Greenleaf [93] we have that the
Figure 4.4: **Digital refocusing**: For the scene shown in the upper left corner (Y channel) the depth map and color image are computed. Using the depth map it is possible to refocus the image digitally (bottom row). The focal plane sweeps from the front to the back of the scene. Notice that in this more complicated scene a few color artifacts are present near the mug.

The hyperfocal distance is

\[
h = \frac{f^2}{Nc} + f, \tag{4.10}
\]

where \(h\) is the hyperfocal distance, \(f\) is the focal length, \(N\) is the f-number, and \(c\) is the circle of confusion. Given the focus distance \(s\), the near \((D_n)\) and far \((D_f)\) distances of acceptable sharpness can be computed as

\[
D_n = \frac{s(h-f)}{h+s-2f}, \tag{4.11}
\]

and

\[
D_f = \frac{s(h-f)}{h-s}. \tag{4.12}
\]

The experimental setup used to capture the scene shown in Figure 4.4 the camera...
parameters are: \( N = 4, \ f = 12.5\text{mm}, \ c = 0.006\text{mm}, \ s = 1,520\text{mm}. \) Using a larger aperture \( (f = 1.8) \) the new hyperfocal distance can be computed using equation 4.10 and the new near and far focus distances are found using equations 4.11 and 4.12 respectively. For any scene point that is outside of the range \([D_n, D_f]\), the circle of confusion can be computed using equation 4.10 which is then converted into a blur kernel which is used to blur the image. By changing the distance to the focal plane \( s \) it is possible to digital refocus the image at any depth in the scene, an example of this is shown in the bottom row of Figure 4.4. The near depth is focused on the red bird, the middle distance is focused on the large monkey, and the far distance is focused on the hanging monkey and background. The mug is always out of focus because it is the closest scene element to the cameras.

A high quality depth map is required for digital refocusing as any errors in the depth map or subsequent aligned image are compounded when refocusing. For example in Figure 4.4 the depth map for the right edge of the mug is inaccurate which leads to color artifacts. This is not readily obvious in the all-in-focus image but stands out in the refocused image.
Chapter 5

Conclusions

The proposed Flutter Shutter Video Camera in Chapter 2 comes as an advancement over two recently proposed computational cameras [8,9] that use a per-pixel shutter. The per-pixel shutter control enables different temporal codes at different pixels. In contrast, in our design, the control of the shutter is global and, hence, all pixels share the same temporal code. At first glance, this might seem like a small difference—yet, the implications of this are profound.

A global shutter leads to ill-conditioned measurement matrix. An easy way to observe this is to study properties of the adjoint operator of the measurement matrix defined in equation (2.2). The adjoint of an observed video is a high-resolution video with zeros for frames at time-instants when the shutter is closed. The abrupt transition from the video signal to the nulls introduces high temporal frequencies. Hence, the adjoint operator is highly coherent to high frequency patterns and is predisposed to selecting high-frequency atoms when used with a sparse approximation algorithm (as in [9]). In contrast, the adjoint operators associated with both the P2C2 and CPEV [9] are less coherent with such bases. Hence, it is much easier to obtain higher quality reconstructions with these. Shown in Table 5.1 is a comparison of reconstruction performance using 3 different modulation schemes, per-pixel flutter shutter as in P2C2 [8], per pixel single short exposure as in [9] and global flutter shutter video camera as proposed in this chapter. It is clear that the simple and easy to implement architecture of FSVC results in reconstruction performance that is about 6 dB lower.
Table 5.1: **Comparison of compression vs reconstruction PSNR in dB** for compressive video acquisition architectures in the absence of observation noise. FSVC is a much simpler hardware architecture than either P2C2 or CPEV, but it results in reasonable performance.

<table>
<thead>
<tr>
<th>Compression Vs PSNR (dB)</th>
<th>P2C2 [8]</th>
<th>CPEV [9]</th>
<th>FSVC (this chapter)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>41.5</td>
<td>41.8</td>
<td>35.6</td>
</tr>
<tr>
<td>8</td>
<td>37.4</td>
<td>38.8</td>
<td>33.1</td>
</tr>
<tr>
<td>12</td>
<td>34.2</td>
<td>36.0</td>
<td>29.8</td>
</tr>
<tr>
<td>18</td>
<td>31.2</td>
<td>34.5</td>
<td>29.1</td>
</tr>
</tbody>
</table>

than per-pixel coding architectures. Further, these experiments did not include observation noise. Since CPEV has a very low light throughput (1/compression), the performance of CPEV will degrade significantly (compared to P2C2 and FSVC) in the presence of noise especially in dimly lit scenes.

In Chapter 3 we presented a framework for a generalized assorted camera and demonstrated examples of capturing multi-spectral and high dynamic range images and video. Robust point correspondence using a subset of the camera elements allows all of the camera views to be warped to an arbitrary viewpoint within the camera array. Moreover, the simultaneous capture of scene information allows for imaging of dynamic scenes and video capture.

In Chapter 4 we offer a method for removing the redundant cameras used in Chapter 3 by computing dense 3D correspondence across color channels using an exemplar–based prior. Whereas in Chapter 3 more than $N_C$ cameras are needed to recover $N_C$ color channels, we have shown that $N_C$ cameras can be used to recover $N_C$ color channels. A future extension of this work will be to use hyper-spectral
multiplexing so that fewer than $N_C$ cameras will be needed to recover $N_C$ color channels.

**Future Work:** The imaging modalities presented in Chapters 3 and 4 are just a few examples of the richness of the camera array imaging space and many areas remain open for future exploration. In the short term, we would like to (1) extend cross channel correspondence matching to narrowband multi-spectral imaging, (2) investigate multiplexed multi-spectral imaging to further reduce the number of cameras needed in the array, and (3) extend the work in Chapter 4 to application specific tasks such as performing multi-spectral aided segmentation.

In the long term we would like to investigate (1) high-speed and low-speed imaging for motion blur removal, (2) variable focal length imaging, and (3) super-resolution for arbitrary scenes. We would also like to allow for flexible array architectures with the ultimate goal of fusing across multiple imaging domains in a non-rigid array implementation.
Bibliography


