Signed Coded Exposure Sequences for Velocity and Shape Estimation from a Single Photo

by

Tyler Hutchison

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Masters of Engineering in Electrical Engineering at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2011

© Massachusetts Institute of Technology 2011. All rights reserved.
Signed Coded Exposure Sequences for Velocity and Shape

Estimation from a Single Photo

by

Tyler Hutchison

Submitted to the Department of Electrical Engineering and Computer Science
on February 1, 2011, in partial fulfillment of the
requirements for the degree of
Masters of Engineering in Electrical Engineering

Abstract

In this thesis, I analyze the benefits of signed coded exposure for velocity and shape detection of moving objects. It has been shown that coded exposures enhance de-blurring of motion blurred photos [28]. However, these non-negative binary codes (1 or 0) only suggest opening and closing of the shutter to allow or prevent light from entering the camera.

Signed codes (+1 or -1) for camera exposures offer accumulation or removal of light over the course of a single exposure. I show that signed codes provide dramatic benefits over unsigned code for motion estimation due to better frequency domain properties and auto-correlation characteristics. I analyze the space of such codes with invertibility analysis and a cross-correlation metric. Motion estimation is important to a number of computer vision problems such as tracking, segmentation and recognition. New emerging hardware, in the commercial and research domains, provides signed coded for exposure, but their full capabilities have not been explored. Part of my efforts involved experimenting with the electronics of such cameras. The emphasis in this thesis is on the computational aspects of a framework which employs new codes for motion parameter estimation.

I demonstrate the ideas on a variety of synthetic images and real-world photographs. I hope the cameras and theory of signed coded exposure will be a new motion-analysis tool in the field of computational photography.

Thesis Supervisor: Dr. Ramesh Raskar
Title: Associate Professor
Acknowledgments

My sincerest thanks to my thesis advisor, Ramesh Raskar, for the chance to come back to the Camera Culture to participate in a number of cutting edge projects. His guidance was integral to my growth when I was a UROP, and ever more so during my graduate year. He also recognized my desire to explore varied projects, giving me the chance to work on everything from circuit consultation, to laser displays, to three-dimensional animation, to cutting edge technology only available at Lincoln Labs.

I would also like to acknowledge my other mentors on projects of which I was a part. Many, many thanks to Daniel Schuette, Bill Freeman, Doug Lanman, and Daniel Saakes. Of course, thanks to Ahmed Kirmani for his additional guidance, friendship, and odd taste in music. Finally, thanks to Andreas Velten for continually putting up with my sense of humor and my verbose way of writing MATLAB scripts. Our lab group was one of constant discussion and interaction, and I appreciate every bit of it.

Many thanks to my good, good friends who allowed me to vent when things didn't go so well, rampantly discuss ideas when I wasn't sure what to do next, and sit in silence with a furrowed brow when necessary. Thanks to Violetta Wolf, Anton de Winter, James Hollenbach, Jesse Carr, and Jamie Waltz for being wonderful roommates. Additional thanks to James Hollenbach for sometimes being a roommate even when he lived across the country. Thanks to Anne Schweitzer for being there when I needed her and inspiring me to work when I was ready to give up. Thanks to Rachel Miller who cheered me on. Thanks to Michael Craig for laughing at my silly complaints. And thanks to many more friends who helped along the way.

Above all, thanks to my family. Thanks Mom, Dad, Jill and Cole for always offering care packages and support. Also, of course, always letting me know when something was broken at home, so that I could fix it... or rip it apart.
THIS PAGE INTENTIONALLY LEFT BLANK
## Contents

1 Introduction ........................................... 13
   1.1 Contributions ..................................... 16
   1.2 Benefits .......................................... 17
   1.3 Limitations ....................................... 17

2 Background ............................................ 19
   2.1 Motion Deblurring Introduction ....................... 19
   2.2 PSF Estimation ..................................... 21
      2.2.1 Frequency Spectrum Analysis ..................... 21
      2.2.2 Optical Flow ................................... 26
      2.2.3 Multiple Image Capture ......................... 27
      2.2.4 Motion From Blur - α Matting Techniques ....... 28
   2.3 Deblurring Methods ................................ 30
      2.3.1 Classical Methods - Richardson-Lucy and Wiener Filtering .. 30
      2.3.2 Homography Model ................................ 32
      2.3.3 Frequency Shaping - Flutter Shutter ............. 33
   2.4 Blind Deconvolution Methods ......................... 35
      2.4.1 Brief Overview .................................. 35
      2.4.2 Blind Deconvolution with Frequency Spectrum Methods .. 37
      2.4.3 Single Photo, Iterative Techniques ................ 40
      2.4.4 Multi-frame Approaches .......................... 41
      2.4.5 Hardware Modifications .......................... 42
      2.4.6 Blind Deconvolution with Coded Exposure .......... 45
3 Benefits of Signed Coding

3.1 Information Capture
3.2 Background Subtraction
3.3 Width and Velocity Determination
3.4 Optimal Code for Motion Discrimination

4 Width and Velocity Estimation for Synthetic Scenes

5 Width and Velocity Estimation for Photographs Captured by a High-Speed Camera

5.0.1 Analysis of Real-World Results
5.0.2 Uniform Intensity Object
5.0.3 Object with Text
5.0.4 Object with Text and Intensity Differences
5.0.5 High-Frequency Texture
5.0.6 Velocity Estimation and Deblurring Example
5.0.7 Spatially Varying Motion

6 Discussion and Future Work

6.1 Discussion
6.2 Future Work
List of Figures

1-1 Accurate velocity and width estimations for synthetic scenes

1-2 New hardware provides for signed coded exposures.

1-3 A comparison of signed and unsigned codes that shows the frequency shaping capabilities of both code types, and the additional benefits of signed codes.

2-1 Simplistic deblurring results in noise amplification

2-2 The cepstrum of a blurred photograph

2-3 The flutter shutter camera retains more frequency information.

2-4 Blind deconvolution estimates the PSF and deblurred image directly. Or iterative techniques can minimize an error metric.

3-1 Signed codes preserve more information of $l(t)$ due to the higher sampling rate of signed codes.

3-2 The incoming light to a pixel over time, $l(t)$, is dominated by background, but signed codes can attenuate low spatial frequencies.

3-3 Correlating two blurred signals results in pulse mixing, which allows for correlation peak minimization if signed codes are used.

3-4 A method for comparing two random textured objects with slightly different widths and velocities shows how peaks in the signed coded correlation diminish.

4-1 The velocity/width estimation pipeline
4-2 The LUT is created using images of synthetically blurred bars of varying width and velocity.   

4-3 Correlation maps between the various blurred bars. 

4-4 An example of determining width and speed for synthetic data with accuracy despite a complex background. 

4-5 Spatially varying motion can also be recovered by this simple pipeline.

5-1 A third person view of the simple setup. 

5-2 Eight captured datasets where mean errors have been compared. 

5-3 The row-by-row reconstruction of a Styrofoam cup. 

5-4 The row-by-row reconstruction of a bottle. 

5-5 The row-by-row reconstruction of a marker. 

5-6 The row-by-row reconstruction of a vase. 

5-7 An entire deblurred scene when signed codes have been used to estimate velocity. 

5-8 Spatially varying motion of real-world scenes.
List of Tables

2.1 Approximate shape of the PSF for standard motion types . . . . . . . . 24
2.2 Transfer functions, $H$, of typical blur patterns. . . . . . . . . . . . . . 38
Chapter 1

Introduction

Motion estimation in relation to deblurring is an important problem in computational photography. High-speed cameras provide a solution for certain applications, but illumination intensity and sensor bandwidth restrict their use to well-lit environments. The computational complexity of deconvolution methods also limit their application domains. However, motion blur patterns encode a great deal of information concerning object velocity and shape; analysis of blurred photos from conventional camera exposures can recover object motion, provided the photo has a high signal-to-noise ratio and the scene has a low-frequency background. While object motion can be estimated, conventional exposures create nulls in the frequency spectrum and the photo cannot be deblurred without specific priors or more sophisticated deconvolution algorithms. To resolve this limitation, the coded exposure camera introduced by Raskar et al. [29] modulates the incoming light by rapidly opening and closing the shutter over the exposure sequence. A pseudorandom, binary-valued coded exposure (i.e., a \{1, 0\} code) is employed. This code is discretized into a finite number of equal-duration chops, during which the attenuation introduced by the shutter is held constant. This camera modification allows for deblurring without noise amplification, but cannot provide a code simultaneously-optimized for motion deblurring and point spread function (PSF) estimation [1].

Coded exposure assists in motion deblurring by eliminating nulls in the Fourier transform of the PSF. However, object motion must also be estimated to deblur a
Figure 1-1: Accurate velocity and width estimations for synthetic scenes. Four blocks, shown in gray, are translated by 10, 20, 30, and 40 pixels during a simulated coded exposure. (a) An idealized image captured during one chop. (b) An image obtained with a \{1, 0\} code, which both accumulates and blocks light. Note the strong background presence. (c) An image obtained using a \{1, -1\} code, with black and white denoting negative and positive values, respectively. Note that the background is suppressed. (d) The \{1, -1\} code allows for improved estimation of object width and velocity. On the left, the estimated object widths are shown in white (for each image row). On the left, the corresponding estimated object velocities are shown in red. In both cases, ground truth is denoted by the green line.

photograph. An efficient method for simultaneously estimating object velocity and shape from signed coded exposure sequences (i.e. \{1, -1\}) sequences will be explored. Specifically, a look-up table (LUT) is constructed using synthetic blurs, containing object velocities and widths of interest. Each element of the table stores the expected image of a variable width light source under the known coded exposure sequence. Representations of the steps in the image deblurring pipeline are shown in Figure 1-1. To estimate object motions, a single coded exposure photo is recorded. Next, each moving object is segmented from the background. Afterward, the velocity and width are estimated, independently for each object, by selecting the LUT element with the maximum cross-correlation with the object blur pattern. This information could be used for deblurring, data collection, or other post-processing steps.

While the proceeding algorithm could be applied using existing \{1, 0\} codes, it was observed that recently-developed sensors enable signed codes, spanning attenuation
Figure 1-2: New hardware provides for signed coded exposures. They have been used for ambient light suppression and high-speed motion capture. (a) A photograph of a gun being fired, as recorded by a sensor supporting temporally-modulated, signed charge integration. The sign of the exposure is rapidly varied such that bullet velocity is encoded in the motion blur pattern. Note that stationary objects do not appear in the photo. (Figure reproduced from [44].) (b) Using the Eclipse Pixim camera, ambient light can be removed from the scene before sensor read-out. (Figure reproduced from [28]).
on the $(-1, 1)$ interval. One such sensor is the Eclipse by Pixim, Inc. [28]. This sensor uses a coded exposure to suppress ambient illumination. An infrared strobe is synchronized to a periodic set of integration intervals. The remaining intervals, during which the strobe is off, are subtracted from the first before read-out. Tyrrell et al. [44] describe a sensor for which incident photons generate electron-hole pairs that either add to or subtract from the accumulated charge in a programmable manner. The sensor has been used for high-speed motion capture. For instance, a bullet fired at 397 m/s as shown in Figure 1-2. Reich et al. [32] demonstrate a similar sensor containing multiple storage cells per pixel, each with a built-in electronic shutter. These emerging architectures allow for the use of signed modulation within computational photography. In this paper we consider their application to motion estimation and deblurring. As will be shown in Section 3, signed coded exposures significantly improve cross-correlation-based motion estimation.

1.1 Contributions

This thesis makes three contributions:

- I consider signed coded exposures $\{1, -1\}$ motivated by emerging sensors that allow removal of charge by time-varying modulation. I demonstrate that signed coded exposures better preserve motion information than unsigned coded exposures $\{1, 0\}$.

- I analyze signed and unsigned coded exposures for their benefits to object velocity and shape estimation. In order to estimate these parameters, I implement an efficient and simple cross-correlation-based LUT table algorithm. Based on an invertability and cross-correlation metric, I describe optimal codes for velocity/width detection.

- Using simulated images and real-world photographs, I analyze the benefits and limitations of signed coded exposures for motion parameter estimation.
1.2 Benefits

Signed codes have a number of unique benefits which will be described in detail in Section 3. One benefit over traditional, binary-valued coded exposures, is that information about the object is captured at every time step. Since a signed code is integrating charge at every time step during an exposure, additional information is captured. A broadband code, suitable for deblurring, also captures a more complete intensity profile. Additionally, background subtraction is performed automatically on the photo. Segmenting motion from background is often a challenge in motion estimation and deblurring. Here, that step is performed automatically since stationary information is removed with a suitable code. Background removal also provides for a decreased relative amplitude of the background component which is unavoidably integrated into the blur profile. Finally, by examining the correlation of two similar blur kernels, it is found that signed codes provide greater differentiation of blurred objects based on a cross-correlation metric.

1.3 Limitations

Despite the benefits that signed codes offer to motion discrimination, there are problems with such an approach. Objects with occluded or merged motion blur patterns present additional challenges. Since signed codes remove background information, there is significant attenuation of slow-moving (i.e. low spatial frequency) objects. To capture an photo without background attenuation, a camera that records multiple photos simultaneously (see [32] or multiple cameras could be used. However, we consider only motion in a single frame from a single camera. Finally, high-frequency textures and low-brightness objects create false peaks in our cross-correlation metric; this failure case will be explored in Section 5.
Figure 1-3: A comparison of signed and unsigned codes that shows the frequency shaping capabilities of both code types, and the additional benefits of signed codes. (a) and (c) show the frequency spectrum for an unsigned and signed code respectively. Both code types are able to preserve high spatial frequencies. (b) and (d) show the cross correlation between scaled codes (i.e. $\text{corr}(f(x), f(ax))$) based on a simulated object velocity. Signed codes have a clear peak at the max of the autocorrelation (i.e. $\text{corr}(f(x), f(x))$). (e) shows that signed codes perform better in minimizing a discrimination metric between nearly identical codes. From code sequences plotted in the middle for signed and on the right for unsigned codes, it is clear that high frequency codes result in clearest discrimination under this metric. The worst code, at the bottom of the plot, is a step function which preserves mainly low frequencies.
Chapter 2

Background

2.1 Motion Deblurring Introduction

The process of motion deblurring is often addressed in separate ways. The basic deblurring pipeline, in informal terms, is as follows:

1. Capture blurred photograph

2. From camera parameters and the photograph, estimate the object motion (i.e. PSF)

3. Determine the sharp image using the motion information

The processes of PSF estimation and deblurring are considered separately. Papers on the full motion deblurring pipelines, usually called blind deconvolution, are directly inspired by papers on the separate problems. In order to motivate both problems, a simple photo blurring model will be considered. A simple convolution model for image capture has been the motivation for a number of PSF estimation and deblurring concepts. The spatial domain equation is

\[ i_b = h \ast i_s + n \]  \hspace{1cm} (2.1)

where \( i_b, h, i_s \), and \( n \) are the blurred image, the PSF, the sharp image, and the image noise respectively. The most important assumption that this model takes into account
is that the PSF is spatially-invariant, which may or may not be a safe assumption and will be discussed in future sections. Convolution is multiplication in the frequency domain, so more intuition may be offered if the equation is transformed.

\[ I_b = HI_s + N \]  \hspace{1cm} (2.2)

Throughout the paper, capital letters will designate the Fourier transform of the corresponding signal, that is, \( I_b = \mathcal{F}(i_b) \). Assuming the noise, PSF, and captured blurred photo are known, \( I_s \) can be easily solved for as

\[ I_s = \frac{I_b - N}{H} \]  \hspace{1cm} (2.3)

However, \( H \) is often ill-conditioned for simple inversion. An example suffices for demonstration. In this simple scenario, a one-dimensional camera opens its shutter for \( T \) seconds and captures a delta function object moving at a constant velocity, \( v \). The captured image is a line proportional to \( vT \). If plotted across the image (i.e. the spatial domain), the captured image looks like a box function. The Fourier transform of a box function is a sinc with nulls at locations proportional to integer multiples of \( f = \pm \frac{1}{vT} \). When solving for \( I_s \), it was assumed that the PSF was defined at all spatial frequencies. Simple motion, such as constant velocity, results in noise amplifications when the PSF is inverted. Further complicating the problem is the fact that it was initially assumed that the PSF was known. Even for a constant velocity, high frequency texture, object width, and angle of motion may complicate the determination of the velocity. Researchers have attempted to solve these problems using post-processing, clever capture methods, and camera hardware modifications. Figure 2-1 shows an example of the problems with simplistic deblurring.
Figure 2-1: Attempting to deblur simple photos with a known PSF (in this case, a uniform disk) results in noise amplification with an overly simplistic approach. (a) Shows an original frame. (b) Shows the photo blurred by a uniform disk. (c) Shows the naive deblurring approach applied to each color channel separately.

2.2 PSF Estimation

2.2.1 Frequency Spectrum Analysis

Frequency spectrum analysis for PSF estimation will first be considered. It was noted that frequency spectrum nulls occur at predictable locations in the frequency spectrum based on the type of motion the scene, or object, experienced. Rekleitis [33] considers the power spectrum of a single image for arbitrarily large blurs. It is not assumed that anything is known about the blur pattern, including orientation, but that the motion is constant velocity. Thus, one expects a two-dimensional sinc-like function if the two-dimensional Fourier transform is taken. A single image patch is windowed, manually, where the patch has a locally spatially invariant PSF. The Fourier transform of the patch results in a spectrum that has the same orientation as the motion and a period equivalent to the magnitude of the spatial blur. Direction of motion is detected using steerable Gaussian filters. The precise motion blur is determined using a modified form of cepstral analysis. The basic algorithm is as follows.

1. Isolate motion patch
2. Gaussian filter to reduce ringing and zero pad the spectrum

3. Take the logarithm of the magnitude of the Fourier transform (cepstrum)

4. Use a second order Gaussian derivative filter aligned to the orientation of motion (steerable filter)

Ideally, merely performing analysis on the spectrum of the photo alone should produce enough information to determine angle and magnitude of the blur. The cepstrum method makes the analysis more resilient to noise. In practice, however, the added resiliency is not usually enough for precise motion detection from realistic images. Still, the method offers tools to use in more advanced techniques. The cepstrum, ignoring noise, is defined as in Equation 2.4,

\[ C(i_b) = \mathcal{F}(\log|F(i_b)|) \]  \hspace{1cm} (2.4)

we can see that we can separate the PSF and the image spectrum as in Equation 2.5. An example cepstrum can be seen in Figure 2-2

\[ C(i_b) = \mathcal{F}(\log\left|H I_s\right|) = C(h) + C(i_s) \]  \hspace{1cm} (2.5)

Since the PSF may have strong, periodic negative peaks, especially in the case of uniform motion captured with a normal camera exposure, it is possible to detect these peaks. The Fourier transform of this periodic signal results in a strong spike at the frequency which describes the blur width. Results were shown to be accurate for artificial blurs of stationary objects where the kernel was also generated using anti-aliasing techniques. Real-world results were also tested, though could not be sufficiently analyzed since ground truth was not known. Qualitative analysis suggested that the results were fairly accurate.

Though cepstrum analysis provides some noise resiliency, real-world photos with arbitrary motions are problematic. Therefore, Ji and Liu [16] provide an analysis for more general motion identification using the image power spectrum. Types of motion analyzed here include uniform velocity, highly accelerated motion, and uniformly
Figure 2-2: For a scene blurred by uniform motion, spikes at frequencies proportional to the length of motion, yield PSF information. (a) Shows a photo blurred by horizontal uniform motion. (b) The two-dimensional Fourier transform of the photo results in a clear sinc function in the direction of motion blur. (c) Analyzing a few averaged rows results in characteristic periodicity by the log of the fourier transform. (d) The cepstrum of the photo yields a spike corresponding to the frequency of periodicity.
Type of Motion | Shape     | Value  | For i equals
---|-----------|--------|-------------------
Highly accelerated | Ramp     | \(i - 1\) | \(i = 0, \ldots, M - 1\)
Uniformly accelerated | Trapezoid | \(M + i\) | \(i = 0, \ldots, M - 1\)
Uniform velocity | Square   | 1      | \(i = 0, \ldots, M - 1\)

Table 2.1: Approximate shape of the PSF for standard motion types

accelerated motion. Their analyses are performed using the cepstrum of the image gradient. Also, the Fourier-Radon transform is introduced to estimate the parameters of the kernel with improved noise robustness. The Fourier-Radon transform is the Radon transform of the Fourier transform. The Radon transform will be described in more detail later since it is the core of another type of analysis.

To reiterate an important fact, zero patterns and PSF symmetry only appear for uniform velocity motions. For image gradients, patterns exist in three motion types: uniform velocity, highly accelerated, and uniformly accelerated motion. Common equations are used for the rough blurring functions of the three motion types, which may need to be normalized depending on application. The equations are represented as the shapes in Table 2.1 for highly accelerated, uniformly accelerated and uniform velocity respectively.

For decelerated motion, the resulting vectors can be reversed.

When viewing the non-gradient cepstrum, the periodicity is nearly impossible to determine for more complicated motion that constant velocity. When the cepstrum of the image gradient are compared, it clearly shows the advantage of image gradient analysis. Still important to this analysis, however, is determining the type of motion to select the exact motion type. This step can be performed by frequency domain phase analysis, but a close-form solution was difficult to determine. Qualitative analysis was adequate to restore a blurred photo. Finally, the Fourier-Radon transform was explored to provide additional noise robustness of the analysis. The Fourier-Radon provides noise resiliency since it is able to pick out strong curve patterns. It’s original use was in detecting lines through a scattering medium in tomography.

It is useful to compare a number of frequency domain methods that have been discussed. The details of steerable filters and the Radon transform will be shown
here. Krahmer et al. [18] look at number of approaches including cepstrum analysis, steerable filters, and the radon transform. The different approaches are compared for their accuracy in determining motion direction and extent of blur for linear motion paths. A linear motion path assumes the exposure is short enough to consider motion for particular direction and velocity. A simple way to determine direction is to take a two-dimensional Fourier transform of the blurred image. A two-dimensional sinc function results, oriented in the direction of the blur. To estimate the motion direction, one can draw a straight line from the origin to the first negative peak. The angle of motion is then the inverse tangent of the slope. Three more advanced methods are compared to each other. The cepstrum is obtained as described previously. Gaussian steerable filters are determined by using basis vectors which can be defined for a desired angle. After the filter is applied to the transform of the image, the highest filter response determines angle of blurring. For reference, the bases for second order steerable filters used in motion angle detection literature are given in Equation 2.6 [12].

\[
\begin{align*}
G_{2a} &= .9213(2x^2 - 1)e^{-(x^2+y^2)} & k_a(\theta) &= \cos^2(\theta) \\
G_{2b} &= 1.843xye^{-(x^2+y^2)} & k_b(\theta) &= -2\cos(\theta)\sin(\theta) \\
G_{2c} &= .9213(2y^2 - 1)e^{-(x^2+y^2)} & k_c(\theta) &= \sin^2(\theta) \\
G_2^\theta &= k_a(\theta)G_{2a} + k_b(\theta)G_{2b} + k_c(\theta)G_{2c} & (2.6)
\end{align*}
\]

The Radon transform can also detect angle and magnitude of blur. The transform should be computed for \( t \) equally spaced angles from 0 to 180\((1 - \frac{1}{t}) \). A unique property of the Radon transform is it’s ability to detect curves in the presence of noise which makes ideal for direction and magnitude detection of image patterns in the frequency domain. The Radon transform is given by Equation 2.7.

\[
R(f)(x, \theta) = \int_{-\infty}^{\infty} f(x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta)dy \quad (2.7)
\]
The highest assumed value results in the angle and magnitude of the blur kernel. Results were also analyzed in depth. Trends calculated here persist in other papers which consider similar approaches. Cepstral analysis is very accurate for low noise, but fails in high noise cases. The Radon transformation is good especially for long blurs, and is even accurate with low signal-to-noise ratios for long blurs. Steerable filters encountered angular matching problems even with no noise, though behaved moderately for long blurs in the presence of noise.

2.2.2 Optical Flow

In order to motivate other PSF estimation techniques, a minor digression from motion blur is important. Authors often assume that the optical flow and the motion field are the same in order to calculate motion blur. However, optical flow, the estimated motion variation of image brightness pattern from an image, and the motion field, the representation of motion from a particular perspective on a two-dimensional field, are not necessarily the same unless certain conditions are met. One of the most important conditions is that the brightness of the moving objects does not change over the course of the exposure. For instance, an object with a specular highlight does not meet this constraint since the location of the specularity changes for long movements. The optical flow constraint can be written as Equation 2.8

\[
\frac{\partial I}{\partial t} = \frac{\partial I}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial I}{\partial y} \frac{\partial y}{\partial t} \quad (2.8)
\]

Meaning, the change in brightness of an image is only dependent on the movement of the object, not a change in object intensity. When the optical flow constraint is met or uniform intensity over the course of the exposure, it refers to this assumption. For short enough exposures, the constraint is often assumed to have been met. Accurate methods exist to estimate the optical flow (and thereby motion if the optical flow constraint has been met). Brox et al. [6] developed a pioneering technique in optical flow registration between two images. The results are dependent on minor, and typical assumptions. Also, the results are resilient to noise and parameter variation.
Three main assumptions are made. Grey value constancy, that movement does not change an objects grayscale values. This assumption does not directly manifest itself as the optical flow constraint, since, as the authors point out, the constraint in its modern form only works for linearized (i.e. short) movement. Additionally, a gradient constancy value is also included which allows for small variations in the grayscale level. The assumption is also used in a non-linearized form, given as Equation 2.9

$$\nabla i_s(x, y, t) = \nabla i_s(x + u, y + v, t + 1)$$  (2.9)

Finally, a piecewise smooth approximation for the optical flow field is included. The technique is then motivated as a coarse to smooth approach. Since for large movements, the optical flow may be non-linear, minimization of an optical flow function could get stuck at a local minimum. Therefore, a coarse estimation is first made. Then a smooth estimation is used to determine the precise optical flow field. An energy function is derived which penalizes deviations from the aforementioned constraints. After extensive manipulation, the highly nonlinear system can be solved by numerical methods to produce the optical flow. The coarse to fine warping approach is used to achieve minimization efficiently due to discovered similarities between the authors’ approach and warping theory. Through the nonlinear minimization term, the algorithm achieves a great deal more accuracy than previous methods. Note that this method is for the registration between two, not necessarily blurred, images.

### 2.2.3 Multiple Image Capture

In order to estimate the exact PSF, some approaches acquire multiple images, possibly with different characteristics. In order to capture a motion field, capturing images with two different exposure times may be useful [36]. Three images are captured in this case. An initial short exposure image, a long exposure, followed by a second short exposure. The short exposures enable direction prediction and sharp image details. The long exposure constrains the solution space for reconstruction and increases the range of motions that can be captured. This work uses multiple images to alleviate
a number of problems in many optical flow reconstructions. The primary problems
addressed are solving large motion disparities and compensating for some occlusion
effects. In theory, the technique does not enforce linearity for the object motion, but
test data considered met this constraint. In order to compensate for occlusion effects,
the authors use a parameterized model. The initial conditions are set by the short
exposure. Then, occlusion effects can be minimized by switching between the image
formation which resulted from the initial or final short exposure image at the moment
of occlusion. The object from the initial or ending image is simply propagated along
the motion vector. This assumes that a point is occluded or not occluded between
one image, but not both.

2.2.4 Motion From Blur - α Matting Techniques

An important PSF estimating technique has been used in a number of complete blind
deblurring pipelines. The technique uses ideas common to alpha matting and is known
as Motion from Blur (MFB) [11]. A similar constraint to the optical flow constraint
is enforced and the simple convolution image model is used. The authors develop a
motion constraint known as the alpha-motion blur constraint model. Motion can be
estimated without deblurring using the alpha channel technique. The alpha-motion
blur constraint limits blur estimation to local pixels so that the number of degrees of
freedom are reduced and the motion can thereby be estimated from a single image
without an image deblurring step. The alpha matting approach reduces the number
of dimensions, however, a degree of freedom remains due to the presence of only a
single image. Still, using minimization, the blur is estimated. Alpha channel detection
for deblurring has been used previously, but the novel constraints developed in this
paper are new [17, 38] Due to its importance in a number of blind motion deblurring
techniques, more consideration will be given to the mathematical details. While the
simple convolution model, in essence, is used for inspiration, the blur kernel, \( h \), is
broken into orthogonal bases and vectorized as \( b = (l \cos \theta, l \sin \theta)^T \), where \( l \) is the
length of the blur and \( \theta \) is the angle of the blur. It is then proven that, evaluated at
a position \( p \)
\[ \nabla i_b|_p \cdot b = i_s(p + \frac{b}{2}) - i_s(p - \frac{b}{2}) \quad (2.10) \]

This expression relates the local motion blur to the local gradient and two un-blurred pixels (i.e. pixels of \( i_s \)). This consideration alone allows for motion blur estimation. \( \alpha \)-channel modeling allows for a systematic approach. Essentially, the authors seek to model the original image as foreground or background and then apply this to the formation of a blurred image. \( \alpha \)-channel techniques allow for edge values can be normalized to a 1 or 0 (i.e. foreground or background respectively of the sharp image since opaque objects are assumed), many image processing problems become significantly easier. From experimental evidence, \( \alpha = 0 \) mostly when \( \nabla \alpha = 0 \) in the blurred image. Cases where \( \alpha = 1 \) and \( \nabla \alpha = 0 \) are treated as outliers. For positions in the blurred image where \( |\alpha| \neq 0 \), it is assumed that \( \nabla \alpha = \pm 1 \). \( \alpha \) channels model an image by decomposing an image into a superposition of foreground and background information as

\[ i = \alpha F + (1 - \alpha)B \quad (2.11) \]

When this equation is combined with Equation 2.10, a similar equation arises, given by Equation 2.12

\[ \nabla \alpha_b|_p \cdot b = \alpha_s(p + \frac{b}{2}) - \alpha_s(p - \frac{b}{2}) \quad (2.12) \]

where the subscripts \( b \) and \( s \) again refer to the blurred and sharp image. Since the left-hand side of the equation has been constrained, another degree of freedom has been removed. Finally, an algorithm estimates the PSF of the image. The step by step for determination of the space-invariant vector, \( b \), is given by the authors as

1. Obtain \( \alpha_b \) by spectral matting [21] and obtain \( \nabla \alpha_b \)

2. Check consistency of pixels by reducing noisy components (outliers and incorrect matting components)
3. Iterate:

(a) Randomly select two pixels and assign values for $\alpha_b \cdot b = \pm 1$

(b) Solve least squares system

(c) Obtain number of inlier pixels, $n$, i.e. error based on random assignments is less than some threshold

(d) If $n > n_{max}$ then $n_{max} = n$, $b_{optimum} = b$ and update $\nabla \alpha_b$

The developed framework is also applied to space-variant motion. The types of motion considered include affine motion and rotational motion. A nonparametric MFB method is explored to estimate these motion types, but is beyond the scope of this review.

2.3 Deblurring Methods

2.3.1 Classical Methods - Richardson-Lucy and Wiener Filtering

Classical deblurring methods have been significantly modified, but are foundational for many deblurring algorithms. These methods presume the PSF has been obtained by some other means. Some particular methods that continue to be important are the Richardson-Lucy (RL) algorithm and Wiener filtering. It is important to understand the basics of these methods before more developed methods are considered. The Richardson-Lucy (RL) algorithm was introduced in [34, 25] separately. Fundamentally, a blurred image can be decomposed into likelihood functions. Probabilistic decomposition is possible since there is a certain probability that $i_s$ resulted from a given PSF and $i_b$, which are known. To solve for the crisp image, a guess is made for the crisp image, entered into the known image blurring model, and tested. Since the method is iterative, it can avoid many of the problems associated with direct inversion of the known PSF. Simple Fourier methods have failed to produce recognizable images due to noise amplification. Again, the image formation model is considered
as in Equation 2.1. The motivation for the RL algorithm immediately brings Bayes equation to mind, given in Equation 2.13,

\[
P(i_s|i_b) = \frac{P(i_b|i_s)P(i_s)}{P(i_b)}
\]  

(2.13)

where \(i_s\) and \(i_b\) are the sharp and blurred image respectively, one can determine likelihoods and iterate to solve for the sharp image. Based on the relationship between the original image and the PSF, an initial system test could be as simple as frequency domain division.

Wiener filtering [46] is the optimal filter for a noisy signal and used in image deconvolution [14]. The idea is not image specific in the least, but elegantly applies to this domain. Equation 2.2 is repeated here for reference.

\[
I_b = HI_s + N
\]  

(2.14)

Since simple inversion will amplify noise and create an unrecognizable image, a more elegant method must be determined to invert the system. Instead of simply inverting the known PSF, an optimal filter can be determined so that Equation 2.15 is satisfied.

\[
I_s = G_{opt}I_b
\]  

(2.15)

As shown previously, in the presence of noise, \(\frac{1}{H}\) will not suffice. Instead, one must design this \(G_{opt}\) to minimize the expression [4]

\[
E\left(||I_s - G_{opt}I_b||^2\right)
\]  

(2.16)

By assuming that \(I_b\) and the noise, \(N\), are independent, random processes, we obtain the optimum filter as

\[
G_{opt} = \frac{H}{|H|^2 + \frac{|N|^2}{|I_s|^2}}
\]  

(2.17)

For a noiseless system, the optimum filter is simply the inverted PSF. Assuming
the PSF has been determined through other means, and that the noise power can be characterized by the hardware or advanced image analysis techniques, one can obtain a crisp image with minimum noise amplification without an iterative approach.

2.3.2 Homography Model

While the traditional convolution model has allowed the field of motion estimation and deblurring grow initially, many problems have arisen due to this simplistic model. For instance, the model usually assumes that the PSF of the scene is spatially invariant, meaning, for camera motion relative to the scene, each point is blurred by the same PSF. Another model considers a blurred image the combination of the sharp image transformed by homographies in some fashion. Tai et al. [40] consider this model. They modeled the blurred image as the integration of the clear scene followed by homographies. The homographies describe the motion path in a point by point manner which is more effective at modeling spatially varying motion blur. Additional work was also done to modify the RL algorithm to incorporate this new model. Their model included only minimal assumptions. First, they were only concerned with blur caused by camera movement; a static scene was assumed and lens aberration were not considered. The model is also unique in that it has no apparent frequency domain equivalent. It was also assumed that different methods were used to estimate the camera motion (such as hardware methods or rigid priors). Spatial invariant representation of motion coupled with the convolution model is a special case of the homography transformation approach. This special case assumes each homography is a translation. However, since homographies can account for different types of motion, and not necessarily restrictive to the whole image, this is an important step in generalizing motion capture. As per the paper by Shepp and Vardi [39], the derivation for RL deblurring is shown as a maximum likelihood solution which can be solved using numerical methods. The iterative update rule for the RL algorithm can now be transformed to take into account the homography motion model and outfitted with regularization to reduce deconvolution artifacts.
2.3.3 Frequency Shaping - Flutter Shutter

Raskar et al. introduce a novel technique to avoid noise amplification due lost frequency information. The algorithm extends to linear, first order motion for even large movements. In order to preserve frequency information, the shutter is opened and closed using pseudo-random binary sequences. Normally, a box filter (i.e. a normal exposure) destroys important high-frequency spatial details. Changing the frequency response of the camera exposure to preserve high-frequency information makes the need to iterate or use an optimal filter to reduce noise irrelevant. A plot of a normal exposure versus a pseudorandom coded exposure can be seen in Figure 2-3. This process makes image restoration a well posed problem so a simple linear least squares approach recovers the sharp image. A typical $Ax = b$ linear system is created where $A$ is a circulant blur matrix which represents the discrete convolution of the image. This matrix is created using the known code and the original image is directly solved for. A sequence of $M$ 1,0 chops are used to capture the image. For instance, a signal of $M = 4$ chops could be 1,0,1,1, where the total exposure time is $T$ and the length of each chop is $T/M$. If an object blurs by $k$ pixels over an exposure, the object blurs by only $k/M$ pixels over a single chop. The frequency spectrum for a new pulsed exposure can be computed by zero padding the code. Due to varying length pulse sizes over the exposure, frequency nulls present due to the sinc function are canceled with a suitable code. An optimal code was determined by brute force search. Code fitness was determined by the smallest condition number of the coding matrix, and minimizing the variance of the frequency spectrum of the code. A flat frequency spectrum code, such as a modified uniformly redundant array (MURA) code, seems ideal for this application, but MURA codes are non-ideal for zero padded sequences. This observation resulted in the brute force search method. Additionally, this method can gracefully handle a large degree of motion blur, which is often a problem for many algorithmic or hardware approaches. Again, the method was designed to avoid noise amplification which is a huge benefit over previous approaches. Also, there are little sampling/quantization problems. It is shown that deblurring can still occur in the
Figure 2-3: A pseudorandom sequence of 1s and 0s (red) preserves more spatial frequencies than a normal exposure (blue). Due to this fact, simple PSF inversion occurs even in the presence of noise.

In the case of occlusions, as long as these occlusions are smaller than the actual blur length. Even edge background can be recovered. In many approaches, edge background can cause ringing artifacts near the edges of deblurred objects. Though, to avoid artifacts, the method is limited to constant radiance objects, as in the optical flow constraint. For well-lit, spatially invariant motion, this assumption is fairly safe.

While the flutter shutter approach maintains broadband information over a range of spatial frequencies, there is a fundamental limit as to what spatial frequencies may be preserved. This limit is explored in depth by McCloskey [26]. A given flutter shutter sequence will still produce nulls at a particular spatial frequency. If a spatial frequency of interest is high enough and produces a null, the modulation transfer function cannot be simply inverted. Though the details of the fluttering pattern play some part in determining the modulation transfer function, the effective PSF also depends on the object motion. The flutter shutter pattern actually generates a family of PSFs where the family is determined by the object velocity. If a coded exposure is made of $N$ chops, it is simple to show that any object that moves $2N$ pixels will have a zero in its spatial frequency transform. It can additionally be shown that any object that moves more than $2N$ pixels will contain at least one zero in its spatial frequency transform. The author adds additional terms to the optimal
code finding process in order to create a velocity dependent shutter sequence, though
the analysis still assumes the speed is known a priori. Object constraints, such as
posted speed limits, could serve to set known bounds to calculate velocity dependent
shutter sequences. Terms in the shutter finding metric are as follows: a term to limit
readout noise, mean contrast, minimum contrast, and frequency magnitude variance.
Codes were experimentally tested to determine the best code. The velocity dependent
method shows significant improvement over the standard flutter shutter approach by
reducing blur and ringing.

2.4 Blind Deconvolution Methods

2.4.1 Brief Overview

Blind deconvolution methods provide for the complete restoration of a blurred image
from the data captured. A possible pipeline for deblurring images is shown in Figure
2-4. Iterative or direct estimation strategies can be used. Methods for determin-
ing PSF information and deconvolving the image vary from single-image captured
methods, to multiple image capture methods, to diverse hardware approaches. The
trade offs between many of the approaches are often very similar and related to scene
assumptions. These scene assumptions may be similar to the optical flow constraint
or may be more specific such as first-order, linear motion. Levin et al [23] provide
an overview of many types of blind deconvolution methods. They point particular
attention to the fact that the convolution model creates an ill-posed system in the
first place though the model is commonly used in much of the literature. Also im-
portant to note is that despite exhaustive research, real results are rarely produced.
Often times, a theoretical approach is only used on manually blurred data, whether
from synthetically applied blur kernels or from manually combining a series of images
into a single image. Important progress was made in terms of natural image statistics
[27, 20, 5], which offer an opportunity to know the approximate spectrum of the sharp
image. Luckily, when considering the ill-posed model, usually the size of the blur ker-
Figure 2-4: An initial PSF metric can serve to directly estimate a deblurred image or serve as a seed for an iterative pipeline (shown by dotted blocks). Based on an error metrics, depending on the technique, iterations can further reduce deblurring errors and estimate complex PSFs accurately.

Kernel is relatively small which is the reason blind deconvolution is possible. However, a problem in many papers is the proper choice of an estimator, not the image capture priors. It is pointed out that this is the failure of the maximum-a-posteriori (MAP) approach which usually assumes a sparse gradient distribution to encompass natural images. This approach ends up favoring the no blur solution, as in, the captured blurry image is exactly the image that was attempted to be captured. In fact, a variational Bayes approach, similar to RL, outperforms all existing algorithms on real-world images. Many times, this can be attributed to the fact that it is assumed that the PSF is shift-invariant, meaning, the local PSF is the same across the image, or image window. For large enough windows, however, this assumption is incorrect and can be easily shown by real-world captured images. Even camera shake includes in plane rotation that causes a number of common blind-deconvolution algorithms to fail. With all this said, however, many new approaches have been developed since the time of this paper and should be given due credit for their application to even real-world images.
2.4.2 Blind Deconvolution with Frequency Spectrum Methods

A direct method for deconvolving motion blur would be to solve for the PSF and directly deconvolve the image rather than iterating between estimation of the PSF and crisp image. Yitzhaky et al. avoid the use of iteration and the use of hardware additions in order to restore motion blurred images [48]. Relative motion between the camera and scene with a spatial invariant PSF is considered. Thus, motion is considered to be linear and space invariant. The convolution model, in this case, is made slightly more explicit by considering the convolution a digital convolution of information. The optical transfer function (OTF), i.e. the frequency transform of the PSF, is broken into a phase and magnitude component as is typical for analysis in many system analyses as given in Equation 2.18.

\[ OTF = H = |H|e^{j\theta} \]

(2.18)

This equation is made explicit since the method will solve for $|H|$ and derive the phase information from the magnitude, since, for causal blurs, the two are linked. Like other frequency domain methods, motion direction can be computed. The method employed in this case is slightly different to other methods. The angle is detected by determining where power spectrum of the image derivative is the least. Since blurring occurs in the direction of motion, and is similar to a low pass filter, and the derivative is a high pass filter. Therefore, applying a high pass filter on information already low passed filtered results in minimum energy. The method then used to recover the PSF is rather unique. Based on the assumptions included in this paper (and in many), the PSF is only along one direction. The PSF is then not correlated to image information perpendicular to the motion direction. A whitening filter perpendicular to the direction of motion suffices to suppress a majority of the image information. Additionally, a whitening filter in the motion direction will form patterns similar to the PSF derivative. Using filters in both directions will result in decorrelated regions as a result of remove image information and PSF derivative patterns. By averaging the
<table>
<thead>
<tr>
<th>Cause</th>
<th>$h(x,y)$</th>
<th>$H(f) = \sqrt{f_x^2 + f_y^2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Out of focus, circular aperture</td>
<td>$\frac{1}{\pi R^2}$</td>
<td>$J_1(2\pi fR)$</td>
</tr>
<tr>
<td>Uniform velocity, constant exposure</td>
<td>$\frac{1}{d}$</td>
<td>$\frac{\sin(\pi df)}{\pi df}$</td>
</tr>
<tr>
<td>Simple harmonic vibration</td>
<td>$\frac{1}{\pi \sqrt{2\pi \sqrt{1-r^2}}}$</td>
<td>$J_0(2\pi Af)$</td>
</tr>
</tbody>
</table>

Table 2.2: Note, the out of focus aperture Equation includes no motion, $J_n(x)$ is the $n^{th}$ order Bessel function, $R$ is the focus defect extent, $d$ is the number of pixels in the motion blur, $A$ is the amplitude of harmonic vibration, and $\text{rect}(dx)$ is the rectangular function, centered at zero, of width $\frac{1}{d}$.

autocorrelation of each line in the motion direction, additional correlated data from the original blurred image and noise from the whitening process will be hopefully minimized and the autocorrelation of the PSF derivative will remain. Fourier transforming the autocorrelation function yields the power-spectrum of the PSF derivative which can be directly manipulated into the OTF. The method is applied to accelerated motion as well. The estimation of the PSF yields noisy results however since iteration is not provided and since frequency nulls will be present in the power spectrum due to the capture method. Still, this is one of the more direct approaches to blind deconvolution.

Frequency domain analysis methods have also been applied to blind motion deblurring in other means. As in, frequency domain inspection yields precise motion information and that information is then used for deblurring in a single pipeline [8, 15]. These techniques universally employ the simple convolution model for a blurred image. Cannon [8] assumes linear camera motion and an out of focus lens system. The power cepstrum of the image provides sufficient information to deconvolve the motion blur information. The analysis considers motion blur and lens blur similarly since both provide analyzable power spectrum information. The blur PSF, given by $h(x,y)$ in the spatial domain can be analyzed, as typical. Table 2.2, from [8] and [15] contains the frequency transforms of a few typical transfer functions.

The examination of the zero crossings of $H(f_x, f_y)$ which will determine $d$ and $r$ where $d$ is the amount of motion blur and $r$ is the amount of radial blur. While lens defocus is not as clear to define in the frequency domain as motion blur, it still
provides some information. Still, the basic power spectrum analysis and cepstrum approach are usually impossible except in synthetic cases. In this particular case, the effects of image noise are mitigated by windowing the image and iterating over these windows which include the overall PSF. For lens blur and a static scene, most of the image can be assumed to have nearly identical PSF. This approach can also be optimized for deblurring the image by constraining areas dominated by noise.

The frequency domain analysis and deblurring of additional types of motion were explored in detail by Tan et al [42]. Additional techniques were presented to analyze zero crossings, determine special image features, and inclusion of a maximum likelihood framework for symmetric and asymmetric frequency spectrum information. Another approach to determine motion type and to use that information for deblurring is also presented. The type of motion can be determined by deconvolving the image using an assumed ramp function, from the typical three fundamental motion types describes by Table 2.1. The function to detect the three motion types does not necessarily have to be a ramp, but any asymmetric PSF will maintain adequate information for motion type detection. One of the problems with the traditional cepstrum approach is that it must be assumed that sharp zero crossings are present in the frequency domain. However, when sharp features are not present, the frequency domain will not offer this information. The paper then takes a unique step. It considers previous literature in which ringing artifacts were analyzed for structure. Essentially, when there were errors in exact PSF size or type, the resulting deblurred image contains some information depending on the type of PSF that was assumed [47, 43, 19].

The ringing and ghosting present in a deblurred image is dependent on the solved size of the blur and the type of blur (e.g. constant velocity or acceleration). Depending on the type of PSF that was assumed, the resulting deblurred image shows information about the true PSF. Interestingly, not much information is preserved for the traditional assumption of constant velocity motion. That is, if the motion was actually accelerated motion and it was assumed to be constant velocity, the resulting deblurred image does not offer much of a pattern to detect the actual type of motion.
However, if the true PSF is assumed to be generated from a ramp, the actual motion that generated the image can be easily detected from the erroneous deblurred image. Each of the three typical motion types are shown to be very different, qualitatively, when deblurred with an assumed ramp. Since a ramp is an asymmetric PSF, the mistakenly deblurred result provides information as to the correct type of motion. Assuming a ramp function offsets image ghosting enough to accurately give blur extent and give the actual PSF shape. However, a square pulse is not suitable since the deblurred image primarily contains information related to the assumed PSF, even if that PSF was incorrect.

2.4.3 Single Photo, Iterative Techniques

Image deblurring from a single image is the desired method for blind deconvolution, but blurred image deconvolution is an ill-posed problem. A uniform probabilistic model for restoration and kernel estimation allows for an iterative approach to solve for the blur kernel and sharp image [37]. Though other iterative methods have been attempted, this algorithm enforces a local smoothness prior to reduce ringing artifacts in the deblurred image. Again, a spatial invariant PSF is assumed so the method in this state can only apply to camera shake blurs. Three important observations allow this method to achieve improved results. First, local smoothness is assumed in low contrast areas which reduces deblurred ringing. Also, using frequency domain optimization, the method is made efficient. Finally, rather than assuming spatially invariant image formation noise, which results noise highly correlated to the sharp image itself, a spatially variant model is employed which allows more precise error analysis. Essentially, in traditional image formation models, the noise distributions create correlations at strong edges of the crisp image which does not allow for separation of the image noise and crisp image. This new model allows for noise/image separation. Ringing artifacts in image deblurring where thought to be a result of Gibb's phenomenon [49], but, for spatial frequencies of interest, few Fourier bases allow for representation of enough information to make the effects of Gibb's minimal. This particular paper finds that the inaccurate model of image noise and PSF er-
rors actually result in ringing artifacts. Similar to the RL algorithm, Bayes is used to model image formation likelihoods. The image noise here is modeled as independently and identically distributed (iid) per pixel, and several orders of the noise's derivatives are constrained to provide for spatially varying noise. The blur kernel is originally modeled as an exponential distribution since camera shake usually results in a sparse kernel. Using natural image statistics from [35, 45], a simple piecewise fitting for the image statistics is created. Again, a local prior is used to constrain the image gradient which also serves to suppress ringing. Smoothness is enforced in local windows of the same size as the blur kernel. Finally, each piece is in place to create an iterative algorithm. First, the blur kernel is fixed and the image is estimated. Then the image is fixed and the blur kernel is estimated. The algorithm completes when the changes between the derived error functions are small or maximum iterations have been reached.

2.4.4 Multi-frame Approaches

Multi-frame deblurring approaches attempt solve image deconvolution problems through additional information capture. Many approaches use images taken in different manners to fill-in information not present in a single, long-exposure, blurred image. For instance, one method takes a high-noise, short exposure image to captured high-frequency spatial details, and a longer exposure to characterize the image motion [49]. However, even a multi-frame approach is challenging due to the need for precise alignment between images. Since the motion blur kernel is sparse in certain domains, multiple images can characterize the kernel and restore the full image. One multi-frame approach employs this method to provide a method that is robust to image formation noise and alignment errors [7]. The motion blur kernel is expressed as a smooth function which varies along a continuous curve and can be estimated using a simple digital camera. This approach assumes some method exists during pre-processing to align blurred images. While an accurate alignment algorithm does not fully exist, existing approaches can minimize alignment noise for small perturbations. Since statistics exist for the sparsity of blur and the image, a metric can be made to
measure the soundness of a blur kernel which employs a smoothness regularization. Again, the simple convolution model is considered, which assumes the image contains a spatially invariant PSF. Given $M$ blurred images, a step-by-step method estimates each blur kernel through iteration. First, an initial guess, $g(0)$ allows for estimation of particular blur kernels. Then using the estimated blur kernels, the following images, $g(k + 1)$, are estimated. By tracking variance between steps, a steady state solution is achieved. Particularly, least-squares deblurring is initially used to provide rough registration and $l1$ minimization is used in the final steps to achieve accuracy. The particular metrics determine blur kernel soundness in each step.

2.4.5 Hardware Modifications

The fundamental trade off between spatial and temporal resolution was used to construct a hybrid camera by Ben-Ezra and Nayar [3, 2]. This hardware approach used two camera sensors to produce a hybrid camera that can measure it’s own motion in order to calculate the path of the camera during the exposure. Ideas were considered on how to extend the hybrid camera approach to individual object motions, but camera path motion blurring was the primary focus. Other hardware approaches, such as Canon’s commercially available cameras [9] were only suitable for small exposures. Or, using specialized CMOS sensors, image integration was stopped in areas where motion was detected [24]. Using two sensors, a primary sensor that is high resolution and a secondary sensor that is low-cost, low-resolution, yet high-speed, motion can be both tracked and deblurred. Since short exposures may produce low signal to noise for high-resolution (i.e. small pixel and small integration areas) cameras, by using low resolution cameras (i.e. large pixels and large integration areas) motion can be captured. This observation can be explored further based on physical sensor characteristics and yields the fact that exposure time is proportional to spatial resolution. Three different design possibilities are presented with their trade offs. The tree designs are: primary camera with a secondary attached rig, an assymetric mirror setup, and on chip division with pixel clustering. In the real-world experiments, the rig setup was used with commercially available hardware. The secondary camera
was a camcorder enforced to be low-resolution to simulate the proposed setup. The secondary camera then has short, but non-zero, integration time. Assumed computed motion between frames was calculated as the center of gravity of these nearly instantaneous displacements. By assuming motion between the two frames as translational only, as in Equation 2.19, and minimizing an error function, a continuous PSF can be estimated.

\[
\begin{bmatrix}
  u \\
  v
\end{bmatrix} =
\begin{bmatrix}
  \cos \theta & \sin \theta & \Delta x \\
  -\sin \theta & \cos \theta & \Delta y
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}
\] (2.19)

The PSF is fitted to a spline in order to make it twice differentiable which meets physical constraints of known scenes. Constant radiance is also assumed to provide for reconstruction. The approach still suffers from a number of fundamental deblurring flaws, such as the fact that background is integrated in the image PSF, and results in information that cannot be removed using current methods.

A unique hardware solution for blind motion deblurring was introduced by Levin et al. and is known as Motion-Invariant Photography [22]. Many times the spatially invariant PSF assumption is not adequate, especially when there are complex objects in the scene moving with different speeds. A possible technique would be to force all blur kernels identical. Sensor movement has been used in commercial cameras [9] to compensate for camera shake, but only works for camera motion and does little to deblur object motion. Other panning approaches have captured multiple images or considered specific panning speeds [31, 30]. It may be possible to pan the camera in such a way that all objects can be deblurred identically. Levin et al. analyze this optimal panning strategy. The resulting system does have a downfall that it only enforces identical PSFs on motion that is parallel to the sensor motion. However, the novelty of such an approach deserves further study. When motion blur is analyzed as integration in a space time volume, it shows that only parabolic motion results in motion-invariant PSF. By parabolic motion, the authors mean a one-dimensional acceleration, fast in one direction, progressively slowing down, then accelerates in
the other direction. Essentially, at one moment the camera is perfectly following a uniform velocity, moving object so its sharp information is captured. The approach assumes sufficiently small exposures so that first order approximation of image motion (constant velocity, linear motion path, and motions can be approximated by a simple shear). The captured image was modeled as a sheared parabola as given in Equation 2.20 where $v$ is the object velocity.

\[
f(t) = a_0 t^2
\]

\[
f_{sheared}(t) = f(t) - vt = a_0 \left( t - \frac{v}{2a_0} \right)^2 - \frac{v^2}{4a_0}
\]  

(2.20)

All projections of the sheared parabola, denoted by Levin et al. as $\phi(s)$, are identical up to a spatial shift. The spatial shift merely corresponds to the position of moving objects at different time instants. One can also show that, for a given range of velocities, this parabolic integration technique maintains near optimal spectrum information. Despite the fact that all moving objects are blurred with near identical PSFs there are still some drawbacks to this approach. First, high contrast textures cannot be deblurred at motion layer boundaries. Also, the shear invariant approximation only extends to a theoretical infinite integration time. However, for careful choice of integration time, the variance in PSF can usually be ignored. Finally, a moving sensor preserves the PSF of moving objects nearly optimally, but degrades static background information after deblurring. As previously noted, this technique can only be applied for motion parallel to the sensor movement. In an attempt to extend this method to arbitrary motions, two sensor movements along orthogonal bases could also be employed [10]. The above motion invariant framework was extended so that motion could be deconvolved along orthogonal bases (i.e. vertical and horizontal motion), but motion had to be constant between the two captured frames in order to completely remove motion blur.
2.4.6 Blind Deconvolution with Coded Exposure

While the flutter shutter approach from Raskar et al. proposed a near-ideal way to deblur images, it did not provide a method for PSF estimation. Agrawal and Xu address this issue by using the MFB algorithm and the flutter shutter approach [1].

The authors first prove that a coded exposure camera also meets the same constraints proposed by the MFB algorithm. As in, it is shown that $\nabla \alpha \cdot k = \pm \frac{n}{s}$ where $k$ is the image blur, $n$ is the number of chops where the shutter is open, and $s$ is the total number of chops. The paper goes on to apply the algorithm to coded exposure imaging by following the MFB alpha matting approach. Motion is separated into orthogonal movement directions and accurately estimated. It is also shown how an optimal code can be found that provides for accurate PSF estimation and near-ideal deblurring. A code cannot optimize for both constraints at the same time. Certain trade offs exist.

Since the MFB algorithm requires a smooth region, and a smooth region results in a string of on chops, frequency spectrum shaping is limited. As the longest strings of ones increase (i.e. there is more local smoothness so alpha matting can perform better), the code loses invertability but gains power in PSF estimation. Essentially, the same trade offs as other alpha matting algorithms exist in PSF estimation. Also, the trade offs exist in PSF estimation in other areas such as the effect of background and the necessity of manual brush strokes for matting initialization.

Another paper extends the coded exposure work by considering projective motion as it’s applied to coded exposure [41]. The usual spatial invariant assumption is not used in this case since in general it cannot be applied to an arbitrary scene. Thus, the PSF can now vary anywhere in the image. Though the method makes important approaches towards considering spatially variant PSFs, the method is not completely automatic. Some user input is required. As the method which only computes the image PSF, blur is considered to be a series of homogeneous coordinate transformations. The model can represent spatially variant and invariant motion without explicit PSF at each pixel location. Spatially variant motion is also consistent with the MFB constraint. The user input in the framework does not need to be accurate and merely
identifies motion paths of feature points. Otherwise, deblurred results closely resemble images which are deblurred from the ground truth PSF for synthetic blurs. In real data, fine details are often preserved to some degree, but ringing and errors present from matting errors and incorrect foreground color estimation still exist.

The PSF estimation technique considered here is meant as an introduction to a new type of coded exposure. Signed codes, when properly chosen, can capture high spatial frequencies as unsigned coding strategies. However, signed codes have a number of benefits that make the estimation of the PSF a simpler problem. Essentially, since extremely low spatial frequencies (static backgrounds, slow movements, etc.) are attenuated, more of the motion information that the user cares about is accentuated. Some of these benefits will be presented in the following sections to spur future research on more creative coding schemes.
Chapter 3

Benefits of Signed Coding

As seen in Figure 1.3, signed codes and unsigned codes both preserve high spatial frequencies for motion deblurring. However, signed codes provide additional benefits. Some of these benefits will be discussed in the following sections.

3.1 Information Capture

Figure 3-1: Signed codes preserve more information of $l(t)$ due to the higher sampling rate of signed codes. The light intensity over time, $l(t)$, is integrated over the chop duration $T$.

Binary coded exposure cameras can estimate PSFs [1] and provide near optimal...
deblurring [29]. A simple coded exposure camera model offers insight into the additional benefits of signed codes. The light falling onto the sensor is modulated with a code having weights \([a_1, a_2, \ldots, a_N]\) where \(N\) is the number of chops in the code and each chop has a duration of time \(T\). The equation for the light falling on a pixel is then given by Equation 3.1 The incoming light, represented as a time dependent signal \(l(t)\), is integrated over a length of time \(T\) for each chop. The total signal, \(p(x, y)\) at a particular pixel at position \((x, y)\) from a coded exposure is

\[
p(x, y) = \sum_{i=0}^{N} a_i \int_{t_0+(i-1)T}^{t_0+iT} l(t)dt.
\]  

Whenever an \(a_i = 0\), signal is lost. A coded exposure that is half open and half closed loses half of the light level and also half dynamic signal, \(l(t)\). Another way to think of this point is that the sample rate of the signed code is twice as fast. A simple comparison illustrates the point. Two coded images, one modulated with a \(\{1, 0\}\) code and one modulated with a \(\{1, -1\}\) cannot be directly related. Even though a \(\{1, -1\}\) code can be represented by a scaled and offset \(\{1, 0\}\) code, the signed coded image has integrated \(l(t)\) during periods in which the unsigned code was receiving no information. A graphical representation of this benefit can be seen in Figure 3-1.

### 3.2 Background Subtraction

An additional benefit of signed codes is automatic background subtraction for velocity segmentation. Many PSF determination and deblurring algorithms suffer due to the presence of high-intensity, high-frequency backgrounds. Suppose an object is only present in a pixels field of view for a single code cycle. Then, over the course of an exposure, the object only contributes to \(p(x, y)\) over \(T\) seconds. With an unsigned code the background is integrated in this pixel for, maximally, \((N-1)T\) cycles. Unless the object intensity is many times greater than the background intensity, the object signal may be overwhelmed. A signed code also integrates over \((N - 1)T\) cycles. However, for a code with an equal number of 1s and -1s, the background is only
Figure 3-2: The incoming light to a pixel over time, \( l(t) \), is dominated by background, but signed codes can attenuate low spatial frequencies. For the width of a chop, \( T \), an object is present in the field of view of the pixel. Shown in blue, the integrated light signal over time is dominated by the integration of background, shown in red, for unsigned codes.

integrated over \( T \) cycles. For an object intensity comparable to background intensity, a significant amount of object signal is integrated. A graphical view of this benefit can be seen in Figure 3-2. For a code with an equal number of 1s and -1s, background not integrated in the motion path is completely removed. Velocity segmentation becomes as simple as grouping non-zero image information.

### 3.3 Width and Velocity Determination

Determining the precise width and velocity of an object using generic codes will now be considered. In the case of uniform velocity and spatially invariant motion, the velocity completely determines the objects PSF. Our approach is to show that, using a signed code, accurate differentiation can be made between two different velocities. That is, the ability to differentiate between two different velocities for two objects, one moving at \( v_1 \) and the other at \( v_2 \), for \( v_1 = v_2 + \Delta v \). First, a single blurred object will be examined. The photo is captured using a coding pattern \( c(t) \) with magnitudes \( [a_1, a_2, \ldots, a_N] \) where \( N \) is the number of chops of length \( T \) in the code. This code can be represented as the equation \( \text{rect}(t/T) * \sum_{i=1}^{N} a_i \delta(t - iT) \), where \( \text{rect}(t/T) \) is the rectangular function. The spatial blur of an impulse response with
unit magnitude, moving at a velocity \(v_1\) is a scaled version of this code, \(c(x) = \text{rect}\left(\frac{x}{v_1T}\right) \ast \sum_{i=1}^{N} a_i \delta(x - iv_1T)\). The object, \(o(x)\) can vary spatially, so the spatial blur, \(b(x)\), becomes

\[
b(x) = o(x) \ast \text{rect}\left(\frac{x}{v_1T}\right) \ast \sum_{i=1}^{N} a_i \delta(x - iv_1T) \tag{3.2}
\]

Correlation is used to determine similarities between different signals. In order to compare two different blur patterns, the correlation of their spatial blurs will be considered. If two spatial blurs are identical, we expect a peak at a single value in the correlation. A strong peak, at any point in the correlation, suggests some similarity. For instance, objects traveling at velocities close to each other may result in a peak as well. Hopefully, this peak can be minimized. Correlation can be considered to be space reversed convolution, i.e. \(\text{corr}(b_1(x), b_2(x)) = b_1(x) \ast b_2(-x)\). Since convolution is commutative, this expression can be rearranged. The primary differences between two blur patterns are the velocity of the motion and the texture and shape of the object. The correlation between two objects, \(o_1\) and \(o_2\) traveling at two different velocities, \(v_1\) and \(v_2\), results in Equation 3.3

\[
o_1(x) \ast o_2(-x) \ast
\text{rect}\left(\frac{x}{v_1T}\right) \ast \text{rect}\left(\frac{-x}{v_2T}\right) \ast
\sum_{i=1}^{N} a_i \delta(x - iv_1T) \ast \sum_{i=1}^{N} a_i \delta(-(x - iv_2T)) \tag{3.3}
\]

When object widths and velocities are slightly different, two spatial blurs will closely resemble each other. Consider the case in which \(v_2 = v_1 + \Delta v_1\). If \(\Delta v_1 = 0\), then the convolution of the two impulse trains overlap, and the maximum peak is the sum of code coefficients squared. If \(\Delta v_1\) is nonzero, a train of impulses results with peaks separated by \(\Delta v_1T\). The correlation of the spatial pulses overlap. Depending on \(\Delta v_1\), the location of the sharp peak changes. However, if some of the \(a_i\) coefficients have signed values, the correlation peaks can attenuate each other. If the peak in the
Figure 3-3: Correlating two blurred signals results in pulse mixing, which allows for correlation peak minimization if signed codes are used. Here, a graphical view of the correlation of two spatially scaled blurs. Correlation of the two impulse trains results in a mixed impulse train and convolution of the two box functions (exposure chops) results in a trapezoidal structure if \( \frac{v_1}{v_2} \neq \frac{1}{2} \). The spreading of the peaks of the correlation results in overlap between the combined functions which provides for a possible reduction in peak correlation value by careful code choice.

correlation is attenuated, then slight differences in \( \Delta v_1 T \) will result in a signal that appears different than the autocorrelation. Figure 3-3 considers a toy example of a coding scheme where \( \Delta v = 0.25v_1 \) so that \( v_2 = 1.25v_1 \). For small velocity or width differences, the main peak of the correlation will still remain about half of the shift, where the square terms are close together. The difference between \( \{1,-1\} \) and \( \{1,0\} \) codes can be seen clearly in Figure 3-4. The strong peak can be attenuated by using signed codes in adjacent terms in this example.

3.4 Optimal Code for Motion Discrimination

Ideally, an optimal code would be able to differentiate between velocities arbitrarily well. That means if two objects are moving at \( v_1 \) and \( v_2 \) and \( v_2 = v_1 + \Delta v \) then the maximum of the correlation between these two blurs would be small for \( \lim_{\Delta v \to 0} \).
Figure 3-4: A method for comparing two random textured objects with slightly different widths and velocities shows how peaks in the signed coded correlation diminish. The object depicted by the red curve travels at velocity $v_2 > v_1$. The cross correlation of these different blur patterns is diminished using $\{1, -1\}$ codes since the high peak present in the auto-correlation is attenuated by the codes. The coefficients of the mixed impulse train provide the only method to modify the resulting correlation. For two similar velocities and textured object the correlation results in trapezoidal shapes where the width of the top of the trapezoid is equal to the spatial difference. The autocorrelation of one of the blur pattern, $b(x)$, is shown for the unsigned and signed codes on top and bottom respectively.

Minimizing this maximum serves as a differentiation metric, and iterated over all possible length 20 codes. It was determined that the optimal series code for any arbitrary length code is an alternating series of $\{1, -1\}$. To provide intuition for this fact, the frequency spectrum for all codes is examined. This analysis provides the insight that optimal velocity determining codes preserve high-frequencies. Unfortunately, an alternating code would introduce spatial frequency nulls so such a code is non-ideal for deblurring. Likewise, a simple metric was used to determine the optimum deblurring code. The metric in this case was that the optimum code for deblurring would maximize the minimum value of the Fourier transform of the code. In this case as well, the optimum code for deblurring was far from optimum for velocity/width.
determination. Depending on deblurring/estimation needs, a suitable code could be chosen with an appropriate broad band or high-frequency preserving metric.
Chapter 4

Width and Velocity Estimation for Synthetic Scenes

Figure 4-1: The velocity/width estimation pipeline is a simple method to compare signed and unsigned codes for their estimation characteristics. A LUT is generated for widths/speed of interest with arbitrary resolution. A coded exposure image is captured using a known \( \{1, -1\} \) code. The known code is correlated with each of the objects in the LUT and the maximum value of that correlation is saved. Finally, the correct velocity/width of the object is selected by selecting the maximum of the saved values.

A simple pipeline is proposed for estimating object width and motion. The method will then be applied to synthetic and real-world data. This pipeline is shown graphically in Figure 4-1. A LUT is generated for velocities of interest. The LUT is shown in Figure 4-2 for a subset of widths and velocities. Using a specialized detector, a photograph is captured using a known \( \{1, -1\} \) code. The resulting blurred photo is segmented into distinct motion blurs. Since the background is eliminated using a signed code, velocity segmentation requires grouping non-zero elements in the photo. By indexing a particular blur pattern, the blur pattern can then be compared to the simulated blur patterns. Angle of movement must also be estimated in order to
achieve results. The pattern is rotated to be along the same axis given by the orientation of the simulated bars. The spatial blur is then correlated with each element in the LUT and the flipped version of the LUT which determines direction. The peak value of all correlation maximums determines the width and velocity.

Figure 4-2: The LUT is created using images of synthetically blurred bars of varying width and velocity. Width increases for unblurred bar as rows change from top to bottom and velocity increases for the bars as columns change from left to right.
Figure 4-3: Correlation maps between the various blurred bars compared between signed and unsigned codes. Unsigned codes show high correlations even to off-diagonal entries, whereas signed codes maintain differentiation. Each blurred pattern (shown in Figure 4-2) was correlated with every other synthetic blur pattern. Code lengths were chosen to be 20 chops long in this case. The top row of images were generated using widths and blur lengths of 1 to 20 pixels. The bottom row used blur lengths of 21 to 40 pixels. The signed codes maintain excellent correlation between close widths and speeds even when the spatial sampling is over few pixels.
Figure 4-4: An example of determining width and speed for synthetic data with accuracy despite a complex background. The squares have blurred by 10, 20, 30, and 40 pixels from top to bottom respectively and all have a width of 24 pixels. The peak in the corresponding correlation maps has been labeled in red and corresponds closely or perfectly to the manually determined blur and width, labeled in green. Blur patterns with significant integrated background resulted in errors.

The steps of the pipeline for synthetic and real-world objects are as follows:

1. Capture photo with a known \( \{1, -1\} \) code

2. If necessary, generate LUT by synthetically blurring uniform intensity bars of different widths and velocities using the same code

3. Isolate unique motion blurs through simple thresholding

4. For different motion blurs

   (a) Obtain motion direction and rotate motion vector to be horizontal

   (b) For each row in isolated motion blur

      i. Compare blurred row to LUT to obtain width and velocity
Spatially varying motion can also be recovered by this simple pipeline as shown by this synthetic scene. A synthetic object with spatially varying width and velocity. Spatially varying velocity was generated by rotating the object over a small angle over the simulated exposure.

Even for real-world objects, a LUT of uniform intensity objects works in many cases. A real world object texture, \( o(x) \), can be broken into an average value and a texture so that \( o(x) = o_{\text{avg}} + o_{\text{texture}}(x) \). If \( o_{\text{avg}} \gg o_{\text{texture}}(x) \), the object resembles a uniform intensity object. Experimentally, even significant texture results in accurate detection.

A correlation map provides a view of the discrimination between different velocities and widths of objects shown in Figure 4-3. The horizontal and vertical axes correspond to the various generated blur patterns; a peak is expected along the diagonal which corresponds to the autocorrelation. Off-diagonal entries have lower intensities. Moving away from the diagonal, blur patterns correlate less with the reference blur pattern. This trend shows that different velocities and widths do not correlate well with the original. Using synthetic squares moving on a complex background, shown in the top of Figure 4-4, an example of mapping to the blurred pixel LUT has been generated. The sharp peak has been labeled in red and corresponds to the determined speed and width of the object at the row index of interest. This synthetic scene is a discrete example of spatially varying velocity, such as a rotating object. Using synthetic data of a rotating cone, accurate estimations of width and velocity are still made. A plot of this data and a representative frame is shown in Figure 4-5.
Chapter 5

Width and Velocity Estimation for Photographs Captured by a High-Speed Camera

Figure 5-1: A third person view of the simple setup used to capture real-world photographs.
Real world results were captured using a Casio EX-F1 at 300 frames per second as shown in Figure 5-1. 20 frames were then combined into a blurred image using signed or unsigned codes. This technique also allowed for ground truth comparisons. Code lengths were arbitrarily chosen as 20 chops since it allowed for blur lengths that were long. The optimal codes for velocity and motion detection were used for signed and unsigned codes. The optimal code for signed codes is an alternating series of ones and zeros and the optimal code for unsigned codes is a symmetric string of ones and zeros with two zeros in a row in the middle (i.e. \(1, 0, 1, 0, 0, 1, 0, 1\)). Ground truth width was estimated by manually measuring the width of the objects in a single frame. Speed was estimated by comparing left edge indeces between the original and final frame. Scenes were captured by moving an object in front of the camera by hand in a particular direction. Photo sets were captured in front of a black background initially. To show the additional benefit of signed codes, objects were then moved in front of a complex background. Motion and width estimations were determined for rows of pixels individually. That is, a single blurred row was compared to the LUT and a velocity and width were selected. Then the next row of pixels was considered. The velocity and width of the object was determined for each row of the velocity segmented image. A comparison of mean errors in width/speed estimations for objects captured with signed and unsigned codes is shown in the bottom of Figure 5-2. Representative frames from the photo sequences are shown in the top of Figure 5-2.
Figure 5-2: Eight captured datasets where mean errors have been compared. Velocity errors are particularly small for signed codes (a) A single frame from each test sequence. Each object was imaged with a black matte background and with a complex background (b) A comparison between signed and unsigned codes using average error and error variance between ground truth and estimated width. (c) A comparison between signed and unsigned codes using average error and error variance between ground truth and estimated speed.
5.0.1 Analysis of Real-World Results

A comparison of the row-by-row reconstruction of the photo sequences can be seen in Figures 5-3, 5-4, 5-5, and 5-6. Signed and unsigned codes have been compared to show the differences between the two. To decrease analysis time, only blur lengths from 1 to 20 pixels were considered, which produce the most ill-posed cases due to low sampling. The velocity estimation from unsigned codes often saturates at an estimated 20 pixels of blur. In order to perform the correlation task, the velocity segmentation data from the signed coding scheme was used for both the signed coding and unsigned coding schemes. To be more precise, each of the results will be considered in detail. For quantitative measurements of the average error in each experiment, Figure 5-2 should be referenced.

5.0.2 Uniform Intensity Object

In Figure 5-3 a Styrofoam cup was moved in front of a black or complex background. The simple black background case will be considered first. The Styrofoam cup is the simplest of the objects since it is uniform intensity and a simple shape. The width estimation has some problems at the lip of the cup where the shadow (i.e. dark colors) seem to fool the correlation method into thinking that there is no object present at this row of pixels. The unsigned codes decently follow the shape of the cup, but do not estimate the width correctly in each pixel row. However, the velocity calculation saturates for every row. Spatially scaled blurs, therefore, are not able to be distinguished accurately. Signed codes accurately estimate speed and more accurately estimate the width, though have some problems with the shadows at the lip of the cup. This could be solved using image priors or more complex estimation methods. Considering the simple case of the Styrofoam cup captured on a complex background, the unsigned code again saturates when estimating velocity. Again, this means that unsigned codes cannot distinguish well between spatially scaled signals. Width also causes some problems as well, though does capture the change in width at the lip of the cup. The signed code clearly is much more accurate at predicting
the spatial blur of the cup. The width also nicely tracks the ground truth data. With a complex background, the width estimation actually seems to perform better which may be due to the presence of textures and colors to define sharper edges.

Figure 5-3: The row-by-row reconstruction of a Styrofoam cup moving on two different backgrounds results in fair accuracy from unsigned codes and high accuracy from signed codes. (a) and (b) show representative photos from the cup moving on a black and complex background respectively. For (c) through (f), width reconstruction is on the left and shown in white, and speed reconstruction is on the right and shown in red. Ground truth estimated from manual analysis of the frames in a video sequence and is shown in green. (c) shows the width and speed reconstruction of the cup using an applied \{1, 0\} code when the cup was on a black background. Note, in a number of cases, the speed estimation using the binary code has saturated so an accurate error measurement was not possible. (d) shows the width and speed reconstruction of the cup using an applied \{1, -1\} code when the cup was on a black background. (e) shows the width and speed reconstruction of the cup using an applied \{1, 0\} code when the cup was on a complex background. Note, in a number of cases, the speed estimation using the binary code has saturated so an accurate error measurement was not possible. (f) shows the width and speed reconstruction of the cup using an applied \{1, -1\} code when the cup was on a complex background.
5.0.3 Object with Text

Figure 5-4 shows representative photos and results from velocity and width estimation of a bottle with dark text. In order for the best results, textures on the objects should be low-amplitude which provides the most accurate results. Thus, as with the lip of the bottle, one can expect some width estimation errors in lines with significant text on the bottle. When imaged on a black background with an unsigned code, the bottle width is consistently underestimated, and, as usual, the speed is consistently over-estimated. More interestingly, the speed estimation is dependent on the width perceived by estimation method. It is possible that higher than normal values are perceived due to the integrated background information combined with the motion blur which ambiguates information from spatial scaling. The signed code width estimation has significant errors in width estimation, due to dark (i.e. very low amplitude) text on the object. The velocity estimation is extremely accurate, though suffers slightly when the width of the bottle changes significantly. Similar trends can be seen when the bottle is moved in front of a complex background. The width estimation with an unsigned code is highly accurate, though at the top of the bottle the estimation suffers due to near identical background present. The velocity estimation is, again, highly inaccurate, depends on object width, and saturates at a number of image rows. The sensitivity of spatial scaling and the benefit of automatic background attenuation can be seen nicely using the signed codes. The width estimation is more accurate in this case most likely due to the fact that the photo sequence used a bright light source that increased the grayscale amplitude of the bottle text. The velocity estimation is extremely accurate, even with background, since signed codes are extremely sensitive to spatial scaling and background effects are reduced. In some cases, it appears that background may even aid in velocity detection which needs further analysis.
Figure 5-4: The row-by-row reconstruction of a bottle moving on two different backgrounds results in accurate estimation for signed codes. Unsigned codes estimate the velocity poorly. Unsigned codes result often saturate and depend on object width. (a) and (b) show representative photos from the bottle moving on a black and complex background respectively. For (c) through (f), width reconstruction is on the left and shown in white, and speed reconstruction is on the right and shown in red. Ground truth estimated from manual analysis of the frames in a video sequence and is shown in green. (c) shows the width and speed reconstruction of the bottle using an applied \{1, 0\} code when the bottle was on a black background. Note, in a number of cases, the speed estimation using the binary code has saturated so an accurate error measurement was not possible. (d) shows the width and speed reconstruction of the bottle using an applied \{1, -1\} code when the bottle was on a black background. (e) shows the width and speed reconstruction of the bottle using an applied \{1, 0\} code when the bottle was on a complex background. Note, in a number of cases, the speed estimation using the binary code has saturated so an accurate error measurement was not possible. (f) shows the width and speed reconstruction of the bottle using an applied \{1, -1\} code when the bottle was on a complex background.
5.0.4 Object with Text and Intensity Differences

Similar results follow when the velocity and width of a marker are determined as shown in Figure 5-5. When unsigned codes are used we again see that velocity determination is inaccurate and dependent on estimated width. Specifically, estimation of the hand holding the marker results in accurate velocity, whereas the marker itself results in over estimation of velocity. The signed code has problems estimating the width of the hand, again, most likely due to strong regions of shadow, but the marker width is fairly accurate. The velocity estimation varies quite significantly, but the average velocity approaches the accurate estimation. In the case of the complex background, velocity estimation is again strongly width dependent. The area of the hand should be ignored in this comparison since neither approach distinguishes the width accurately seeing that it would more accurately be represented as two separate objects blurring across the image which was not accounted for in the simple estimation scheme. The signed code again accurately estimates velocity, but significantly errs in width estimation due to poor lightning.
Figure 5-5: The row-by-row reconstruction of a marker moving on two different backgrounds results in an accurate reconstruction in the case of signed coding. (a) and (b) show representative photos from the marker moving on a black and complex background respectively. For (c) through (f), width reconstruction is on the left and shown in white, and speed reconstruction is on the right and shown in red. Ground truth estimated from manual analysis of the frames in a video sequence and is shown in green. (c) shows the width and speed reconstruction of the marker using an applied \{1, 0\} code when the marker was on a black background. (d) shows the width and speed reconstruction of the marker using an applied \{1, -1\} code when the marker was on a black background. (e) shows the width and speed reconstruction of the marker using an applied \{1, 0\} code when the marker was on a complex background. (f) shows the width and speed reconstruction of the marker using an applied \{1, -1\} code when the marker was on a complex background.
5.0.5 High-Frequency Texture

The photo of the vase breaks a number of assumptions in our simple estimation scheme. With added complexity to the estimation pipeline a complex, highly textured image, such as the vase, may be able to be recovered. However, the LUT, created with uniform intensity bars, performs best when grayscale values vary slightly around a given mean. Due to the wild fluctuations in grayscale values, the width estimation will behave poorly. In future experiments, to achieve accurate width estimations, the velocity segmentation image could mask the entire vase and use the masked image to compare to the LUT. Important to see from these results is that the signed coding, despite problems with trying to match the widths of many objects, accurately estimates object velocity since it is sensitive to spatial scaling. Using unsigned codes, the shape of the vase is actually captured quite well with a black background. The velocity, however, saturates. With a black background, the signed codes estimate the width poorly but estimate the important component for deblurring, the velocity, accurately. The presence of background makes unsigned codes fail on width and velocity estimation. The width estimation overestimates to such a degree the velocity estimates the minimum value. However, the signed code fails to accurately estimate the width accurately due to the highly variant texture, but the velocity estimation is still accurate on average again due to spatial scaling sensitivity.
Figure 5-6: The row-by-row reconstruction of a vase moving on two different backgrounds causes problems for the simple pipeline, but signed codes still provide accurate velocity estimation. (a) and (b) show representative photos from the vase moving on a black and complex background respectively. For (c) through (f), width reconstruction is on the left and shown in white, and speed reconstruction is on the right and shown in red. Ground truth estimated from manual analysis of the frames in a video sequence and is shown in green. (c) shows the width and speed reconstruction of the vase using an applied $\{1,0\}$ code when the vase was on a black background. Note, in a number of cases, the speed estimation using the binary code has saturated so an accurate error measurement was not possible. (d) shows the width and speed reconstruction of the vase using an applied $\{1,-1\}$ code when the vase was on a black background. (e) shows the width and speed reconstruction of the vase using an applied $\{1,0\}$ code when the vase was on a complex background. Note, in a number of cases, the width and speed estimation using the binary code has saturated so an accurate error measurement was not possible. (f) shows the width and speed reconstruction of the vase using an applied $\{1,-1\}$ code when the vase was on a complex background. While the $\{1,-1\}$ code has trouble with width estimation in the presence of a simple and complex background due to the high frequency texture of the vase, the overall speed estimation is fairly accurate due to the spatial blur pattern produced by the particular code.
5.0.6 Velocity Estimation and Deblurring Example

As a final example of the benefits of signed coding, a deblurring task is reviewed which can be seen in Figure 5-7. Two batteries were placed on electronically controlled, linear translation stages. The batteries were moving at different speeds. Though background removal allows for easy velocity segmentation, velocity estimation and deblurring was applied to the entire frame. First, the estimation of velocity fails with the unsigned code case, and is accurate in the signed code case. The accurate velocity estimation was used in order to deblur the image. With poor segmentation, the motion window may include static parts of the scene which could introduce artifacts. Despite inaccurate segmentation, the text on the deblurred battery is clearly readable when deblurred from the signed coded image.

5.0.7 Spatially Varying Motion

Also, in order to demonstrate that varying spatial motion can also be captured, results from a rotating ruler is shown in Figure 5-8. The ruler was held at the top of the screen and rotated around this point. Since the bottom of the ruler is further from the pivot point, one would expect the bottom of the ruler to move at a higher velocity if the rotational motion is approximated as linear. The estimation of the width and velocity as the velocity varies spatially closely matches the actual values.
Figure 5-7: An entire deblurred scene when signed codes have been used to estimate velocity and poor windowing has occurred which results in artifacts. Signed codes still maintain accurate reconstruction due to attenuation of background information. (a) A single frame of a series of photos taken of two batteries translating on linear stages. The batteries were moving at different speeds and the velocity of the bottom battery is greater than the top battery. (b) The deblurred image from signed coding with improper windowing. (c) The deblurred image from unsigned coding with improper windowing. Despite errors in windowing, the high frequency information is still preserved in the signed coding case. Also, since the velocity for the bottom battery was estimated, the parts of the scene moving at a different velocity (i.e. the top battery, shadows, etc.) are indiscernable. In order to achieve velocity estimation, data from the signed coded image had to be used since the unsigned estimation was inaccurate.
Figure 5-8: Spatially varying motion of real-world scenes with complex backgrounds can also be accurately estimated. The speed and width estimation for an object with spatially varying velocity generated by rotating the object by hand. Width estimation is shown on the left in white and velocity is shown on the right in red. Ground truth, shown in green, was determined by measuring data points in a single frame for width and between multiple frames for velocity.
Chapter 6

Discussion and Future Work

6.1 Discussion

Signed codes are a new tool that offer a number of benefits in the area of sensor image processing. With state-of-the-art sensors, the user obtains a number of benefits such as easy motion segmentation. With the aid of post-processing, the user obtains precise velocity information due to the benefits of signed coding documented in this paper. In order to preserve a broadband frequency spectrum for deblurring, a non-optimal motion detection code could be used. By preserving information at every time step in an exposure, while still using a time varying exposure, precise object motion is obtained and the idealities of a coded exposure camera are maintained. Coded exposure using signed codes is just one area in which signed codes will begin to offer increased signal processing power with continued work on generalized, signed coding schemes in all types of coded imaging which are being explored.

6.2 Future Work

Generic image coding offers many opportunities. As more versatile hardware begins to develop, such as on chip signal processing, multiple storage elements per pixel, and diverse charge integration, the more general past image coding schemes will have to become. There are number of areas in which a generalized signed code, or even
unsigned codes, can develop for information capture. One such area is in the determination of motion from generalized occluded blurs. In the past, some algorithms have sought to segment out areas in which two motion blurs have occluded each other, but a great deal of information is still present in these areas. A new analysis of motion detection using signed codes could lead to further optimizations for determining motion blurs from motion occlusions and many objects with complex, independent movements. Also, speed filtering using coded exposures and a metric to determine this code could be explored. Temporal filters persist as an important field of research [13]. Since unsigned codes integrate DC information regardless of code type, it is difficult to spread energy to user defined areas in the spectrum. A signed code may have the ability to designate areas were spatial frequencies should be amplified with regards to other spatial frequencies. Essentially, a user could optimize a code to attenuate all frequencies except a particular spatial frequency, which may be useful in speed filtering and could utilize current algorithms for tapped filter design. Additionally, an approach to decompose movement into two orthogonal axes for independent treatment would aid in the efficiency of motion detection algorithms since only angular estimation from the frequency transform, or the motion blur is currently available. Custom built sensors may also allow for continuous exposure readout with an appropriate signed code. That is, readout may never be necessary if the sensor technology is such that event detection can occur using only on-chop electronics. The signed codes would allow for sensor normalization over long exposures. Signed codes, made available through unique cameras, provide new codes to be explored in any number of computer vision analysis tools such as coded aperture codes and image stabilization.
Bibliography


