Robust Real-Time Image Processing Through Dynamic Mode Decomposition

Jacob Grosek

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2013

Reading Committee:

J. Nathan Kutz, Chair

Ka-Kit Tung

Eric Shea-Brown

John Sahr

Program Authorized to Offer Degree:
Applied Mathematics

University of Washington

Abstract

Robust Real-Time Image Processing Through Dynamic Mode Decomposition

Jacob Grosek

Chair of the Supervisory Committee:
Dr. J. Nathan Kutz
Applied Mathematics

In many areas of research, robust and efficient data-mining, and data-driven modeling, have become essential to progressing our understanding of increasingly complex and nonlinear relations often embedded in cluttered and/or corrupted data. Applications in the industry and technology sectors are demanding high-performance algorithms that can tease out important information in real-time, and help humans interact better with computers. Researchers are starting to explore sparse modeling techniques that can sample sparsely and project onto a low-dimensional bases, where the essential information that drives the system can be extracted without having to build and run costly models of the full dynamics of the entire system. The focus of this work will be on the development of data analysis techniques that can be applied to solving the gesture recognition problem, with the goal of improving performance.

One of the most exciting results of the research presented here is found in the use of dynamic mode decomposition (DMD) as a viable and effective method for subtracting out the backgrounds for moving objects in videos. The method will be introduced and compared against the current standard method in the background subtraction field, namely robust principal component analysis (RPCA). The computational speed and accuracy of the DMD separation method offers a comfortable margin for real-time, online data processing. The fact that DMD approximates RPCA's low-rank/sparse

separation of data matrices opens the possibilities of various applications in the time-scale separation of dynamics arena, and even in the data compression and sparse sensing fields.

It will be shown that through enhanced pre-processing techniques, robust, accurate, and real-time gesture recognition can be achieved, even on heavily down-sampled images where gestures are nearly indistinguishable to the human eye. This is compared to the current trend in the computer vision field that advocates more complicated and computationally expensive feature selection and statistical learning routines that become problem specific and often still suffer from gesture irregularity in the data. Gesture recognition can be further improved by selecting the most appropriate gestures for the given task, accounting for ergonomic, vernacular, and algorithmic considerations. A new method, which combines these subjective and objective constraints into a single measure that indicates which gestures are best for the given application, is developed and tested on a real hand gesture recognition problem. Insights from the way that gestures render themselves in feature space help guarantee that this best lexicon methodology will be useful and reliable for realistic recognition problems.

In the appendix of this work, the theoretical background and practical implementation techniques needed in order to model a high-power Raman fiber laser/amplifier system that includes stimulated Brillouin scattering (SBS) and four-wave mixing (FWM) effects will be delineated, and the strengths and weaknesses of the computer model will be discussed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

The author wishes to express sincere appreciation to all of the professors and staff of the Applied Mathematics Department at the University of Washington for the great educational experience he had in that department. A special thanks is given to his advisor J. Nathan Kutz for all of his support, guidance, mentoring, patience, and enjoyable times working together! Also, Thomas Shay, Leanne Henry, Timothy Madden, Gerald Moore, and Eric Bochove acted as temporary mentors and advisors for the author while he worked at the Air Force Research Laboratories at Kirtland AFB. The author also acknowledges the generous support from the ARCS Foundation and the SMART Scholarship Program from the Department of Defense.

Chapter 1

INTRODUCTION

The consumer electronics market and technology industry have been seeing a growing interest in and demand for robust, real-time gesture recognition algorithms that can improve the human-machine interaction process. Indeed, technologies are now being proposed for future integrated voice, eye movement, facial expression, and hand gesture recognition capabilities in many electronic devices such as smart phones, laptops, televisions, game consoles, and tablets (See Fig. 1.1). The computer vision and statical learning communities have been working hard to answer this call, and to provide software that is intuitive and reliable enough to be used by the general public. Computer vision broadly includes mathematical methods and algorithms for acquiring, processing, analyzing, and understanding images and videos, often which are high-dimensional, in order to produce accurate decisions and classifications about what is observed [1, 2, 3, 4].

Many modern electronic devices now come with built-in cameras that can take fairly high resolution pictures and videos. Though, these cameras offer access to a great deal of information imbedded in the pixels and colors, it comes with a computational efficiency trade-off, where more data translates into slower processing times. Most image recognition techniques need to rely on an underlying *low-dimensional* structure to the gestures that they are attempting to detect and identify in order to meet the real-time operational requirement.

This dissertation will present novel and intelligent methods for aiding the recognition process, while achieving the ability to operate in real-time, and under a robust set of circumstances. Also, an improved method for background/foreground subtrac-

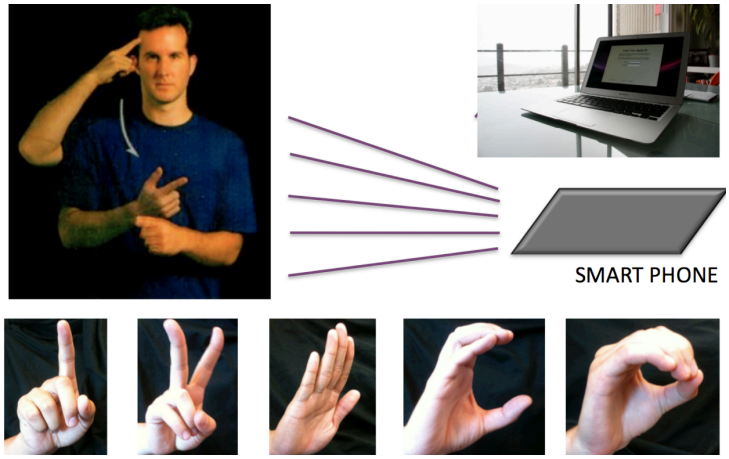


Figure 1.1: This figure illustrates a potential consumer electronics application for computer mouse control using hand gesture recognition. Five hand gestures represent the mouse operations of "single-click", "double-click", "cursor movement and control", and "scroll-up" and "scroll-down".

tion in video sequences will be introduced, which also has applications in time-scale separation for dynamical systems.

This dissertation can be outlined as follows...

1.1 Gesture Recognition

In chapter 2, the recognition process will be explained for gesture images, and some common algorithms for completing the recognition process will be delineated. The first step of the recognition process has been a particularly difficult challenge in the recognition field, namely subtracting stationary backgrounds out of video streams. This has been at the forefront of modern data-analysis research for quite a while, and has been a vexing task that has seen a very slow progression toward a truly competent and efficient algorithm. Background/foreground separation can be an integral step in detecting, identifying, tracking, and recognizing objects in video sequences. Most

modern computer vision applications demand algorithms that can be implemented in real-time, and that are robust enough to handle diverse, complicated, and cluttered backgrounds. Competitive methods often need to be flexible enough to accommodate changes in a scene due to, for instance, illumination changes that can occur throughout the day, or location changes to where the application is being implemented. A variety of algorithmic techniques and methods have been developed in order to separate video backgrounds from their foregrounds [5, 6, 7, 8, 9].

At the forefront of background/foreground subtraction in video sequences is a method developed by by Candès et al. [9] called *robust principal component analysis* (RPCA). The RPCA method views this computational task as a matrix separation problem that decomposes the video matrix into *low-rank* (background) and *sparse* (foreground) components. By weighting a combination of the nuclear and the L^1 norms, a convenient convex optimization problem (*principal component pursuit*) was demonstrated, under suitable assumptions, to exactly recover the low-rank and sparse components of a given data-matrix (video). The RPCA technique, which has its computational costs dominated by a convex optimization procedure, was shown to be highly-competitive in comparison to the state-of-the-art computer vision procedure described by De La Torre and Black [10]. This result follows decades of work attempting to robustify the PCA dimensionality-reduction technique, through, for instance, influence function techniques [10, 11], multivariate trimming [12], alternating minimization [13] and/or random sampling [14].

The separation technique presented in this chapter advocates the same approach as Candès et al. except by using the method of *dynamic mode decomposition* (DMD) [15, 16, 17, 18, 19, 20] instead of RPCA. This method, which essentially implements a Fourier decomposition of the video frames in time, distinguishes the stationary background from the dynamic foreground by differentiating between the near-zero modes and the remaining modes bounded away from the origin, respectively.

Originally introduced in the fluid mechanics community, DMD has emerged as

a powerful tool for analyzing the dynamics of nonlinear systems [15, 16, 17, 18, 19, 20]. In the context of fluids, DMD has gained popularity since it provides, in an *equation-free* manner, information about the dynamics of flow even if the underlying dynamics are nonlinear. It is equation-free in the sense that a typical application requires collecting a time series of experimental (or simulated) velocity fields, and computing DMD modes and eigenvalues from them. The modes are spatial fields that often identify coherent structures in the flow. The corresponding eigenvalues define growth/decay rates and oscillation frequencies for each mode. Taken together, the DMD modes and eigenvalues describe the dynamics observed in the time series in terms of growth, decay, and oscillatory components; i.e. it is a decomposition of the data into Fourier modes in time [17].

In the application of video surveillance, the video frames can be thought of as snapshots of some underlying complex/nonlinear dynamics. The DMD decomposition yields growth, decay, and oscillatory time components of the video frames that have contextual implications. Namely, those modes that are near the origin represent dynamics that are unchanging, or changing slowly, and can be interpreted as stationary background pixels, or low-rank components of the data matrix. In contrast, those modes bounded away from the origin are changing on $\mathcal{O}(1)$ timescales or faster, and represent the foreground motion in the video, or the sparse components of the data matrix. Thus, by simply applying the dynamical systems DMD interpretation to video frames, an effective RPCA technique can be enacted at a fixed cost of a singular-value decomposition and a linear equation solve, $\mathcal{O}(N^3)$. Unlike the convex optimization procedure of Candès et al. [9], which can be guaranteed to exactly produce a low-rank and sparse separation under certain assumptions, no such guarantees are currently given for the DMD procedure. Regardless, the RPCA [9] and computer vision [10] methods are orders of magnitude slower in computational performance, while the DMD method works in real-time on laptop-class computing power, which suggests that this DMD technique merits serious consideration.

While most of the current research is focused on creating more elaborate and sophisticated methods for extracting better gesture features and developing smarter statistical learning algorithms, an alternative approach to achieving robust recognition in real-time will be advocated. Through some simple, static hand gesture recognition examples, it will shown that pre-processing is key to improving the accuracy and computational efficiency while maintaining the robustness needed to handle a wide variety of situations that are experienced in real-world applications. Indeed, pre-processing is the area in this field that is most overlooked, and has the greatest potential to be improved algorithmically.

The lack of direct, head-to-head comparisons between both feature selection techniques and statistical testing methods as a function of image resolution and pre-processing methods in the recognition field is addressed for some common, yet important recognition algorithms. These comparisons will be used to indicate that with a well-designed pre-processing procedure, even simple, well known feature extraction and classification procedures can produce excellent results, and do so with very low overhead and processing times. The objective will be to demonstrate that improving the pre-processing routine is a competitive alternative, or even a complement, to developing more sophisticated feature selection and statistical discrimination methods for applications in the consumer electronics arena.

1.2 Best Gestures

Having a good gesture recognition algorithm that works does not necessarily guarantee accurate recognition for every application. In order to ensure a successful automatic recognition process, one needs to consider how the gestures of the application are handled by the computational process. For many software recognition applications, it is often the case that only a few gestures are needed for accomplishing the tasks of the application, which can be chosen from among a nearly endless number of gestures that a person can articulate. Thus given a large lexicon of gestures and an application

that requires only a small subset of those gestures, one would like to know which are the best gestures to choose for the given application; this will be explored in chapter 3. More succinctly, an algorithm is developed that chooses the best n gestures from a lexicon of m gestures, where typically $n \ll m$.

There are many factors, both objective and subjective, which may determine which gestures are most appropriate and useful for a particular application, i.e. best gestures. Certain gestures may be easier for the computer to process and recognize, but may not necessarily be comfortable or suitable for humans to articulate. Ergonomics, or the ease of articulating the various gestures, and relations to physical signs and gestures that are already in use in the culture or that are appropriate for the given application constitute *subjective* measures for claiming some gestures may be better than others. The *objective* reasons for ranking the quality of gestures comes purely from the computer's ability to distinguish and recognize gestures in a statistical sense. Both the subjective and objective reasons for determining the n best gestures from a lexicon with m elements need to be considered for designing gesture-based, robust software interfaces.

Objectively, the best lexicons of size n are determined by having the computer attempt to recognize every combination of n out of m gestures, and then choosing the best gesture set as the one with the highest overall successful recognition rate during a training process [21, 22]. This can be a time-consuming and combinatorially challenging problem. In this chapter, a new metric, called the *ellipsoidal distance ratio metric* (EDRM), is introduced that provides an excellent indication as to which gestures will be easily recognizable according to the computer. The EDRM is applied to the feature space of the gestures, and therefore does not require the computer to complete the entire recognition process in order to gain some notion as to which are the best gestures in the entire lexicon of available gestures [23]. Thus a robust and efficient algorithm is developed to extract the best n of m gestures as required.

1.3 Outlook

In chapter 4, the main points of this dissertation will be recapped and emphasized. Also, some future projects that can be derived from the work presented here will be mentioned.

Chapter 2

GESTURE RECOGNITION

In computer vision, the gesture recognition process starts when a raw image is imported into the computer. The gesture recognition process can be broken down into the following steps:

(i) gesture *detection*, *segmentation*, and *background subtraction*, which is the process by which the gesture is found and isolated within the image frame,

(ii) *pre-processing*, which is the process of normalizing like-gestures to similar sizes, shapes, colors, positions, and orientations,

(iii) *feature selection*, which is the process of determining important characteristics and aspects of gestures that will simultaneously distinguish between different gesture classes and will highlight like-gesture classes, and

(iv) *statistical learning* and *classification*, which is the process of training the computer to identify and recognize a gesture's articulation by statistical means and predictive functions that draw upon feature data.

Not all recognition schemes use all of these steps, or even complete the steps in the same order as they are listed above; however, the general procedure is still valid [4].

A great number of recognition techniques have been proposed [24, 25, 26, 27, 28, 29], which mostly focus on the image features being extracted and/or the development of the statistical learning techniques involved in the recognition process. However, many state-of-the-art techniques come at the cost of increased processing

times, trading for more complex and sophisticated algorithms that work on more realistic datasets.

This work allows for a direct performance and robustness comparison of various feature extraction and statistical testing techniques that hitherto have been deficient in the general literature of the recognition field. Such comparisons, made across different pre-processing scenarios, illustrate that accurate recognition can be accomplished using very little processing power and at low-resolution. By combining and implementing simpler, more computationally efficient recognition algorithms in unique and clever ways, robust and reliable recognition can be achieved in real-time. It will be demonstrated that the performance of recognition algorithms can be most enhanced, using minimal processing overhead, through improved pre-processing techniques.

Also, in this chapter, it will shown that the methods for finding the most identifiable features for each gesture suggest that there are ways to determine which features play more significant roles in the recognition process, i.e. which are the *best features*. Likewise, a novel approach for inferring which gestures are most easily and efficiently distinguished from one another is presented, i.e. a technique for finding the *best gestures*. This method allows for one to create gesture lexicons that are best suited for given applications, guaranteeing proficient algorithmic performance, and accounting for both subjective and objective considerations that make some gestures more optimal than others.

2.1 Background/Foreground Separation

Segmentation and background removal is an important step in gesture recognition. It is understood that most modern applications to the gesture recognition field will want to take advantage of the movement of gestures that can be obtained from video sequences. This gesture movement, along with well-established gesture identification or localization techniques like the Viola-Jones detector [30], the Gaussian mixture model (GMM), kernel density estimation (KDE), the max/min inter-frame differences tech-

nique [31], and robust principal component analysis [9] (RPCA) provide excellent ways to identify the gesture within its background. Further, the use of passive infrared cameras in consumer electronic devices may afford a great deal of flexibility in automatically removing the hot gesture from its cold background. No doubt, a noisy, complex, and/or cluttered background will make any detection algorithm less robust. A robust method that can segment an object of interest in a static image from its background is an essential part of almost all static gesture recognition processes, and is an on-going topic of research. With dynamic gestures in videos, this segmentation is known as background subtraction, and has the added advantage of containing extra information about what objects are moving through time, leaving background variations that are highly correlated between frames.

This background/foreground separation problem has been looked at by many researchers over the years with limited success [5, 6, 7, 8, 9]. However, recently, Candès et al. [9] advocated an approach that breaks apart the video matrix into low-rank and sparse components, which, for most conditions, separates the video into its background and foreground, respectively. This is accomplished through robust principal component analysis (RPCA), which decomposes a given data matrix $\mathbf{X}$ into

$$\mathbf{X} = \mathbf{L} + \mathbf{S}, \tag{2.1}$$

where $\mathbf{L}$ has a low-rank and $\mathbf{S}$ is sparse. At first, the ability to separate such a matrix $\mathbf{X}$ into its constitutive components seems to be an impossible task considering that the rank of $\mathbf{L}$ and the locations of the non-zero (sparse) components of $\mathbf{S}$ are unknown. Amazingly, Candès was able to prove that the convex optimization procedure could *exactly* recover the low-rank and sparse matrices under suitable circumstances, thus allowing for a highly-robust procedure for matrix separation and completion.

Presently, it will be demonstrated, without proof of convergence, that this low-rank/sparse separation can also be performed by the dynamic mode decomposition (DMD) procedure. And, in some aspects, the DMD method is more robust and

efficient at effecting this separation than the RPCA method. At its core, the DMD method relies on a underlying *separation of time-scales* between the sparse and the low-rank structures in the data matrix. This also means that, through the DMD method, other dynamical systems can be decomposed into their fast and slow time-scale components, an appealing ability for a wide range of applications.

2.1.1 Robust PCA Theory

For many modern data analysis problems, *principal component analysis* (PCA) has become a fairly ubiquitous means for achieving dimensionality reduction. The PCA method, and many other statistical methods for reducing dimensionality, rely on the L^2-norm for optimizing data-fitting because of its convenient mathematical properties and ease of interpretation as a measure of an *energy* in the system. However, the L^2-norm suffers greatly in the presence of corrupted data because of its sensitivity to outliers, which easily skew the results by weighing them more heavily than one would desire. Refer to subsection A.2.1 of the appendix for further details on PCA.

There are other norms available to be used as a measure for data-fitting; specifically, the L^1-norm is a good candidate because it does not over-weight sparse outliers in the data and can be used to promote sparsity. Indeed, the L^1-norm has been recently demonstrated as a successful and extremely broad measure with many favorable physical interpretations [32]. The robust principal component analysis (RPCA) algorithm improves the classical PCA algorithm by mixing the use of the two different norms in order to make the dimensionality reduction more robust to, potentially grossly corrupted, sparsity in the data. More details about the RPCA algorithm are sketched out in subsection A.1.1 of the appendix.

Video Interpretation of the RPCA Method

In a video sequence, stationary background objects translate into highly correlated pixel regions from one frame to the next, which suggests a low-rank structure within

the video data. In the case of videos, where the data in each frame is 2D by nature, frames ought to be reshaped into 1D column vectors and united into a single data matrix $\mathbf{X}$. The RPCA algorithm can then implement the background/foreground separation found in equation (2.1), where the low-rank matrix $\mathbf{L}$ will render the video of just the background, and the sparse matrix $\mathbf{S}$ will render the complementary video of the moving foreground objects. Because the foreground objects exhibit a spatial coherency throughout the video, the RPCA method is no longer guaranteed a high probability of success; however, in practice, RPCA achieves an acceptable separation almost every time [9].

Figure 2.1 illustrates the quality of background/foreground separation as the regularization parameter λ is varied about its suggested value of $\lambda = (\sqrt{n})^{-1}$. In this case $n = 11520$, being the number of pixels per frame; hence, $\lambda \approx 9.32 \cdot 10^{-3}$. Artificially brightened by a factor of 10, the RPCA method performs on par with the DMD method, which will be explained in the next section, in terms of separation. However, as λ is decreased, the sparse reconstruction of the video, which is stored in matrix $\mathbf{S}$, starts to incarnate more of the original video, including erroneous stationary pixels that should be part of the low-rank background. When λ is increased, the sparse reconstruction of the video begins to see a decrease in the pixel intensities that correspond to the moving objects, and even some foreground pixels disappear all together.

2.1.2 DMD Theory

Dynamic mode decomposition (DMD) is a mathematical method that was developed in order to understand, control, or simulate inherently complex, nonlinear systems without knowing fully, or partially, the underlying governing equations that drive the system. Experimental, or simulated data, collected in snapshots through time, can be processed with DMD in order to mimic, control, or analyze the current state or dimensionality of a system, or even to predict future states of, or find coherent struc-

tures within, that system. The power of DMD is found in exploiting the intrinsic low-dimensionality of a complicated system, which may not be obvious *a priori*, and then rendering that system in a more computationally and theoretically tractable form. DMD has been traditionally used in the fluid mechanics, atmospheric science, and nonlinear waves communities as a popular method for data-based learning and discovery [15, 16, 18, 20]. The details of the DMD method are delineated in subsection A.1.2 of the appendix.

2.1.3 Video Interpretation of the DMD Method

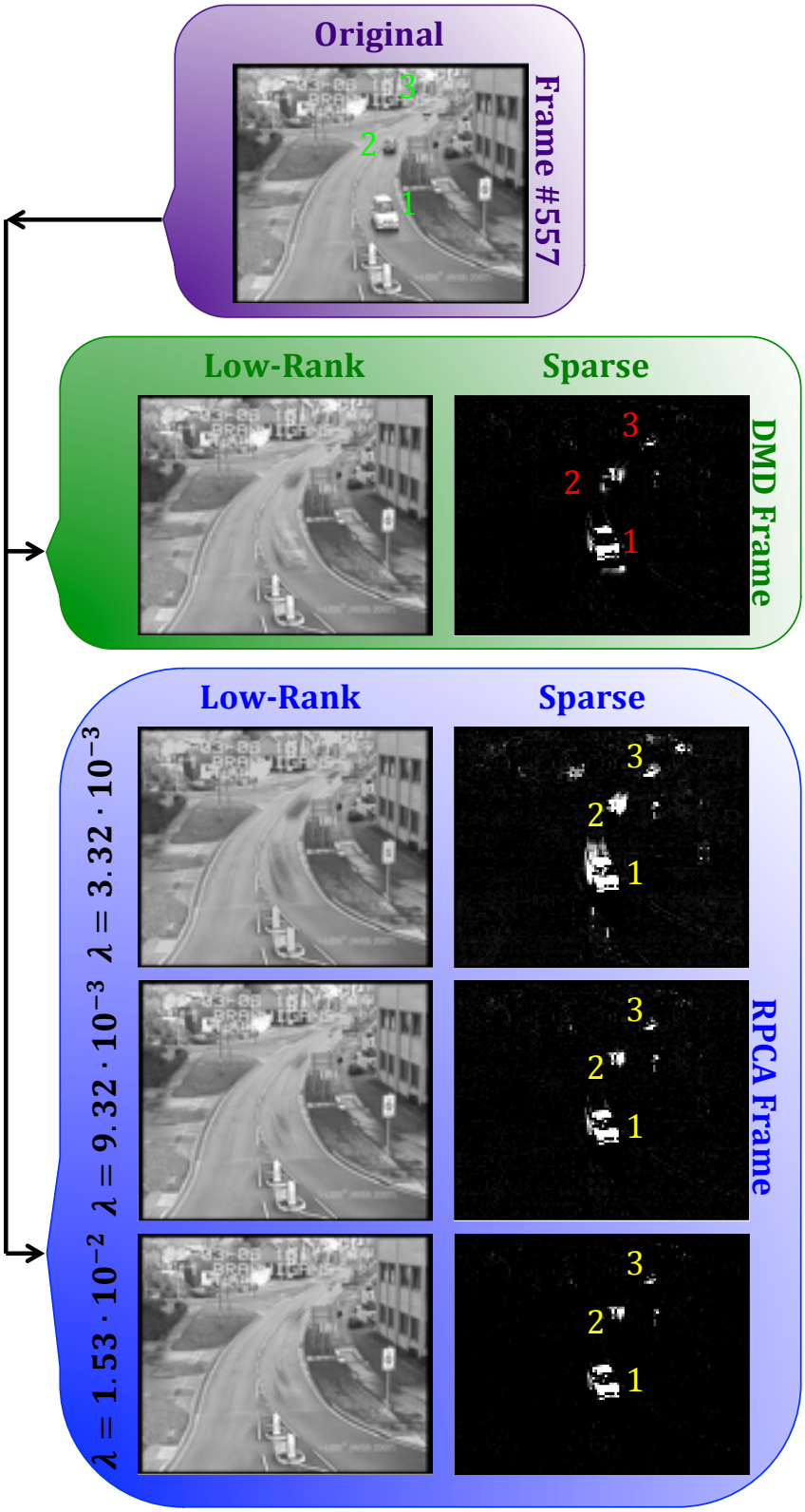
A video sequence offers an appropriate application for this DMD method because the frames of the video are, by nature, equally spaced in time, and the pixel data, collected in every snapshot, can readily be vectorized. Given m frames to the video stream, the $n \times 1$ vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$ can be extracted, which contain the pixel data of each frame; there being n pixels in total per frame. The DMD method can attempt to reconstruct any given frame, or even possibly future frames, by calculating $\mathbf{x}_{\text{DMD}}(t)$ at the corresponding time t, as is described in equation (A.6). The validity of the reconstruction depends on how well the specific video sequence meets the assumptions and criteria of the DMD method.

In order to reconstruct the entire video, consider the $1 \times m$ time vector $\mathbf{t} = [t_1 \ t_2 \ \ldots \ t_m]$, which contains the times at which the frames were collected. If $t_j = j - 1 \ \forall j$, then time becomes equivalent to the frame count, where the first frame is labelled as 0 and the m^{th} frame is labelled as $m - 1$. The video sequence $\mathbf{X}$ is reconstructed with the DMD technique as follows...

$$\mathbf{X}_{\text{DMD}} = \sum_{j=1}^{\ell} b_j \boldsymbol{\varphi}_j e^{\omega_j \mathbf{t}} := \boldsymbol{\Phi}\boldsymbol{\Omega}^{\mathbf{t}}\mathbf{b}.$$

Notice that $\boldsymbol{\varphi}_j$ is a $n \times 1$ vector, which is multiplied by the $1 \times m$ vector $\mathbf{t}$, to produce the proper $n \times m$ video size. By the construction of the DMD methodology: $\mathbf{x}_1 = \boldsymbol{\Phi}\mathbf{b}$, which means that $\boldsymbol{\Phi}\mathbf{b}$ renders the first frame of the video with a dimensionality

Figure 2.1: This figure demonstrates the sensitivity of the RPCA method to the λ parameter value. In Frame #557 of the "Parked Vehicle" video [33], there are 3 moving vehicles to be separated from the background. Using video segments of 30 frames, the DMD background/foreground separation method finds the 3 moving vehicles, with few erroneous pixels included as part of the foreground. The RPCA method also achieves similar results with the recommended regularization parameter value of $\lambda = 9.32 \cdot 10^{-3}$. As this parameter is changed by only ± 0.006, either extra, foreign pixels get included in the foreground ($\lambda = 3.32 \cdot 10^{-3}$), or two of the three cars begin to fade away and blend into the background ($\lambda = 1.532 \cdot 10^{-2}$). Note that the sparse results for both the DMD and RPCA methods are artificially brightened by a factor of 10 in order to illuminate all the dark extraneous pixels that would normally not be visible against the black background.


Original
Frame #557
3
2
1
Low-Rank
Sparse
DMD Frame
3
2
1
Low-Rank
Sparse
RPCA Frame
$\lambda = 1.53 \cdot 10^{-2}$ $\lambda = 9.32 \cdot 10^{-3}$ $\lambda = 3.32 \cdot 10^{-3}$
3
2
1
3
2
1
3
2
1

reduction chosen through the parameter ℓ. Thus, the diagonal matrix $\mathbf{\Omega}^t$ dictates how that first frame gets altered over time in order to reconstruct the subsequent frames. It becomes apparent that any portion of the first video frame that does not change in time, or changes very slowly in time, must have an associated Fourier mode (ω_j) that is located near the origin in complex space: $\|\omega_j\| \approx 0$. This fact becomes the key principle that makes possible the ability of the DMD method to separate background (low-rank) information from foreground (sparse) information.

Assume that ω_p, where $p \in \{1, 2, \ldots, \ell\}$, satisfies $\|\omega_p\| \approx 0$, and that $\|\omega_j\| \ \forall \ j \neq p$ is bounded away from zero. Thus,

$$\mathbf{X}_{\text{DMD}} = \underbrace{b_p \boldsymbol{\varphi}_p e^{\omega_p \mathbf{t}}}_{\text{Background Video}} + \underbrace{\sum_{j \neq p} b_j \boldsymbol{\varphi}_j e^{\omega_j \mathbf{t}}}_{\text{Foreground Video}} \tag{2.2}$$

Assuming that $\mathbf{X} \in \mathbb{R}^{n \times m}$, then a proper DMD reconstruction should also produce $\mathbf{X}_{\text{DMD}} \in \mathbb{R}^{n \times m}$. However, each term of the DMD reconstruction is complex: $b_j \boldsymbol{\varphi}_j \exp(\omega_j \mathbf{t}) \in \mathbb{C}^{n \times m} \ \forall j$, though they sum to a real-valued matrix. This poses a problem when separating the DMD terms into low-rank and sparse reconstructions because real-valued outputs are desired and knowing how to handle the complex elements can make a significant difference in the accuracy of the results. Consider calculating the DMD low-rank reconstruction according to

$$\mathbf{X}_{\text{DMD}}^{\text{Low-Rank}} = b_p \boldsymbol{\varphi}_p e^{\omega_p \mathbf{t}}.$$

Since it should be true that

$$\mathbf{X} = \mathbf{X}_{\text{DMD}}^{\text{Low-Rank}} + \mathbf{X}_{\text{DMD}}^{\text{Sparse}},$$

then the DMD sparse reconstruction,

$$\mathbf{X}_{\text{DMD}}^{\text{Sparse}} = \sum_{j \neq p} b_j \boldsymbol{\varphi}_j e^{\omega_j \mathbf{t}},$$

can be calculated with real-valued elements only as follows...

$$\mathbf{X}_{\text{DMD}}^{\text{Sparse}} = \mathbf{X} - \left| \mathbf{X}_{\text{DMD}}^{\text{Low-Rank}} \right|,$$

where $|\cdot|$ yields the modulus of each element within the matrix. However, this may result in $\mathbf{X}_{\text{DMD}}^{\text{Sparse}}$ having negative values in some of its elements, which would not make sense in terms of having negative pixel intensities. These residual negative values can be put into a $n \times m$ matrix $\mathbf{R}$ and then be added back into $\mathbf{X}_{\text{DMD}}^{\text{Low-Rank}}$ as follows...

$$\mathbf{X}_{\text{DMD}}^{\text{Low-Rank}} \leftarrow \mathbf{R} + \left|\mathbf{X}_{\text{DMD}}^{\text{Low-Rank}}\right|$$
$$\mathbf{X}_{\text{DMD}}^{\text{Sparse}} \leftarrow \mathbf{X}_{\text{DMD}}^{\text{Sparse}} - \mathbf{R}$$

This way the magnitudes of the complex values from the DMD reconstruction are accounted for, while maintaining the important constraints that

$$\mathbf{X} = \mathbf{X}_{\text{DMD}}^{\text{Low-Rank}} + \mathbf{X}_{\text{DMD}}^{\text{Sparse}},$$

that none of the pixel intensities are below zero, and ensuring that the low-rank and sparse DMD reconstructions are real-valued. This method seems to work well empirically.

In terms of video streams, dimensionality reduction, done through the parameter ℓ, has the effect of blurring the frames together through time. Thus, for the sake of most accurately separating foreground objects from their background counterparts, it is recommended that ℓ is fixed to be the as large as possible, which is one less than the number of frames in the video sequence, or $m-1$. This way there are sharp contrasts in movements between frames so that the algorithm can accurately detect the object itself, and not a blurred pixel trail that follows behind the movement.

2.1.4 Performance Metrics and Limitations

With the DMD method explained, the task of separating background/foreground information in video sequences, which is an application of great importance to the computer vision community, can be accomplished.

Surveillance Video Application

Using the Advanced Video and Signal based Surveillance (AVSS) Datasets [33], specifically the "Parked Vehicle - Hard" and "Abandoned Bag - Hard" videos, the DMD separation procedure can be compared and contrasted against the RPCA procedure. The original videos are converted to grayscale and down-sampled in pixel resolution to $n = 120 \times 96 = 11520$, in order to make the computational memory requirements manageable for personal computers. Also, the introductory preambles to the surveillance videos, which constitute the first 351 frames of each video, are removed because they are irrelevant for the following illustrations.

The video streams are broken into segments of $m = 30$ frames each, which are analyzed individually using both the RPCA and DMD methods. Frame numbers 500, 1000, and 2000 of the entire video streams are depicted in figures 2.2 (a) and (b), along with their separation results for easy comparison. Although one has the option of manually tuning the regularization parameter of the RPCA method to best suite the given application, for fairness λ is set at $\lambda = (\sqrt{n})^{-1} \approx 9.32 \cdot 10^{-3}$, as is suggested by Candès et al. [9] for creating a reliable automatic algorithm. Likewise, the dimensionality reduction parameter of the DMD method is held constant at $\ell = m - 1 = 29$ for the sake of producing the best results. For enhanced contrast and better visibility, the sparse results from both methods are artificially brightened by a factor of 10.

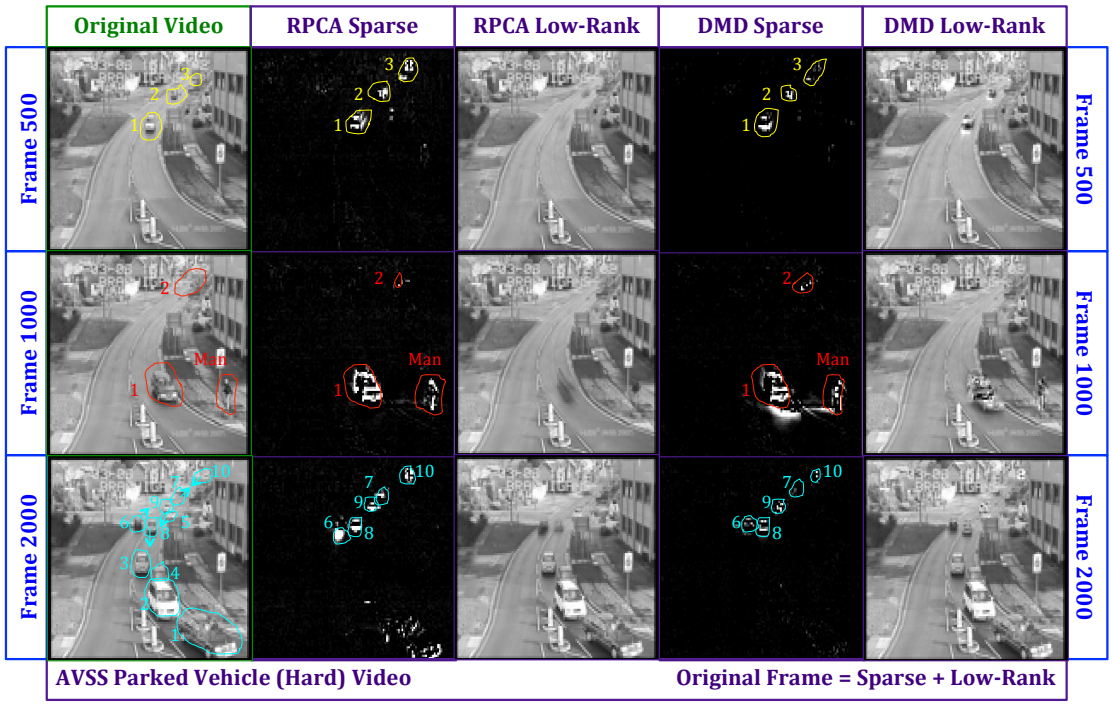
Consider Fig. 2.2 (a) of the AVSS "Parked Vehicle" surveillance video, which generally shows various vehicles traveling along a road, with a traffic light (not visible) and a crosswalk (visible) near the bottom of the frame, and with an occasional vehicle parking along side the road. Sometimes, In the distance, moving vehicles become difficult to perceive with the naked-eye, limited by the pixel resolution. Note that, for all three frames, the RPCA method is unable to eliminate spurious background pixels from its sparse results, whereas the DMD method does a much better job

of capturing only the moving objects. Many of the erroneous pixels in the sparse structure could be eliminated by applying a simple thresholding criterium; though the RPCA results would likely still contain unwanted background pixels because they are nearly as bright as the foreground pixels.

Consider Fig. 2.2 (b) of the AVSS "Abandoned Bag" surveillance video, which generally depicts people walking, standing, and sitting as trains come and go in a subway station. Shadows are relevant in this video because they do move with their respective foreground objects, and they do sometimes change the background significantly enough to be viewed as extensions of the moving objects themselves. Note that, for frames 500 and 1000, both methods struggle with the fact that between the numerous moving objects and their shadows, many of the pixels in the video change intensity at some point. Objects of low intensity, nearly or all black, create bright pixelated movement trails that provide a cartoonish sense of motion. The DMD method is generous in depicting moving shadows in the its sparse results, while the RCPA method is more conservative. In Frame 2000, the DMD method picks up many unwanted background pixels in its foreground estimation of the walking woman, but the RPCA method seems to capture her motion much better, as is evidenced by the fact that she is almost completely absent from the RPCA low-rank frame.

Figures 2.2 (a) and (b) seem to indicate that the RPCA and DMD methods have comparably good background/foreground separation results. The difference in the separation quality seems to depend on the specific situation, likely because the assumptions that determine when the RPCA and DMD methods will perform well are also different. Though, given the relatively consistent quality of the separation results, other factors, such as computational effort, become the distinguishing characteristics that determine which method is more suitable for the given application.

Figure 2.2: The DMD and RPCA background/foreground separation results are illustrated for 3 specific frames in the "Parked Vehicle" video (a) and in the "Abandoned Bag" video (b). In video (a), the 30 frame video segment that contains frame #500 has 3 vehicles driving in the same lane toward the camera. In video (a), the 30 frame video segment that contains frame #1000 has a man stepping up to a crosswalk as a vehicle passes by, and a second car in the distance, barely perceptible, starts to come into view. In video (a), the 30 frame video segment that contains frame #2000 has 3 vehicles stopped at a traffic light at the bottom of the frame with another 2 vehicles parked on the right side of the road, and 5 moving vehicles, 2 going into the distance and 3 coming toward the vehicles waiting at the light, the last vehicle being imperceptible to the eye at this pixel resolution: $n = 11520$. In video (b), the 30 frame video segment that contains frame #500 has 3 people stepping slightly closer to an arriving train, and another person walks behind a support pillar, toward the train in the upper right area of the frame. The two people closest to the camera walk toward the bottom of the frame. In video (b), the 30 frame video segment that contains frame #1000 has 4 people walking in different directions in the middle of the frame, one of which comes out from behind a support pillar, while, farther down the platform, 2 people enter the train. In video (b), the 30 frame video segment that contains frame #2000 has a woman walk out from behind a support pillar, moving to the left, and then turning somewhat toward the camera. The train is moving and the man sitting closest to the support pillar adjusts his backpack.



(a) "Parked Vehicle" video.

(b) "Abandoned Bag" video.

Timing Performance

The real difference in effectiveness between the RPCA and DMD algorithms is found in the amount of computational time needed to complete the background/foreground separation. Again, consider the AVSS Datasets, with the two videos used previously: "Parked Vehicle" and "Abandoned Bag". For a timing performance experiment, consider having these videos down-sampled to various pixel resolutions n and separated into various video segment sizes m. Averaging the computational time for either fixed numbers of pixels or fixed video segment sizes, and using both the RPCA method and the DMD method on both videos, Fig. 2.3 was produced. Both the exact ALM and the inexact ALM convex optimization routines were used for solving the PCP problem (A.1) of the RPCA method.

In Fig. 2.3, the empirical computational times are plotted on a logarithmic scale, along with best-fit curves found by the linear least squares method. It is clear that the DMD method is about $2-3$ orders of magnitude faster than its RPCA method counterpart using the exact ALM optimization procedure, and is about 1 order of magnitude faster when the inexact ALM optimization procedure is employed. In fact, given that many cameras operate at a rate of about $20-30$ frames per second and that the DMD method can be completed for those video segment sizes in about $0.1-0.01$ seconds for high and low resolution images, respectively, then real-time, on-line data processing is possible, even without downsampling.

Parameter Test

As was mentioned previously, there is no need for implementing a dimensionality reduction in the DMD algorithm for background/foreground video separation. However, in order to portray the effect that the dimensionality reduction parameter ℓ has on the DMD process, consider the foreground DMD results represented in Fig. 2.4.

As is expected, the highest contrast in movement between frames, and therefore

the best sparse separation results, corresponds to having the least amount of dimensionality reduction, namely when $\ell = m - 1$. The surprising result depicted in this figure is the quality of the low-rank and sparse separation even when ℓ is set as low as 1. Since dimensionality reduction has the effect of blurring the video frames together, it is impressive that there are not more erroneous background pixels appearing in the sparse results for the smaller values of ℓ. However, in the next experiment, it will seen that these good results are in part due to the fact that the vehicles in the video of Fig. 2.4 did not move very far within the frame, and so the smearing of pixels over the places of movement is very localized. In either case, this suggests yet another *time-saving* procedure that can potentially reduce the costs even further of reconstructing the sparse and low-rank video segments.

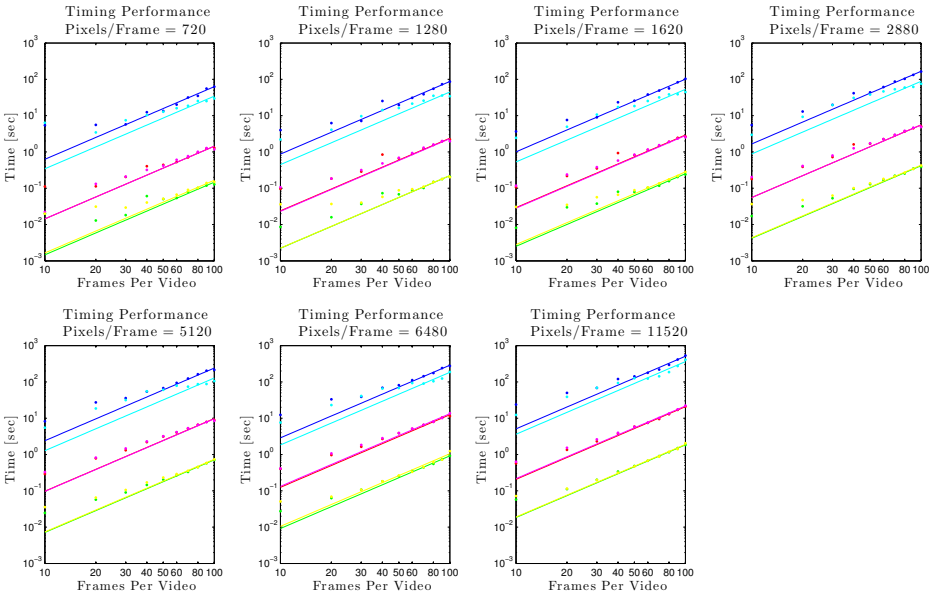
DMD Error Analysis

In order to accurately measure how well the DMD method captures actual foreground movement, and not the stationary background, it would be helpful to have a constructed video where the true background and foreground are known. The error could then be measured precisely between the actual and DMD reconstructed backgrounds. Moreover, the quality of reconstruction can be also compared with the RPCA technique.

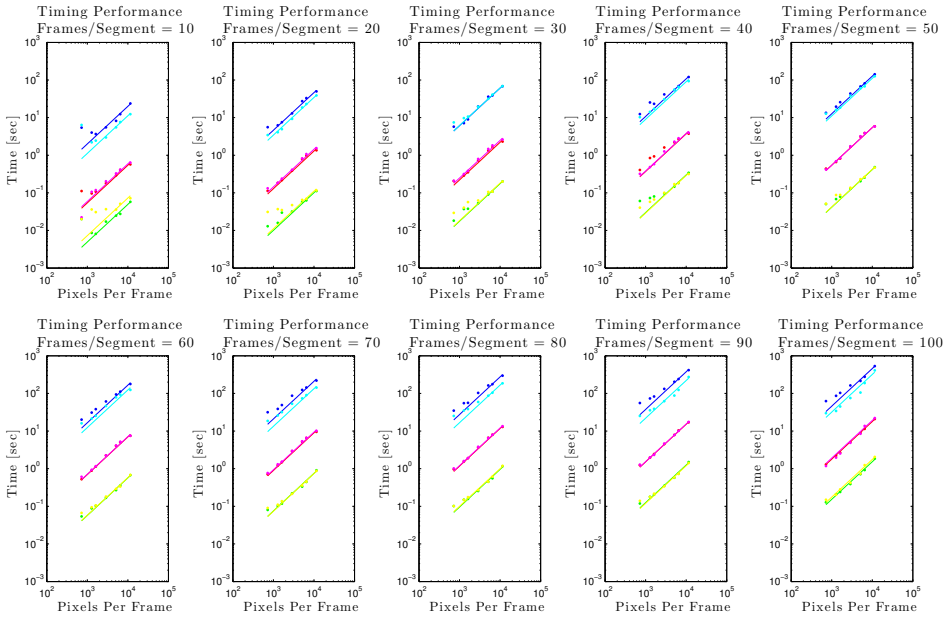
One such video was constructed for calculating the error in the DMD low-rank/sparse separation method with 300 frames in total, each being 100×100 pixels. The background to the video is produced by a uniform random, grayscale intensity field, ranging form pure black to pure white. The background remains constant throughout the entire video.

This video contains three moving objects: (1) a black square, 7×7 pixels in size, with four white pixels in each corner and a white "plus" sign centered in the middle, (2) a light gray circle with a diameter of 9 pixels, and (3) a transparent square, 13×13 pixels in size, lined by black pixels and with a black "X" centered within. Object (1)

Figure 2.3: DMD timing performance on a 1.86 GHz Intel Core 2 Duo processor using Matlab. For the "Abandoned Bag" and "Parked Vehicle" videos, the computational times of the DMD (green & yellow, respectively), inexact ALM RPCA (red & magenta, respectively), and exact ALM RPCA (blue & cyan, respectively) background/foreground separation methods are graphed on a logarithmic scale. Figure (a) holds the numbers of pixels per frame fixed at $\{720, 1280, 1620, 2880, 5120, 6480, 11520\}$. This timing data fits reasonably well with a quadratic fit: $t = cs^2$, where t is the computational time, $c \in \mathbb{R}$, and s is the video segment size (number of frames per video segment). Figure (b) holds the numbers of frames per video segment fixed at $\{10, 20, 30, \ldots, 100\}$. This timing data fits reasonably well with a linear fit: $t = cs$, where t is the computational time, $c \in \mathbb{R}$, and s is the frame resolution size (number of pixels per frame).



(a) Fixed pixel resolution.



(b) Fixed video segment size.

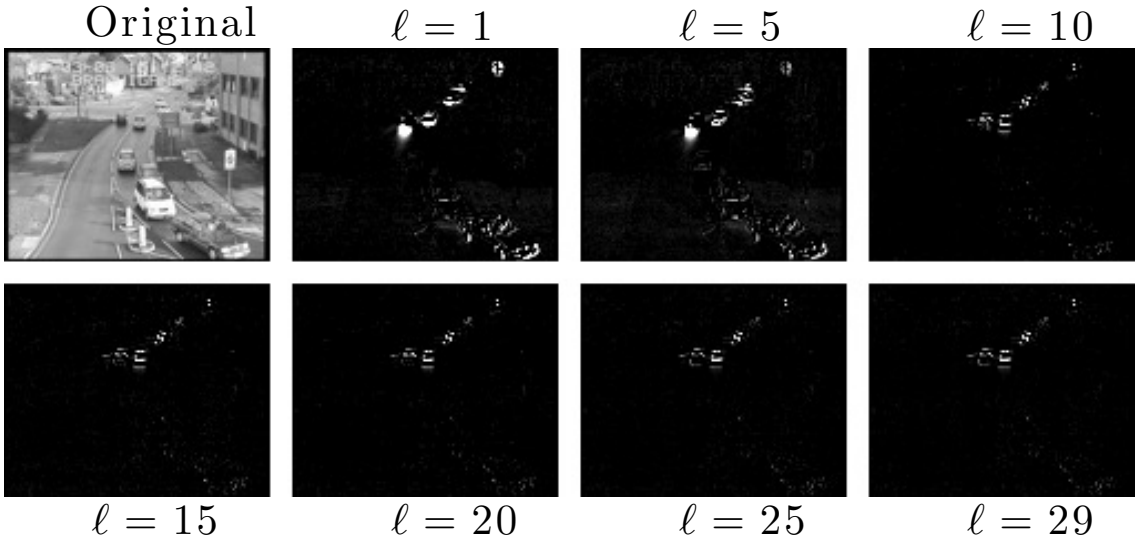


Figure 2.4: The effects of the parameter ℓ that controls the dimensionality reduction of the DMD process is depicted. The original frame, and the DMD sparse reconstructions of that frame, artificially brightened by 10 times the true intensity, are illustrated for frame #2000 of the "Parked Vehicle" video [33] using a 30 frame video segment (See Fig. 2.2 (a)). Notice that the sparsity increases with the parameter ℓ because dimensionality reduction has the effect of blurring the motion between frames, and thus smearing the motion out over a greater number of pixels in any given frame. Also, note that even under tremendous dimensionality reduction (ℓ = 1) the background/foreground separation is still accomplished reasonably well. Compare this to changing the regularization parameter λ in the RPCA method of Fig. 2.1, where ℓ = 29 for DMD method in that figure.

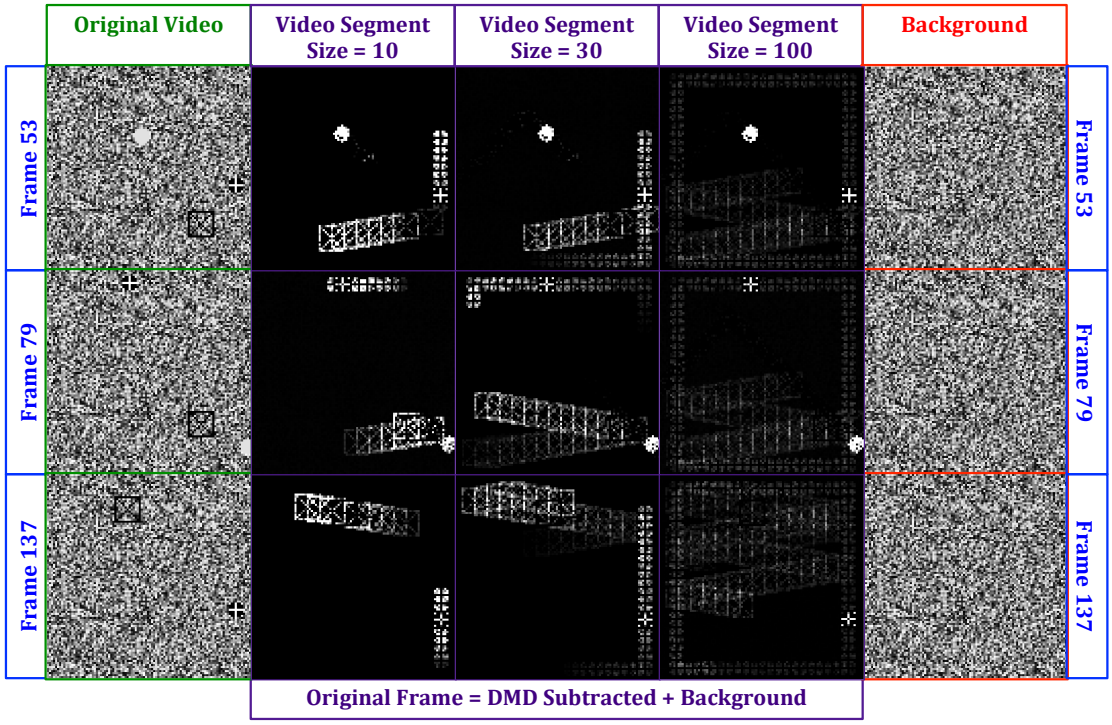
revolves, at a constant rate of 4 pixels per frame, counter-clockwise around the inside edges of the frame starting at the top left corner of the frame for the entire duration of the video. Object (2) enters the video on frame 25 on the left side of the frame, moving up and to the left at a constant rate of 2 pixels per frame in both directions.

This object reflects downward and eventually leaves the frame after bouncing off of an imaginary wall located a fifth of the way down from the top of the frame. Object (3) enters the video on frame on frame 50 on the right side of the frame, moving down and to the left. Once this object enters the frame it stays inside the frame for rest of the video, reflecting off of the edges of the frame. Object (3) maintains a constant vertical velocity of 1 pixel per frame, and a horizontal velocity of 6 pixels per frame. At various times these objects overlap one another, and the precedence is that object (1) is drawn first, then object (2), then object (3), burying the previously drawn objects underneath the newly drawn object.

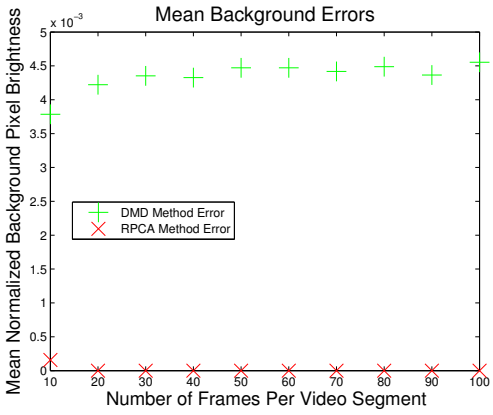
The foreground results of the DMD method applied to this video are illustrated in Fig. 2.5 (a) for frame numbers 53, 79, and 137. Again, the pixel intensities of the sparse frames are amplified by a factor of 10 in order to increase the contrast and visibility of the results against a black background. Note that object (3) cannot be seen in the results because it is pure black. This exemplifies a limitation to the DMD method, in that when there is an object of low pixel intensity, it may be difficult to distinguish it from the zero entries that naturally occur in a sparse matrix. Note that the spurious pixels are both residuals of previous frames and projections of where the objects will be moving in future frames, and have inverted pixel intensities from the black pixels with the objects. This is much like what is seen in the foreground DMD results of frames #1000 and #2000 of the "Abandoned Bag" video (See Fig. 2.2 (b)), where the walking man (frame #1000) and walking woman (frame #2000) are both depicted with mostly dark pixels, and are nearly indistinguishable from the sparse background, but their movement trails are bright white. The gray to white pixels do not seem to create these extraneous movement trails.

The mean, normalized pixel intensity error for this constructed video is shown in Fig. 2.5 (b). This error is calculated on the first $m-1$ frames of the video segments of length m because the last frame is where the DMD procedure accumulates its error. Note that the background to the sparse DMD reconstructed videos should be

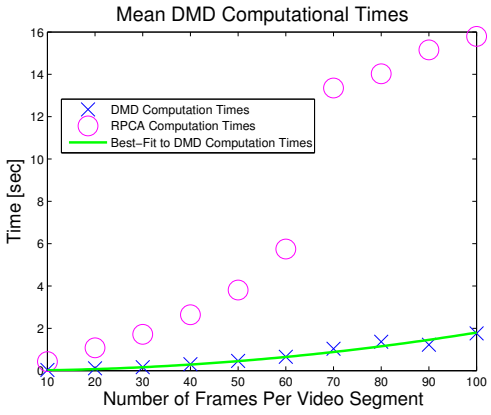
Figure 2.5: The DMD background/foreground separation results for the 300 frame video used for an error analysis are presented here. In the top figure (a), the left column shows the original frames with the random background noise that is fixed in time. The middle three columns show the sparse DMD results using 10, 30, and 100 frames per video segment, which are artificially brightened by a factor of 10, and the right column shows the true background noise that was added to each frame. The erroneous pixels contribute to the mean pixel intensity error of the background. The mean pixel intensity error for the entire video at each video segment size is plotted in the bottom left figure (b) on a normalized intensity scale where 1 is pure white and 0 is pure black, the target background color. The mean computational times t, using Matlab on a 1.86 GHz Intel Core 2 Duo processor, as a function of the video segment sizes s is plotted in the bottom right figure (c); yielding a quadratic best-fit: $t = 1.52 \cdot 10^{-4} s^2$, $R^2 = 0.983$.



(a) Constructed Video DMD Separation



(b) Error Analysis



(c) Timing Analysis

pure black, and on an intensity scale of 0 to 1, the average sparse background pixel intensity is on the order of 10^{-3}, compared to the true average background intensity of nearly identically 0.5. Recall that, in some cases, thresholding techniques can eliminate spurious background pixels from the foreground results, and improve the measured error.

One might have expected that the DMD algorithm errors would have decreased as the video segment sizes increased, due to the fact that having more frames means that the DMD method has more information to work with. However, because of this anomaly with black objects leaving white movement trails, there are extraneous pixels for every video segment size. As the videos get longer, these erroneous movement trails also get longer, projecting both into the past and future increasing the amount of error in proportion to the increase in number of frames per video segment. However, the longer videos do produce less bright pixel intensity trails, which helps reduce the error, which is consistent with the idea that they should be able to make more use of the extra information that they have.

The RPCA reconstructed foreground error results are also presented for comparison, where the suggested $\lambda = (\sqrt{n})^{-1} = 0.01$ is used. In this case, the RPCA perfectly reconstructs every video for segment sizes greater than 10 frames.

Figure 2.5 (c) reconfirms the quadratic growth in computational times that the DMD scheme experiences as the video segment sizes are increased. This quadratic growth trails off slightly for very small segment sizes, likely due to inherent programming processing times that do not scale with data size. For segment size 10, there is, of course, the option of retuning the regularization parameter λ, however, in most cases where the true solution is unknown, this must be done manually by time-consumingly rerunning the RPCA algorithm.

Finally, it should be noted that as object size increases relative to the frame size, the DMD reconstruction error also increases. Likewise, and not surprisingly, the DMD reconstruction error increases if less frames are used to show the same object

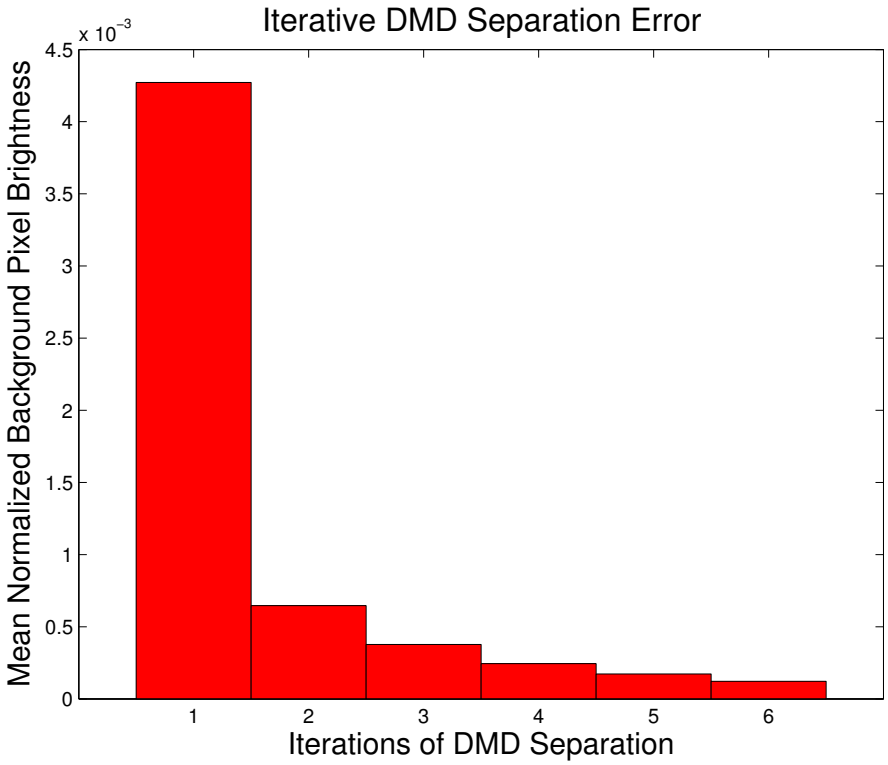


Figure 2.6: In this figure, the mean normalized background pixel intensity, a measure of DMD separation error, is shown to decrease as the DMD separation method is applied iteratively to the previous iteration's sparse component result. The first error depicted above ($i = 1$) corresponds to first separation of the original video data, and the rest of the errors correspond to the iterations of consecutively applying the DMD method ($i = 2, 3, \dots$).

movement. Not all types of object movement are accurately reconstructed with the DMD method; however, it seems that high-speeds and acceleration can still be handled fairly well given enough snapshots to capture the movement. Objects that were once moving and then stop is an event that is not well reconstructed by the DMD method.

Surveillance Video Challenge

It is worth noting that the "Parked Vehicle" and "Abandoned Bag" videos that came from the AVSS Datasets were created and posted publicly as part of the i-LIDS bag and vehicle detection challenge [33]. In order to spark further innovation and progress in the security and surveillance field, the Home Office Scientific Development Branch (HOSDB) of the UK issued a challenge for researchers use computer algorithms to automatically detect any parked cars along a street or abandoned bags in an underground station, and send out an alert. This challenge illuminates another potential application for this DMD method.

Given that a normal frame acquisition rate for most cameras is about 30 frames per second, it is clear that the DMD background/foreground separation method is well within the range of realtime operation, even at high image resolutions. Streaming surveillance video would likely be analyzed in segments of a few seconds or less worth of frames. Effectively, the fact that short video segment sizes can be analyzed separately offers another dimension to the time-scale separation problem because the backgrounds that are detected can be stored and compared with the most recent detected background.

By comparing backgrounds determined by the DMD method over the course of an appropriate amount of time, it would be fairly easy to detect the desired objects that stop moving and become part of the background itself. Objects that change or move on much shorter or longer time-scales than the one that corresponds to the given application are easily filtered out. Vehicles may stop temporarily at traffic lights, or for crossing pedestrians, but they start to move again in a fairly predictable amount of time. Likewise, people and their belongings may stay motionless for a while, waiting for their train, but the arrival and departure of trains allows for a practical limit on the most amount of time that anyone or anything should stay put before one gets concerned. Changing lighting conditions throughout the day may not have much

significance over short time periods. Detecting stopped objects that start moving, and counting the flux of objects moving through the frame, can also be accomplished using this methodology.

One could also design the algorithm to focus only on movement in certain areas, e.g. on the street or on the station platform, where alertable activity is most likely to occur. Similarly, other areas of the image frame can be avoided, e.g train tracks, yards, driveways, buildings, etc., where maybe the activity of concern is not a problem, e.g. parking on one's own property, or where other activities that occur on the same time-scale may set off false alarms, e.g. trains may come and go at about the same rate as one would want to be concerned that bags or packages have been abandoned.

Though this challenge has not been undertaken by using the DMD method as of yet, the concept of examining the data collected in short snapshots of time in order to distinguish events that occur on longer time-scales is intriguing, and may answer the needs of certain applications, e.g. determining when dynamic gestures start and stop in recognition problems.

2.1.5 NLS Examples

The background/foreground subtraction in videos only explores the case when the slow dynamics of the system are stationary. In order to explore time-scale filtering on broadened set of circumstances, consider the nonlinear Schrödinger (NLS) equation

$$iu_t + \frac{1}{2}u_{xx} + |u|^2 u = 0, \tag{2.3}$$

and its initial condition for a soliton solution

$$u(x,0) = N\ \mathrm{sech}(x - x_0)e^{i\Omega x}, \tag{2.4}$$

where N is the soliton number, x_0 is the center-position offset, and Ω represents a center-frequency shift of the soliton. First, consider the case when $N = 1$, $x_0 = -10$, and $\Omega = \pi/2$, and where a random noise, modulated by a low amplitude, low

frequency oscillation $(\cos(t/4 + \phi_{\text{rand}})/20)$ in time, is added to the solution. This modulated background represents, in some sense, a proxy for an artificial background and/or corruption of the original solution.

The initial condition produces a single soliton that moves to the right at a constant velocity. The noisy solution, in the intervals $x \in [-20, 20]$ and $t \in [0, 4\pi]$, sampled evenly in space $(n = 512)$ and in time $(m = 25)$, is illustrated in Fig. 2.7. The process of reconstructing the solution with the DMD method is done without any extra dimensionality reduction: $\ell = m - 1 = 24$. The translation of the soliton over time makes the soliton a sparse structure within the noisy solution, while the slowly varying noise comprises the low-rank structure of the noisy solution.

Figure 2.7 shows that there is a Fourier frequency, labelled with a **1** in the bottom left panel of (a), associated with the DMD process that is located near the origin of the complex plane: $\omega_{\mathbf{1}} = -0.0432 + 0i$. This Fourier frequency corresponds with the DMD term that models the background noise, which, by construction, is nearly stationary in time. Surprisingly, this Fourier frequency has a slight negative real component and no imaginary component, as is depicted in the time dynamics plot (solid red and dashed magenta curves) in the bottom right panel of Fig. 2.7 (a). The remaining 23 DMD terms, which reconstruct the translating soliton, also attenuate slightly over time corresponding to having negative real components to their Fourier frequencies as well. Notice that the DMD basis function modes $\mathbf{\Phi}$ (See the middle left panel of Fig. 2.7 (a)) have very small amplitudes and do not seem to carry much information about the actual shape, size, direction, or speed of the solution. However, it is known that $\mathbf{x}_1 = \mathbf{\Phi b}$ (See Equ. (A.7)), which means that with the initial amplitudes $\mathbf{b}$ (See the middle right panel of Fig. 2.7 (a)), the size and shape of the solution at t_1 is well reconstructed. The solution is translated in time with information embedded in the Fourier time dynamics (See the bottom panels of Fig. 2.7 (a)).

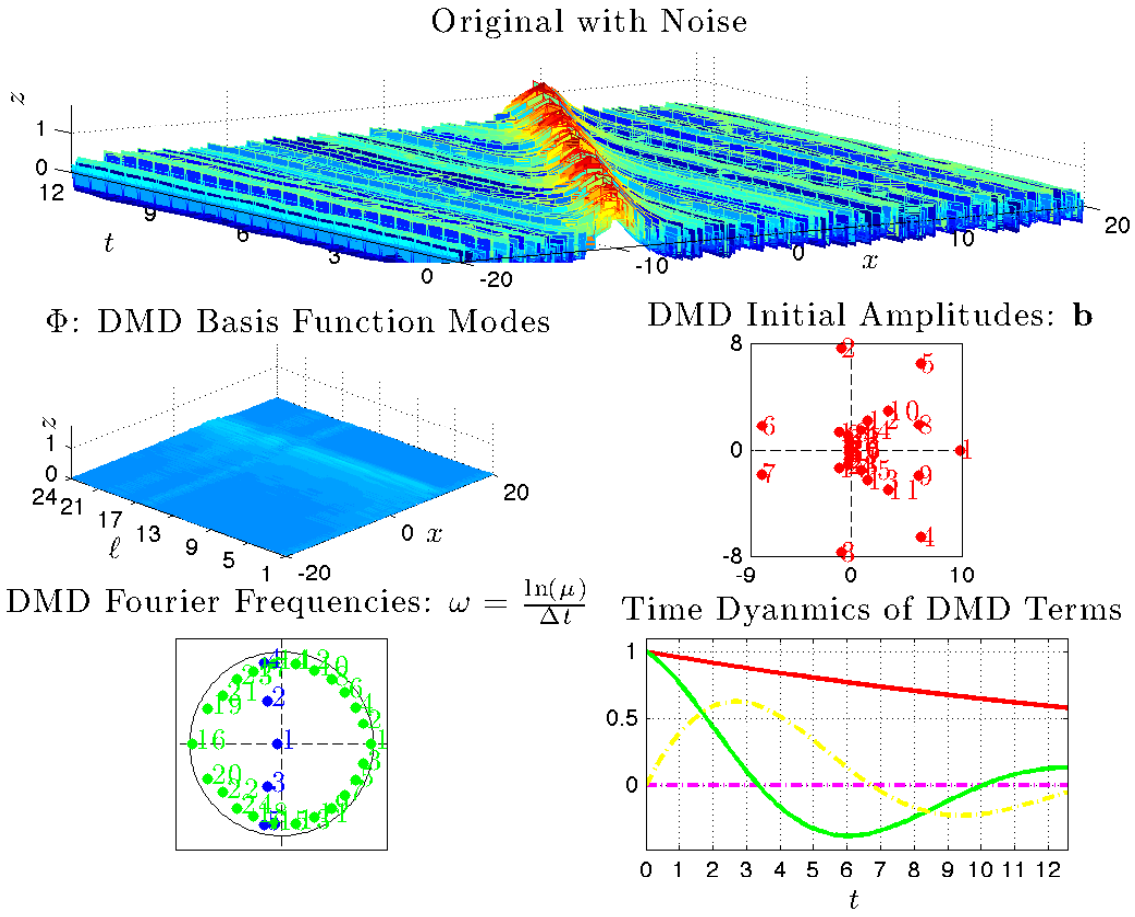
Next, consider the case when $N = 1$, $x_0 = 0$, and $\Omega = 0$, and where a temporal and spatial random noise is added to the solution. In this case, the single soliton is

stationary in time, and is therefore best represented as the low-rank structure of the system. Figure 2.8 illustrates the DMD separation for this solution and its added noise. Note that the error that accumulates in the DMD process is deposited in the last time step of the DMD reconstruction, leading to erroneous spikes as can be seen in the middle left panel of Fig. 2.8 (b). In spite of the fact that the noise is not very sparse overall, the DMD separation still works with excellent results.

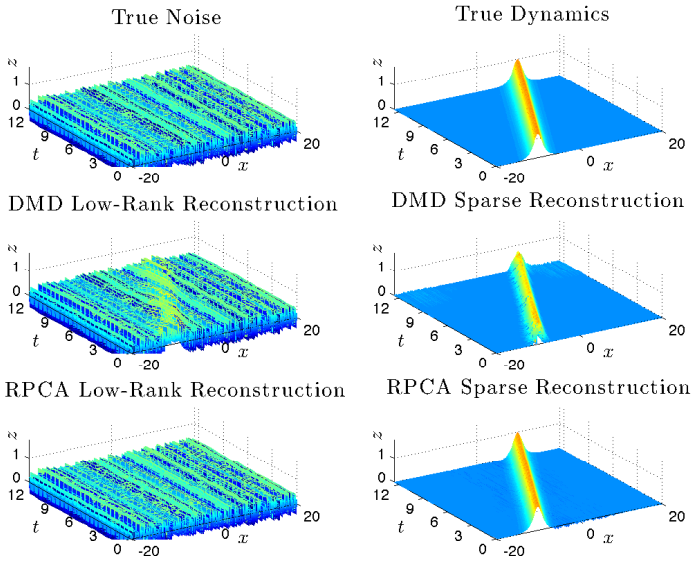
Finally, consider the more complicated case when $N = 2$, $x_0 = 0$, and $\Omega = 0$, which also has a, somewhat sparse, temporal and spatial random noise added to the solution. This is the case of a two soliton breather solution that is stationary in time; however, because the solution "breathes" in time, the low-rank component to the separation cannot be represented by just one DMD term. Figure 2.9 portrays the DMD separation for the two soliton solution and its added noise. Here, three DMD terms accurately reconstruct the true soliton solution from information embedded non-intuitively in the DMD basis function functions $\mathbf{\Phi}$, the initial amplitudes $\mathbf{b}$, and the DMD Fourier frequencies $\boldsymbol{\omega}$. Note that the three best DMD terms for reconstructing the solution without noise are not the three terms that corresponded to the three smallest Fourier frequencies in modulus ($|\omega_j|,\ j = 1, 2, \ldots, \ell = 24$), but were terms 1, 21, and 22, where the terms are listed in order of increasing magnitudes of the Fourier frequencies. Again, erroneous spikes can be seen in the last time step of the DMD low-rank component (See the middle left panel of Fig. 2.9 (b)). Over all, though, the DMD method separates the true solution from the added noise with a great deal of accuracy, indicating that this procedure has a good chance of working in more realistic settings.

The RPCA low-rank/sparse separations for figures 2.7, 2.8, and 2.9 are done perfectly correct. Because the exact solution was known for all 3 of these cases, an optimization routine was established, which found the best regularization parameter λ that made the most accurate separation possible. It should be noted that, for all three solitons, the best λ parameter was not near the suggested value given by Candès et

Figure 2.7: These figures illustrate the DMD process performed on a single soliton solution, with initial condition parameters $N = 1$, $x_0 = -10$, and $\Omega = \pi/2$ (2.4), to the NLS equation (2.3) with a random spatial noise slowly modulated in time. The original solution with the added background noise is shown at the top of (a). The DMD basis function modes $\mathbf{\Phi}$ (A.4) (middle left), the DMD initial amplitudes $\mathbf{b}$ (A.8) (middle right), the DMD eigenvalues $\boldsymbol{\mu}$ (A.3) in green with some of their corresponding Fourier frequencies $\boldsymbol{\omega}$ (A.5) in blue (bottom left), and the time dynamics of the best (real and imaginary components in green and yellow respectively) and worst (real and imaginary components in red and magenta respectively) DMD terms for reconstructing the solution without noise (bottom right) are all depicted above. The true background noise (top left) and the true soliton solution (top right) are shown in (b), along with their versions from the DMD (middle) and RPCA (bottom) low-rank/sparse separation methods. A single term of the DMD defines the background noise (low-rank structure), and the remaining 23 DMD terms reconstruct the soliton solution (sparse structure), for a total dimensionality of $\ell = 24$.

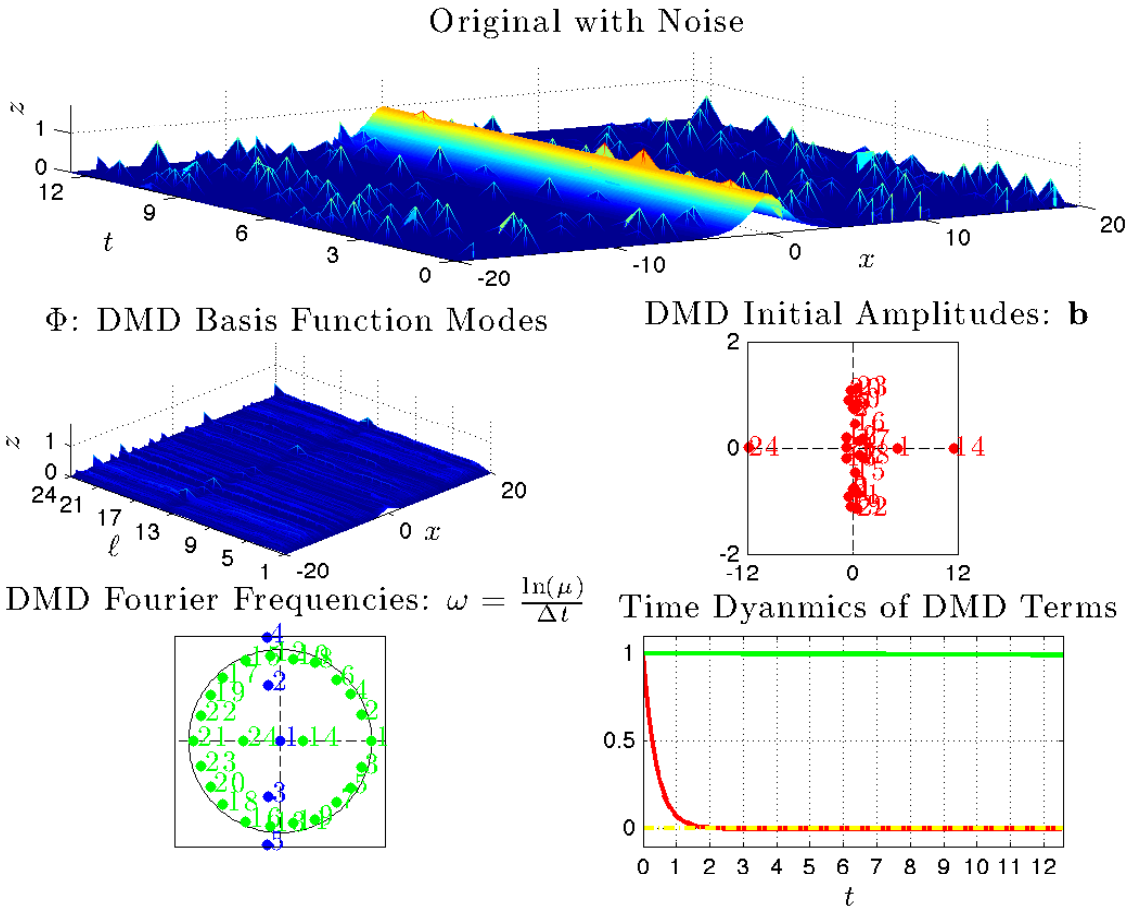


(a) DMD process.

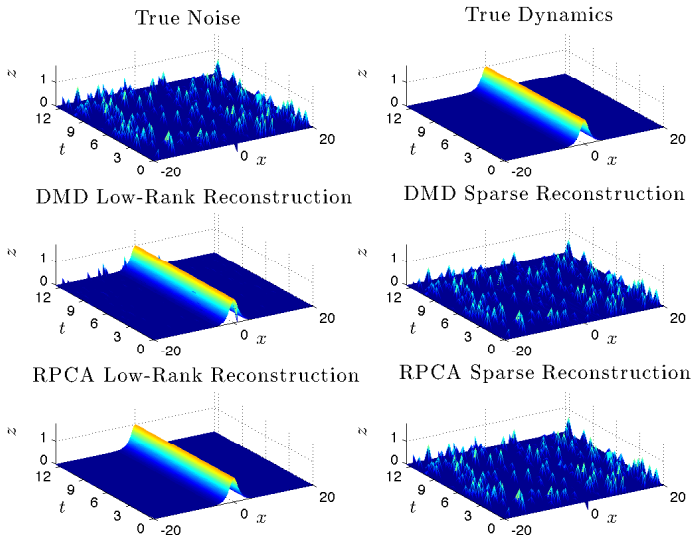


(b) RPCA and DMD comparison.

Figure 2.8: These figures illustrate the DMD process performed on a single soliton solution, with initial condition parameters $N = 1$, $x_0 = 0$, and $\Omega = 0$ (2.4), to the NLS equation (2.3) with a random temporal and spatial noise. The original solution with the added background noise is shown at the top of (a). The DMD basis function modes $\mathbf{\Phi}$ (A.4) (middle left), the DMD initial amplitudes $\mathbf{b}$ (A.8) (middle right), the DMD eigenvalues $\boldsymbol{\mu}$ (A.3) in green with some of their corresponding Fourier frequencies $\boldsymbol{\omega}$ (A.5) in blue (bottom left), and the time dynamics of the best (real and imaginary components in green and yellow respectively) and worst (real and imaginary components in red and magenta respectively) DMD terms for reconstructing the solution without noise (bottom right) are all depicted above. The true background noise (top left) and the true soliton solution (top right) are shown in (b), along with their versions from the DMD (middle) and RPCA (bottom) low-rank/sparse separation methods. A single term of the DMD defines the soliton (low-rank structure), and the remaining 23 DMD terms reconstruct the background noise (sparse structure), for a total dimensionality of $\ell = 24$.

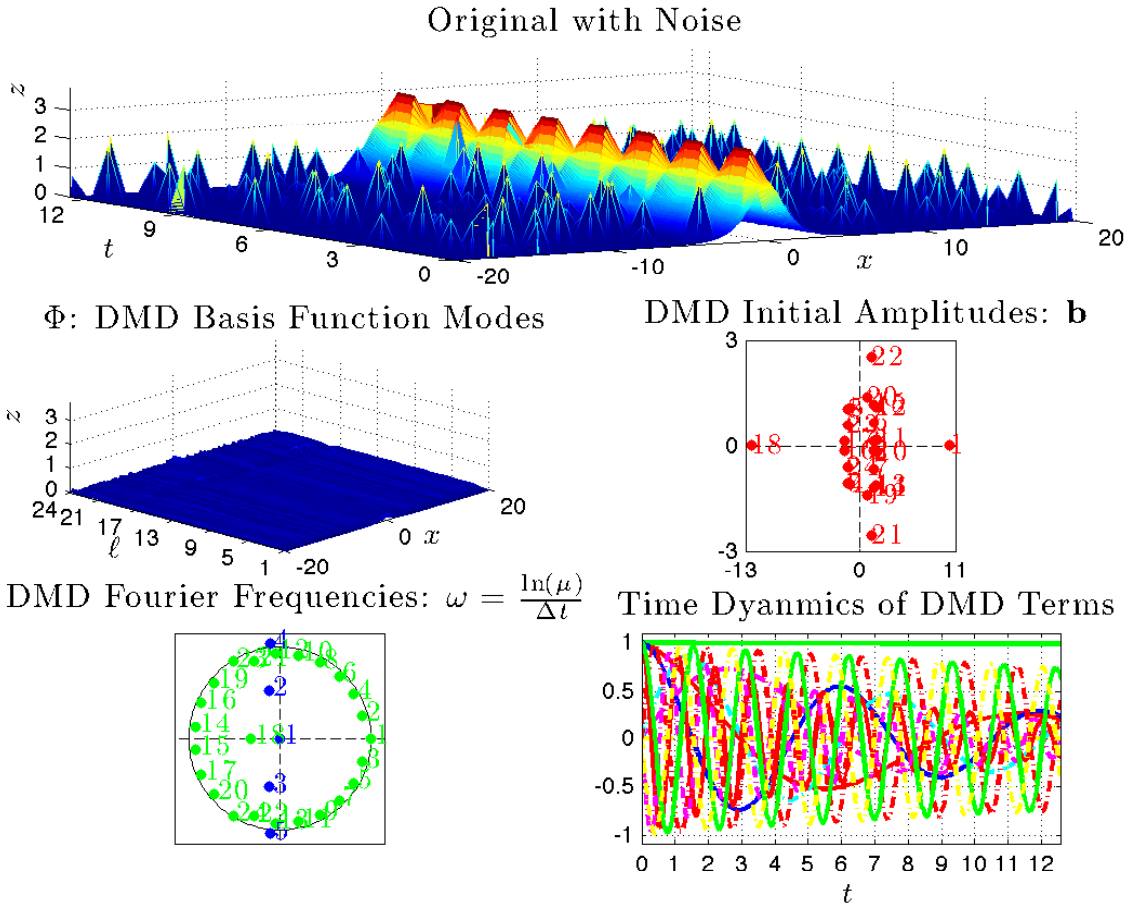


(a) DMD process.

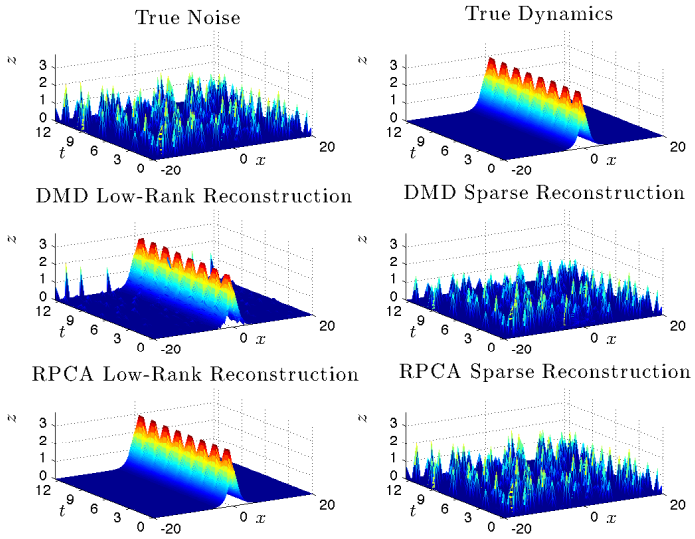


(b) RPCA and DMD comparison.

Figure 2.9: These figures illustrate the DMD process performed on a two soliton breather solution, with initial condition parameters $N = 2$, $x_0 = 0$, and $\Omega = 0$ (2.4), to the NLS equation (2.3) with a random temporal and spatial noise. The original solution with the added background noise is shown at the top of (a). The DMD basis function modes $\mathbf{\Phi}$ (A.4) (middle left), the DMD initial amplitudes $\mathbf{b}$ (A.8) (middle right), the DMD eigenvalues $\boldsymbol{\mu}$ (A.3) in green with some of their corresponding Fourier frequencies $\boldsymbol{\omega}$ (A.5) in blue (bottom left), and the time dynamics of the 3 best and 3 worst (real and imaginary components plotted separately) DMD terms for reconstructing the solution without noise (bottom right) are all depicted above. The true background noise (top left) and the true soliton solution (top right) are shown in (b), along with their versions from the DMD (middle) and RPCA (bottom) low-rank/sparse separation methods. Three terms of the DMD define the soliton (low-rank structure), and the remaining 21 DMD terms reconstruct the background noise (sparse structure), for a total dimensionality of $\ell = 24$.



(a) DMD process.



(b) RPCA and DMD comparison.

al. [9] in equation (A.2), where λ would be given by $\lambda = (\sqrt{\max(512, 25)})^{-1} \approx 0.0442$.

Figure 2.7, specifically in the middle right panel of (b), depicts the fact that the DMD method sometimes is able to capture the proper low-rank and sparse components to the given problem, but ends up reconstructing the solution with a lower amplitude (or, in the case of images, a lower pixel intensity) than the true solution (or the true image intensity). This fact has been seen in many other examples as well, though not presented here. This imperfect reconstruction may not be a significant problem for many applications, especially when the low-rank and sparse components do not overlap, e.g. background/foreground separation in videos, and so the DMD reconstruction can be replaced with the original data in the positions where the dynamics (or components of the solution) are detected.

2.2 Image Pre-Processing

The main purpose of image pre-processing is to develop a representation of the images that preserves both the intra-class similarity and the inter-class distinctions between gestures. This goal is achieved by a consistent normalization of all the images once the gesture has been properly segmented from its background, while maintaining, if not highlighting, the main features and shape of the articulated gestures.

Quality gesture detection and segmentation in mist of complex and cluttered backgrounds the is still an open area of active research. Here, gesture detection and extraction from its background are treated as separate processes from that of pre-processing, though some have combined these steps in other recognition methodologies. In this way, any gesture localization and segmentation algorithm can be added to the front end of the procedures and techniques developed here in order to a get a complete gesture recognition scheme.

2.2.1 Pre-Processing Steps

The image processing can be broken down into the following steps that do not necessarily need to adhere to the given ordering:

(1) *grayscale conversion* - changing raw images from color to grayscale,

(2) *image resizing* - the image is down-sampled, meaning that it is resized in order to reduce the image resolution, usually to a square of the size $n \times n$, where n is an integer,

(3) *intensity normalization* - the pixel intensities are normalized such that the brightest pixel is set to be pure white,

(4) *segmentation/background removal* - the gesture is selected from its background, which is set to be pure black,

(5) *cropping* - the elimination of the excess background pixels around the gesture by cutting them out, which also effects the size (scale) of the gesture within its frame,

(6) *arm/wrist removal* - the arm and wrist regions of the hand are removed, being cut-off at the bottom of the palm and set to pure black,

(7) *centering* - the resulting isolated gesture is set in the centroid of the image frame, where the gesture center is calculated by finding the average pixel position, weighted by the pixel intensities of the gesture,

(8) *orientation detection* - the principal direction of the gesture is determined by the best-fit line that satisfies the linear least squares of the gesture's pixel positions, again weighted by the pixel intensities, and

(9) *rotation* - the angle of rotation is calculated based off of the angle that the best-fit line makes with a horizontal line through the center of the gesture, and the gesture is rotated by a simple linear transformation so that the gesture points in a consistent direction within the frame; consistent to other like-gestures of the same class.

The arm/wrist removal step is, of course, specific to hand gestures, but may have parallels to pre-processing for other types of gestures, e..g cropping out shoulders/necks for facial expressions, or cropping out face regions above and below the eyes when recognizing eye movements. An example pre-processing scheme for hand gestures is depicted in Fig. 2.10, which shows the progression from the original color image to a final normalized gesture image (Specifically, see images (a)-(g) in Fig. 2.10).

Erroneous black spots (See the top left and bottom right panels of Fig. A.2), which appear due to the rotation algorithm and rounding to the nearest pixel position, need to be filled in within the region of the articulated gesture. One could use morphological operations like dilation to handle these erroneous black spots [34]. However one needs to be careful at low image resolutions where dilation may start to erase the prominence of the gesture's features and shape. Even when one follows the dilation with an erosion operator in an attempt to restore the original features of the gesture, the subsequent erosion only rarely can restore loss of shape or feature to very low resolution images.

Erosion morphological operations could also be implemented in attempt to suppress and eliminate background objects. However, these operations are best used at high image resolutions because, at low resolutions, they are only mildly successful in segmenting and removing the background. Plus, they suffer from the same problem of erasing the prominence of the gesture's features and shape as was explained previously.

One clever way to eliminate spurious, non-black, background pixels that sometimes appear away from the gesture, and to eliminate the erroneous black spots within the

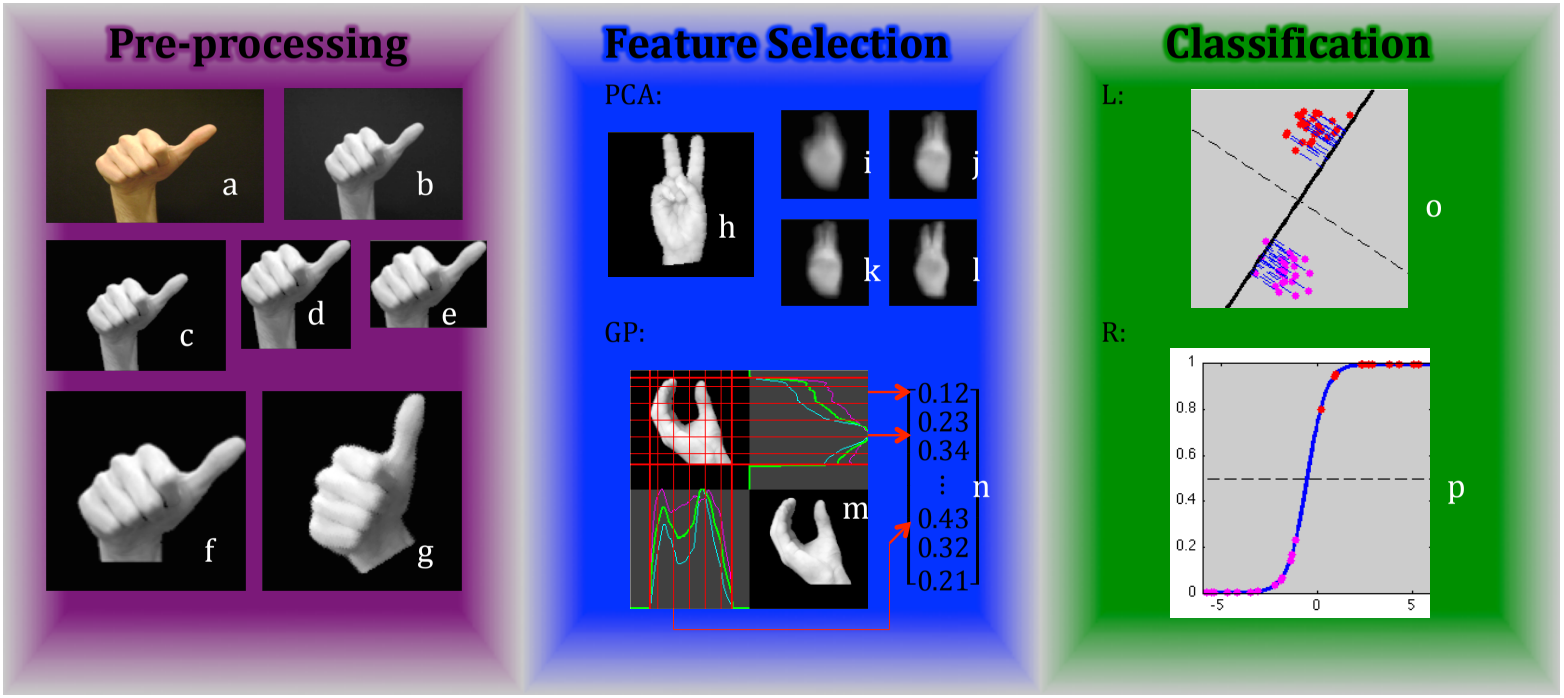
gesture after rotation, is to complete as many of the pre-processing steps as possible, with consideration to the computational time, before down-sampling the image. In the down-sampling process, the interpolation can often eliminate, or at least minimize the intensity of, rogue pixels since their influence on the interpolation is minimal. A thresholding criterium may handle any remaining pixel intensity associated with the background.

Algorithms for finding the principal direction can be confounded by a variety of gesture postures, such as with "o"-shaped hand articulations, which do not have an obvious principal direction. Also, sometimes features outside of the gesture can help indicate the proper orientation of the gesture, e.g. the arm/wrist region for hand gestures or neck and shoulders region for facial expressions. If these outside features get masked and/or separated from the image frame by dark clothing, sleeves, jewelry, watches, or bracelets, then automatic orientation detection algorithms may fail. In these cases the goal would be to get a consistent rotation of like-gestures to a single, suitable orientation.

An advantage of using grayscale images over color images is that one does not need to consider the various skin colors of the general population. Also, low lighting levels can be normalized out in the pre-processing, and storing and manipulating grayscale images is computationally more efficient since less information is being stored than with color images.

Note that when doing gesture recognition, there are a variety of ways in which one can optimize the pre-processing in order to favor a certain image resolution. For instance, there are many parameters that control the automatic segmentation and background removal, which can be tailored for the best results at the desired image resolution. When one creates the training sets for real applications, it is best to choose and label the training set images after all the pre-processing and resizing has been accomplished, so that pre-processing errors do not skew the purity of the training set. Usually one desires only the best articulated gestures of each class to be included in

Figure 2.10: This diagram exemplifies the recognition process for hand gestures starting with an example pre-processing progression (left panel), which includes: (a) original image, (b) grayscale conversion and intensity normalization, (c) segmentation and background removal, (d) crop, (e) arm/wrist removal, (f) downsample and center, and (g) rotation. Two feature selection methods are depicted (middle panel): principal component analysis (PCA) and generalized projections (GP). Using PCA, the image (h) can be reconstructed using (i) 1 feature, (j) 3 features, (k) 5 features, and (l) 7 features. Using GP, a feature vector (n) is formed by extracting horizontally and vertically a discrete number of the projection values (m). The classification methods used are (right panel) linear discrimination analysis (L) and logistic regression (R). Feature vectors from two different classes are illustrated by red and magenta dots, respectively. The optimal threshold (black dotted line) for classification purposes is determined in the training process.


Pre-processing
a
b
c
d
e
f
g
Feature Selection
PCA:
h
i
j
k
l
GP:
m
0.12
0.23
0.34
n
0.43
0.32
0.21
Classification
L:
o
R:
p
1
0.8
0.6
0.4
0.2
0
-5
0
5

the training set, thus labeling at a specific image resolution helps ensure the quality of the training set for that resolution.

Gesture occlusion is a pernicious problem that may need to be addressed for certain applications. Specifically, with hand gestures, the dexterity of a human hand, wrist, and arm allows for many instances when fingers are hidden behind the palm or other fingers, greatly complicating the recognition process. However, occlusion issues are dismissed in the following analyses as being a problem tangential to the primary objectives and goals presented here, and so no steps are taken in the pre-processing to mitigate or overcome this effect.

2.2.2 Segmentation and Background Removal

Static segmentation of gestures that can remove the backgrounds in images is still a difficult problem in the recognition field. Many researchers avoid the issue by controlling or idealizing the backgrounds in their datasets of gesture images; this also done here in this dissertation. Some gesture databases with cluttered backgrounds, like the RWTH German Fingerspelling Database [35, 36], have been studied by others. However, some of the established gesture databases that are available on-line, such as the Cambridge hand gesture database [37] or the Sébastien Marcel datasets [38], also avoid the background subtraction issue by providing gesture sets with uniform backgrounds with well-centered and pre-oriented gestures. Indeed, these datasets remove almost all pre-processing from the gesture recognition process. In contrast, the datasets used here have fairly uniform, dark backgrounds under different lighting conditions, but leave the gesture image noise and variances in rotation, scale, skew and translation to be handled by the pre-processing algorithms (See Fig. 2.11). Even with these significant liberties taken, a gesture identification and background removal scheme is still needed.

In order to identify the gesture from the dark background, a distribution of the grayscale pixel intensities of each image can be made. In a histogram of the pixel

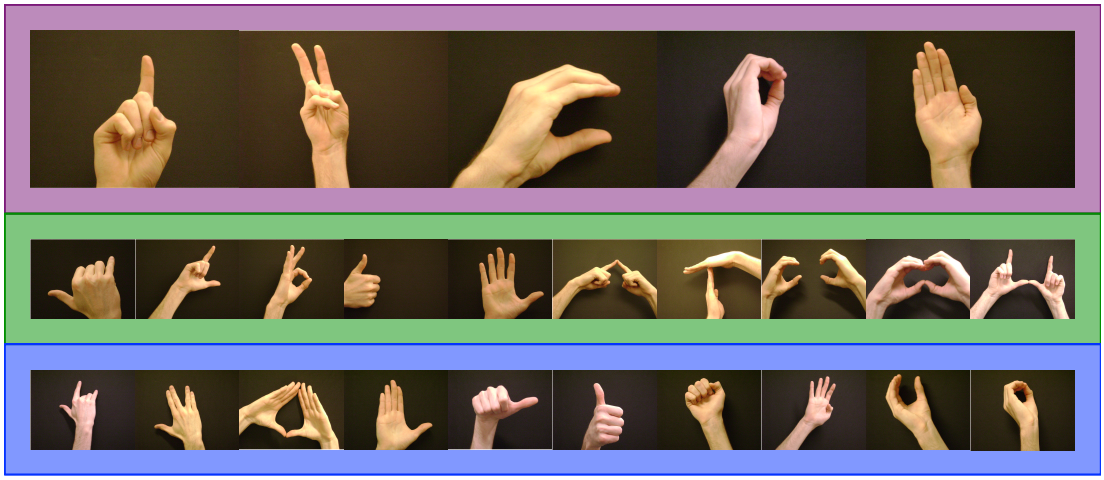


Figure 2.11: These twenty-five hand images comprise a gesture lexicon that is implemented for analysis in this chapter. The first row of five gestures were considered previously for a potential lexicon, which can be used as a computer mouse control application (See Fig. 1.1). Note the various lighting conditions, amount of arm/wrist region present in the frame, translations, scales, rotations, and slight occlusions contained within the dataset. There are even similar gestures that are articulated slightly differently in order to test the robustness of the recognition process. Other images in this dataset come from other people, sometimes with long sleeves of various colors and patterns.

intensity distribution, the dark background appears as a high frequency peak of low pixel brightness values. A threshold can be automatically determined, based on the position of the minimum that separates these low intensity pixels from the brighter pixels of the gesture. Then, all the pixels that are darker (of lower intensity) than the threshold can reliably be considered to be background, and are made to be pure black. This effectively eliminates the background and detects the gesture.

2.2.3 Arm/Wrist Removal

Creating an automatic algorithm for detecting and removing arm and wrist regions from an image can be quite difficult, yet makes an important impact on the performance of gesture recognition algorithms. Since this document focuses on hand images that fill most of the frame of the image, one can safely deduce that the arm/wrist must exit the frame of the image. Assuming that the background removal is done well, sleeves or any other article of clothing or apparel may naturally cut the hand image off near the bottom of the palm or top of the wrist, as is desired. Otherwise, one can find the arm/wrist region on the frame of the image and follow it inward towards the hand, erasing the rows/columns of pixels that belong to the arm/wrist region as one goes. If one is viewing the hand from the front or back (not the side) there is a distinguishing feature that separates the arm from the hand; namely, a sudden increase in width of the arm/wrist/hand region as one transitions from the arm to the hand. Using this distinctive feature and tracking the arm/wrist widths as one moves toward the image center, an automatic algorithm for removing the arm/wrist region is created. Since the examples used in this chapter focus on hand gestures that are only viewed from the front or back, removing arm/wrist regions from side-viewed hands is rendered moot. As with many automatic processes, there is a trade-off between robustness and computational processing time.

2.3 Feature Selection

There are many types of features that can be extracted from gesture images. In order to achieve high accuracies in gesture recognition, the features that are extracted need to be consistent within each gesture class of images but different between classes. It would impossible to create an exhaustive list of all the possible values associated with images that could be used as features. In this chapter, principal component analysis (PCA), generalized projections (GP), and image moment (IM) methods, including

some variations of these algorithms, are implemented as feature selection algorithms in the analysis that follows. More detailed explanations of these feature extraction methods are delineated in the appendix of this work, specifically sections A.2.1, A.2.2, and A.2.3. All of these methods have been used previously in the context of gesture recognition [24, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51], though some their variations are novel to this work.

Feature selection does not necessarily have to proceed the gesture segmentation and preprocessing steps. For instance, it has been found that the aspect ratio of the gesture can be an important feature, especially in the context of severe image downsampling, where similar gesture postures, e.g. a frontal view of an open palm hand with the fingers close together vs. a closed fist, can become confused at low image resolutions. This aspect ratio can best be extracted as a posture feature before the gesture image is resized to be square. Additionally, there are many gesture detection algorithms that find features within the image in order to detect the gesture, and these features can also be used for the posture recognition [30, 31].

Gesture Representation in Feature Space

Most gesture recognition algorithms eventually represent the gestures as a sequence of features, which can be thought of as points in a feature space (See Fig. 2.12). This feature space provides a great deal of information about how similar or different the computer views the gestures. However, it is very common that most gesture recognition algorithms extract many features in order to better their recognition performance, and this corresponds to having high-dimensional feature spaces, which are usually impossible to visualize in a figure. The high-dimensional nature of feature spaces invokes many of the same issues that accompany the well-known *curse of dimensionality* [52] found in the computer vision, statistics, and geometry fields. This curse is basically defined by the fact that high-dimensional data becomes intractable to systematically work with as the number of dimensions increases, especially when

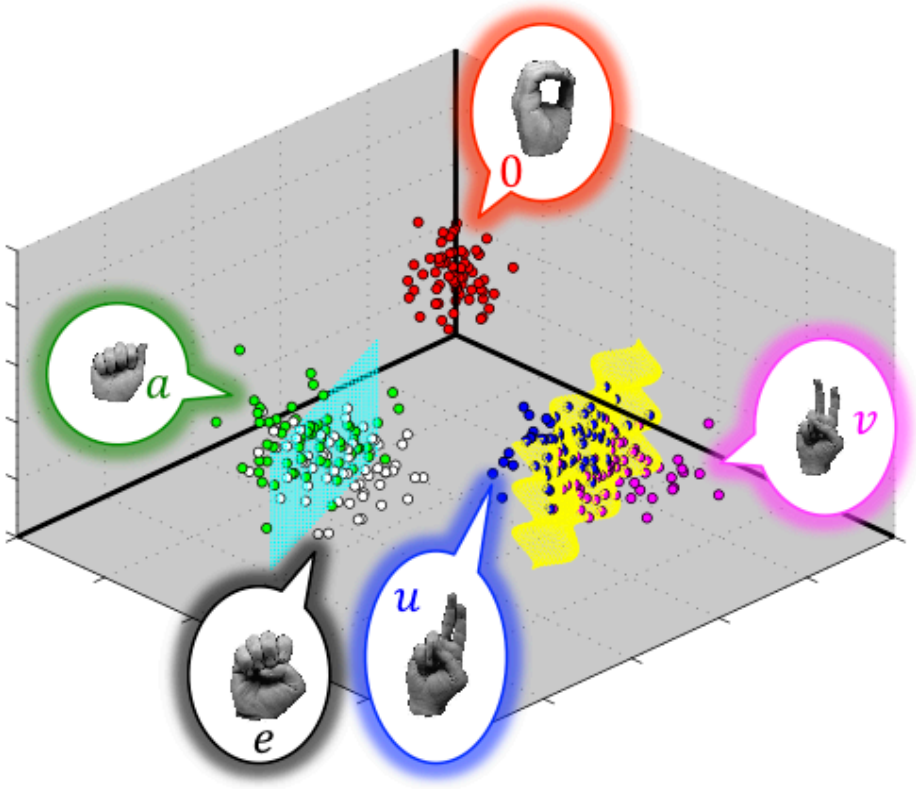


Figure 2.12: This figure exemplifies how gestures may appear in feature space. Gesture classes **a** (green) and **e** (white), which overlap in feature space, are separated by a linear separatrix (cyan) that attempts to distinguish the classes in some optimal sense. Gesture classes **u** (blue) and **v** (magenta), which do not overlap in feature space, are separated by a nonlinear separatrix (yellow). Gesture class **0** is well-separated from the other four classes in feature space.

one does not have a preconceived notion as to how the data ought to render itself.

Figure 2.12 provides a visual example of what a feature space may look like; in this case, it is in 3 dimensions, which corresponds to having three features extracted from every image. A common way to extract features is through, for instance, a principal

component analysis, but there are many other techniques available. Note that gesture class **0** in Fig. 2.12 is well-separated from the other four classes in feature space. Also, the feature points of this class are well-clustered such that there does not appear to be any images with feature points that are extreme outliers to the other images of this gesture class. Well-clustered feature points are a result of extracting consistent feature values from like-gestures of the same class. Clearly, this is a very desirable circumstance for how the feature points are positioned in feature space.

Gesture classes **a** and **e** illustrate a common occurrence with many gestures that look similar in that these classes overlap with one another in feature space. Obviously, similar gestures are likely to produce similar feature values, and therefore will lie near one another in feature space. Also, consider how classes **a** and **e** are not well-clustered, and have many outliers to their apparent centers. A linear separatrix, defined by $y = (2/5)x + 9/5$, can be found by the computer using statistical learning techniques in order to distinguish these two classes in some optimal sense. However, it is evident that there is not a perfect way of separating these classes without likely *over-fitting* the data so that the separation remains general to any new data that may be used in the recognition application.

Gesture classes **u** and **v** of Fig. 2.12 exemplify another possibility in the recognition process in that the classes are separated from one another in feature space, but the boundary between the two class regions is defined by some nonlinear function. In this case, a nonlinear separatrix, defined by $y = (2/5)x + \cos(4x)/4 + \cos(4z)/4 + 6$, distinguishes the two classes in some optimal sense. One can easily image that without enough data points, assuming they are even available, and/or without any *a priori* idea as to how these classes ought to be separated, a computer algorithm may struggle to learn the nonlinear separatrix, although this may be possible with, for instance, adaptive boosting methods [53, 54]. Even still, if the computer is able to learn data that is separated in some complex, nonlinear manner, one would still have to question whether over-fitting has occurred, and thus if the learned separation is generalizable.

It is unlikely for most gesture recognition problems that one will be able to get any useful *a priori* information as to how the features of any class will appear in feature space relative to another gesture class, especially given the high-dimensional nature of most feature spaces.

Therefore, it would be ideal if all gesture classes were comprised of well-clustered feature points, and if they were all well-separated from one another into distinct regions of the feature space. One of the best ways of controlling how the gestures are rendered in feature space is by choosing excellent features in the first place. A quality feature selection algorithm will find features that remain consistent for like-gestures and yet are very distinct for different gestures.

Best Features

A common practice in the recognition field is to extract as many features as possible and allow the statistical learning algorithm to diminish the relevance of the features, usually by controlling weightings, which do not strongly effect the decision making process of the classification. However, in order to determine the best gestures, this practice may be detrimental because weak features are either inconsistent (noisy) for like-gestures of the same class or they are consistent across different classes. This inherently sabotages the chances for like-gestures to be well-clustered in feature space and yet well-separated from different gesture classes. Additionally, by including more features, the dimension of the feature space is increased, making it harder for both the human and computer to understand how the different gesture classes are separated in feature space. Thus, in terms of searching for the best features, it is better to filter out the weak features from the analysis, and only focus on the more effective features.

Currently, there are some tests designed to evaluate the efficacy of a given feature on the entire recognition process [23]. Some examples include the Fisher score [55], the Generalized Fisher score [56], mutual information [57], ReliefF [58], the Laplacian score [59], the Hilbert Schmidt Independence Criterion [60], the Trace Ratio Crite-

rion [61], the Multi-Layer Perceptron Sensitivity Method [62], and Principal Feature Analysis [63].

Another variable selection routine that can help determine which features have the greatest impact on the recognition performance is what is termed here as the *feature selection weakness* (FSW). This measure assesses how close points of the same class/type/variety are to one another against how far points of different classes/types/varieties are separated from one another. This is similar to the Fisher score. Also like the Fisher score, FSW treats the features independently and not in combination with other features.

Assume that there are L features extracted from each image and M gestures in total, there being N_m images of the m^{th} gesture, where $m \in \{1, 2, \ldots, M\}$. Therefore, the total number of images in the dataset is given by $\sum_{m=1}^{M} N_m$. Let X^ℓ be a data structure which contains all of the feature values for feature ℓ, where $\ell \in \{1, 2, \ldots, L\}$, such that X^ℓ has M columns, and the m^{th} column has N_m rows. The element in the i^{th} row and m^{th} column of this data structure will be denoted by X_{im}^ℓ. The average among the feature values for feature ℓ and for gesture class m is given by $\mu_m^\ell = \frac{1}{N_m} \sum_{i=1}^{N_m} X_{im}^\ell$. The mean of these averages across all gesture classes is given by $\bar{\mu}^\ell = \frac{1}{M} \sum_{m=1}^{M} \mu_m^\ell$. The variance among the features values for feature ℓ and for gesture class m is given by $(\sigma^2)_m^\ell = \frac{1}{N_m - 1} \sum_{i=1}^{N_m} \left(X_{im}^\ell - \mu_m^\ell\right)^2$. With this notation, the FSW for the ℓ^{th} feature can be defined as

$$\text{FSW}(\ell) = \frac{\frac{1}{M} \sum_{m=1}^{M} \left[(\sigma^2)_m^\ell\right]}{\frac{1}{M-1} \sum_{m=1}^{M} \left(\mu_m^\ell - \bar{\mu}^\ell\right)^2}.$$

Upon closer inspection, the numerator of the FSW value is the mean, across all classes, of the variances in each class. And the denominator of the FSW value is the variance, across all classes, of the means in each class. If the variance in the feature values within any given class is small ($(\sigma^2)_m^\ell \sim$ small), then it means that feature ℓ produces consistent values for each class, which was already noted to be a beneficial attribute for the recognition process. When there is large variation between

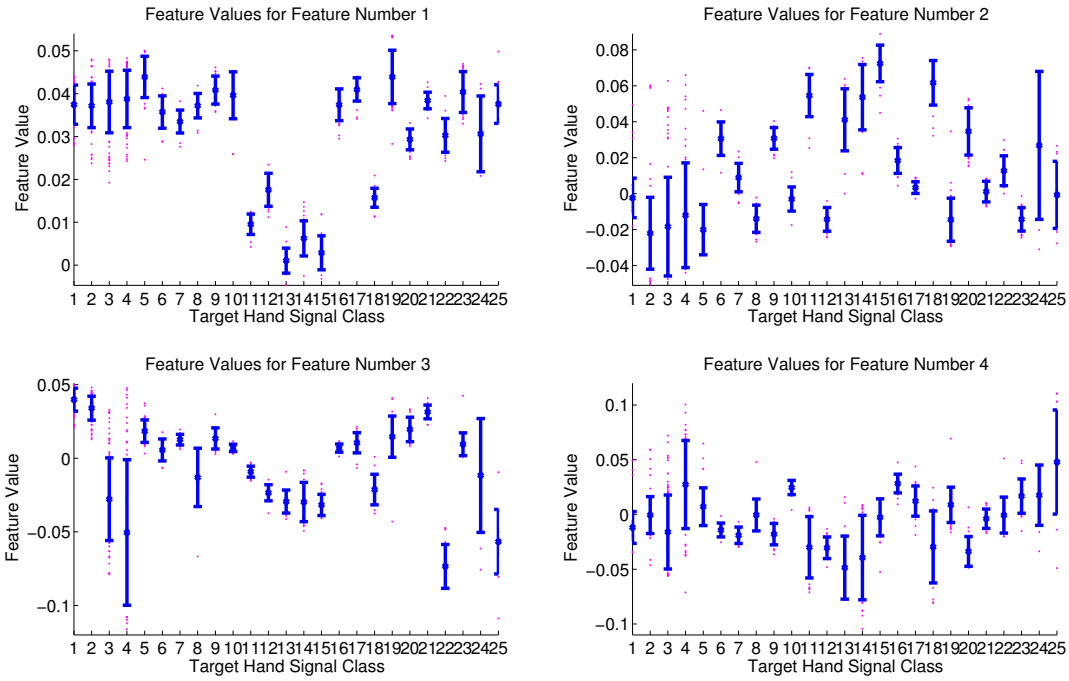


Figure 2.13: Here are the feature values for the first 4 features of a recognition problem with 25 gesture classes. The features are extracted using the PCA method. The average and one standard deviation spreads (blue) of the feature values (magenta) for each class are plotted. Note the smaller spreads in the feature values of feature 1 as compared to feature 4, which has larger error bars. Also, note how, for feature 1, some classes $\{11-15, 18\}$ are well-distinguished from the rest of the classes. But, for feature 4, the classes are not as well-separated. These attributes are reflected in the FSW values shown in Fig. 2.14.

the average feature values in each class (μ_m^ℓ), then these classes are well-separated by feature ℓ. Therefore,

$$\text{FSW}(\ell) = \frac{\text{small} = \overset{\text{in-class}}{\text{clustering}} = \overset{\text{in-class}}{\text{consistency}}}{\text{large} = \underset{\text{between-class}}{\text{separation}} = \underset{\text{between-class}}{\text{distinction}}} = (\text{small}).$$

Figures 2.13 and 2.14 illustrate how FSW values relate to the actual feature value

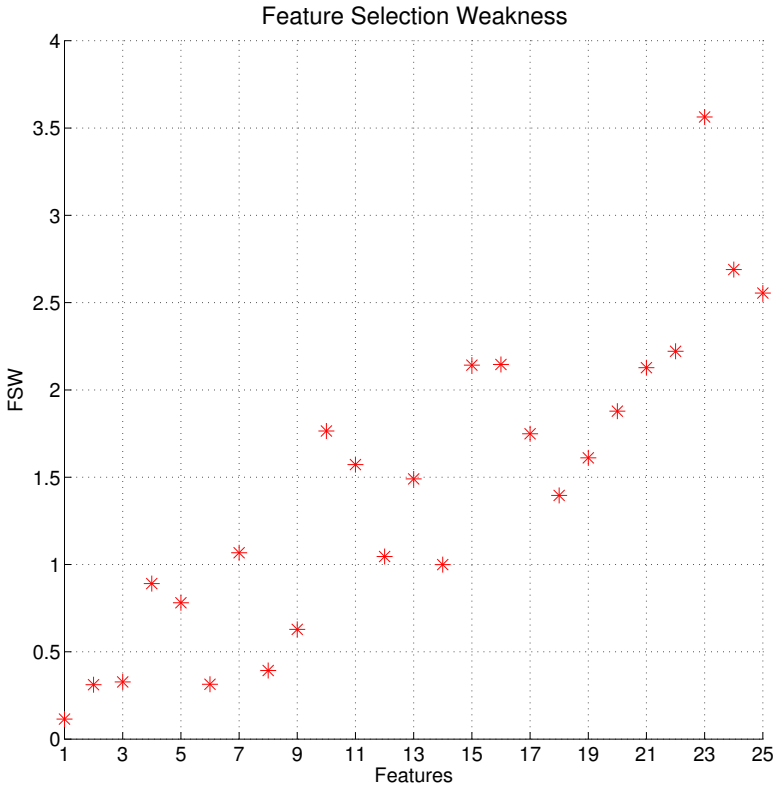


Figure 2.14: Here the FSW values are plotted for the 25 features of a recognition problem that uses PCA feature selection. As would be expected for PCA features, the best features are extracted first, and then the features tend to get progressively weaker as more are extracted. The first 4 FSW values come from the feature values plotted in Fig. 2.13.

data collected from a feature selection algorithm.

Now it is clear that if a given feature (ℓ) receives a large FSW value, then this is a weak feature; hence the "weakness" in the term *feature selection weakness*. The contrapositive is that the better (stronger) features will receive smaller FSW values. There is no definite border between "small" and "large" FSW values; rather FSW values among all the classes ought to be compared. The FSW value must be non-

negative, and is ideally zero. This measure can be used to evaluate the efficacy of the features in the recognition process, and even to remove the weakest features.

2.4 Statistical Learning & Classification

Extracted features are put to use in statistical learning algorithms that can find the best way to separate labeled gesture images so that future gestures can be classified according to their type. These algorithms require an optimization routine to learn the best parameters to some function that draws upon feature data to distinguish between gesture classes. Often, a training set of gestures is used for the learning phase, and for setting thresholds, which help differentiate between gesture classes, on the output function values of the statistical learning algorithm. After the best parameters have been found, classification routines compare the function output value of the statistical learning algorithm to the threshold so that a decision can be made as to what class the given gesture belongs.

Two standard methods are applied here: linear discriminant analysis (LDA) [55] and logistic regression [64], which are described in the appendix sections A.2.4 and A.2.5, respectively. Other, more sophisticated, methods exist and are well known, such as neural networks [65], support vector machines (SVM) [66], adaptive boosting [67, 68, 69], Hidden Markov Models (HMM) [70, 71, 72], and conditional random fields (CRF) [73], and these methods have been shown to perform well even in real-time scenarios.

In most gesture recognition problems there are multiple classes present. In the examples of this chapter, two classification styles are considered: *pairwise testing* and a *one-versus-the-rest strategy*. Pairwise classification checks every possible pairing of classes, deciding in which class a gesture is most likely to be. A one-versus-the-rest classification style checks each individual class against all other classes, which are clumped together and treated as a single class. Pairwise testing has the advantage of having a very high confidence level that a gesture is correctly labeled in the correct

class, but does not scale well with the number of classes and is prone to exclude gestures from their proper class. The one-versus-the-rest strategy scales better with the number of classes, but struggles to make clear distinctions when there are similar gesture classes present in the recognition problem. This dissertation addresses this issue as more of a matter of finding better pre-processing techniques, or even better features, in order to distinguish between similar classes, over improving the classification scheme.

A tree-based hierarchy classification [74], which narrows the search algorithm and eliminates the need for comparing all pairs of classes, can improve the scalability of having many gesture classes. But, only a few gesture classes are used in the analysis that follows, so a tree-based hierarchy classification style is not implemented.

2.5 Hand Gesture Application

In order to highlight the overall influence of pre-processing in the gesture recognition process, consider a hand gesture recognition problem that uses a variety of techniques at different image resolutions in order to discover what method combinations produce the best results. Only five gesture classes are explored in this study (See Fig. 1.1, and the top row of hand gestures in Fig. 2.11), with images mostly taken from two subjects, and with all of the 189 hand images being well articulated, with similar sizes (a consistent scale), similar orientations within each gesture class (less than ± 10 deg rotation variance), and with little to no extra arm/wrist region showing in the image frame. There are 38 class 1 images (one-click), 48 class 2 images (double-click), 29 class 3 images (scroll-up), 45 class 4 images (scroll-down), and 29 class 5 images (move cursor). No outside gestures are inserted into the image set in this study. The raw images were gathered at several different times, with slightly different lighting conditions. This idealized dataset is amassed in order to show that proper pre-processing can still have profound effects on the results. Indeed, without pre-processing, feature selection and classification schemes perform quite poorly (approximately 40%

accuracy).

In this study two different pre-processing methods are employed, which will be called *reduced pre-processing* (RPP) and *cropped pre-processing* (CPP). Both of these methods start by reading in a raw color image. The RPP method completes the pre-processing steps mentioned in the section 2.2.1 in the same order as they are listed, with exception of not completing steps (5) and (6), i.e. the cropping and the arm/wrist removal. The CPP method crops the unnecessary spaces above and below the hand region, and makes sure the hand resides on the center with wrist against the bottom and highest finger tip touching the top. CPP does some extra work to ensure that the background is completely removed from the image by removing any extraneous, isolated bright dots in the image frame. Images are not resized into a square $n \times n$ image until after the background removal and cropping has been completed. Then CPP centers and rotates the image as is done in steps (7)-(9). Linear interpolation is then used to get the pixel value of the rotated hand. Finally, the CPP procedure removes the arm/wrist region from the hand image. Because CPP does more pre-processing, and does it in a more perceptive and insightful way, one should expect the CPP to perform significantly better.

For this hand gesture example, only three feature selection methods are employed, namely, the traditional PCA method, generalized projections, and derivatives of generalized projections. These feature selection methods extract features that are distinct enough from one another so as to allow one to determine if one method is significantly advantageous over the others. Moreover, the fact that these methods lack some of the invariance properties (scale, translation, rotation, and skew), allows for a better investigation of the pre-processing effects on the recognition process.

The two pre-processing and the three feature selection schemes are used in combination with the LDA and logistic regression statistical learning and classification methods. Both the pairwise and the one-versus-the-rest classification styles are also implemented in combination with the other strategies. Table 2.1 provides the acronyms

for the various methods that will be implemented.

In order to best measure the error rates in properly recognizing the hand gestures, 100 rounds of cross-validation testing is completed for all of the method combinations found in Table 2.1. In such tests, the training and testing sets are randomly chosen from the entire set of 189 images, ensuring that each gesture class has at least a few images in the training set. The target is to have the training set size be about 20% of the entire image set size. In each round, the images are randomly reshuffled, and new training and testing sets are obtained. After 100 rounds of testing, the final error rate is calculated from the average error rates from all the rounds.

Figures 2.15 and 2.16 summarize the average error rate results. Associated with each line on the graph are three acronyms from Table 2.1 that highlight the feature selection, classification method and classification style. The resolution is the number of

Feature Selection Method	
S/PCA	singular value decomposition / principal component analysis
2DPCA	two-dimensional principal component analysis
GP	generalized projections
DGP	derivative of generalized projections
CGP	circular generalized projections
IM	image moments / invariant moments
Classification Methods	
R	logistic regression
L	linear discrimination analysis
Classification Styles	
P	pairwise
OR	one versus the rest

Table 2.1: Acronyms associated with feature selection, classification method and classification style.

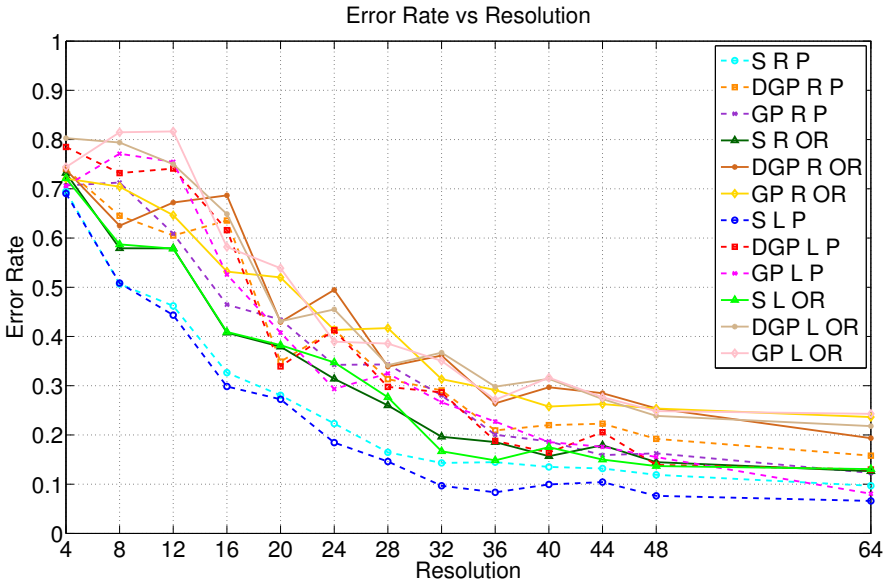


Figure 2.15: The average error rate of various recognition algorithms as a function of the resolution of hand images using RPP pre-processing. Compare the trends of these plots with those of Figure 2.16, which only differ because another pre-processing scheme is implemented.

pixels in the x- or y-direction of the image, which is square. Figure 2.15 illustrates the use of the RPP pre-processing method while Fig. 2.16 shows the CPP pre-processing technique. Fundamental to these graphs is the accuracy of the gesture recognition as a function of the resolution of the images. A lower image resolution will guarantee a faster, real-time algorithm. Figure 2.15 starts with error rates between about 70% and 80% at the 4×4 pixel resolution and achieves about 7% to 25% error rates at the 64×64 pixel resolution. Figure 2.16 fairly consistently achieves error rates between about 1% and 20% at all the pixel resolutions. For well resolved images (64×64 pixel resolution), the difference in the average accuracy of the techniques shifts from $\approx$85% (RPP) to $\approx$95% (CPP). The difference is much more pronounced for lower res-

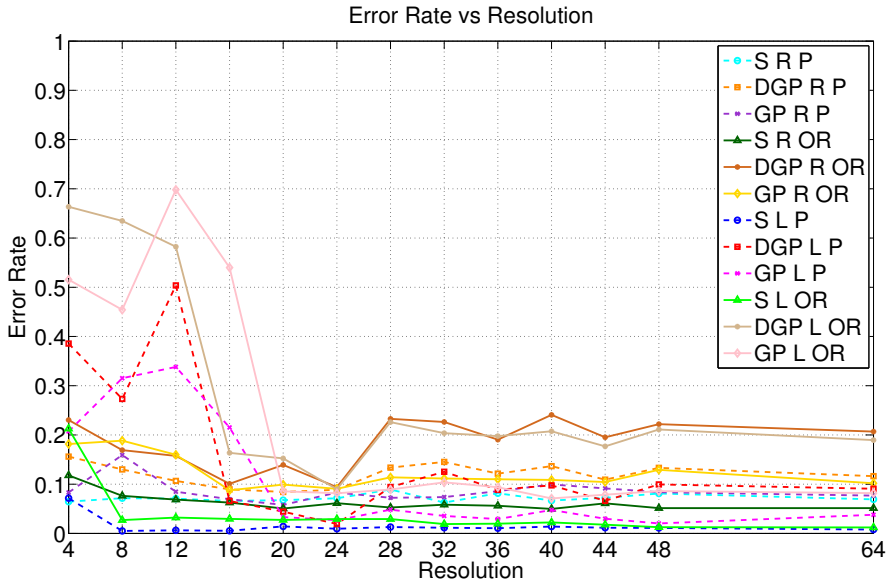


Figure 2.16: The average error rate of various recognition algorithms as a function of the resolution of hand images using CPP pre-processing. Compare the trends of these plots with those of Figure 2.15, which only differ because another pre-processing scheme is implemented.

olutions (8×8 pixel resolution) where a number of the CPP methods can still achieve $\approx$99% average accuracy whereas any RPP technique is only 40% accurate. The most striking aspect of the CPP at such low resolutions is its ability to distinguish between classes when the gestures are no longer recognizable to the human eye. Note that the 8×8 resolution is especially attractive for rapid, low overhead detection applications on hand-held electronics.

To summarize, the CPP method allows for high accuracy at low resolution. The actual accuracies presented in these graphs are not as significant as the general trends of the plots, which, in Fig. 2.15, show a nearly monotonic decline in error rates as the image resolution increases, and in Fig. 2.16, show mostly consistent accuracies

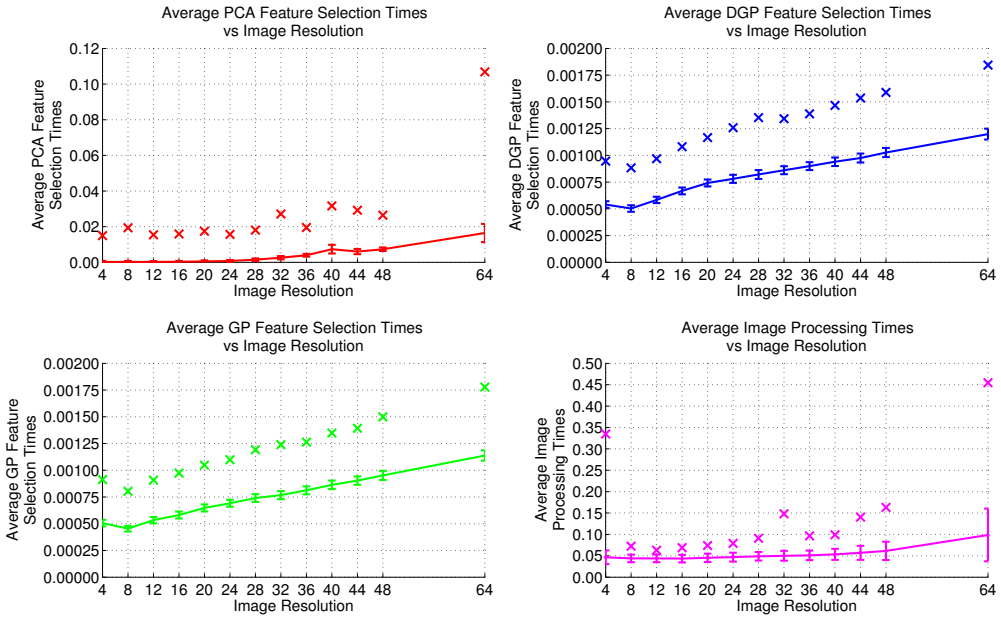


Figure 2.17: The average computation times in order to complete the feature selection and pre-processing phases of the hand gesture recognition problem. All the "x"s mark the maximum computation times, and the error bars represent one standard deviation above and below the average times.

for all resolutions. One would expect that higher resolution images would be easier to classify since the features become more salient and detailed. Of course this is only true to a certain point, eventually further detail does not contribute any new pertinent information about the features; this is why Figure 2.15 starts to level off near the 64×64 pixel resolution. However, Fig. 2.16 shows that even better than the expected performance can be achieved given the proper pre-processing, and using simple, computationally efficient feature selection and classification methods.

In addition to the accuracy as a function of resolution, the computing time required to perform the recognition algorithm is considered. As stated previously, the bulk

of the computational processing time is found in the pre-processing of the images. Figure 2.17 demonstrates the computed processing time, extracting 10 features, for the PCA based method (top left panel) as well as the GP (bottom left panel) and DGP (top right panel) methods. All of the results are presented as a function of the image resolution. The bottom right panel of this figure shows the pre-processing times as a function of resolution for the RPP method, which is where most of the computing time comes from in the recognition process.

2.6 Analysis of Pre-Processing

In order to further elucidate the importance of efficient pre-processing methods, consider the results of Table 2.2 for which a variety of gesture recognition method combinations are used on a more realistic hand recognition problem. The timing results produced here are generated on a 1.86 GHz Intel Core Duo processor, which is fairly standard among moderately powered laptops. All the images have a resolution of 32×32 pixels, and the results are averaged over 1349 images, which contain 20 more gestures other than the 5 gestures previously mentioned, and many non-gestures and/or poorly articulated gestures (See Fig. 2.11). This set of images also has a larger rotational variance among the images. Here $\bar{t}$ is the average time to complete the method, with a corresponding standard deviation σ_t. The minimum ($t_{\min}$) and maximum ($t_{\max}$) times are also given. The accuracy is measured using both LDA and logistic regression classification schemes, and are presented under the "Success Rate" column. These success rates are averages of the *within-class success rate*, meaning when images are correctly labeled within their respective class, and the *out-of-class success rate*, meaning when images are correctly not labeled to belong to classes to which they do not belong. All of the 1349 images used in these performance tests have been pre-processed using all of the pre-processing steps listed in section 2.2.1 as is illustrated in Fig. 2.10. The test that included invariant moments (IM) and logistic regression (R) failed to recognize the hand gestures because of a failure in the

optimization routine that trains the logistic regression scheme.

Figure 2.18 further breaks down the pre-processing time into its constitutive components. A relative comparison of the various pre-processing steps are illuminated so that one can visually determine their importance. This is said with the caveat that the "Background Subtraction" appears to be less significant (2.81%) to the overall pre-processing time than it should be because identifying and localizing the hand in

Feature Selection Method	$\bar{t}$ [msec]	σ_t [msec]	$t_{\min}$ [msec]	$t_{\max}$ [msec]	Success Rates
PCA	2.58	0.946	2.33	24.6	L $\rightarrow$ 0.889 R $\rightarrow$ 0.885
2DPCA	0.340	1.658	0.200	34.0	L $\rightarrow$ 0.940 R $\rightarrow$ 0.962
GP	1.709	0.531	1.482	15.90	L $\rightarrow$ 0.873 R $\rightarrow$ 0.861
DGP	1.799	0.0366	1.767	2.54	L $\rightarrow$ 0.857 R $\rightarrow$ 0.847
CGP	34.5	2.21	29.7	46.0	L $\rightarrow$ 0.897 R $\rightarrow$ 0.859
IM	10.80	1.218	10.47	50.3	L $\rightarrow$ 0.782 R $\rightarrow$ N/A
Classification Method	$\bar{t}$ [msec]	σ_t [msec]	$t_{\min}$ [msec]	$t_{\max}$ [msec]	
R	0.1008	0.0530	0.0908	1.090	
L	0.0621	0.0266	0.0554	0.737	
Pre-Processing Method	$\bar{t}$ [msec]	σ_t [msec]	$t_{\min}$ [msec]	$t_{\max}$ [msec]	
All	88.5	21.5	44.6	195.7	

Table 2.2: Performance of various methods of the gesture recognition process at an image resolution of 32×32 pixels.

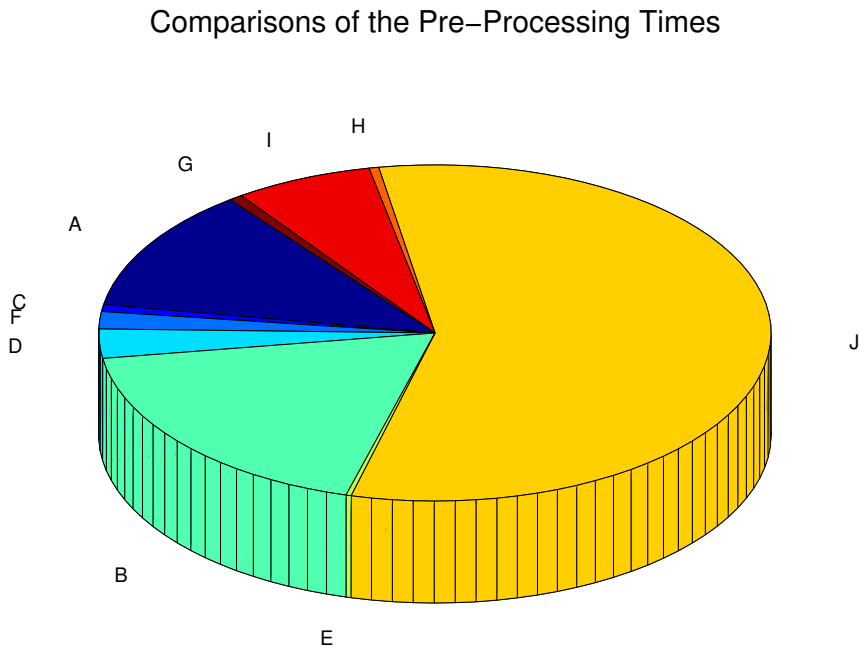


Figure 2.18: The relative comparisons of pre-processing times averaged over 1349 images and 13 different image resolutions, ranging from 4×4 images to 64×64 images. The letter labels correspond to the pre-processing steps as follows: A. Grayscale Conversion (11.87%), B. Image Resizing (18.31%), C. Intensity Normalization (0.67%), D. Background Subtraction (2.81%), E. Cropping (0.25%), F. Arm/Wrist Removal (1.67%), G. Centering (0.64%), H. Orientation Detection (0.47%), I. Rotation (6.63%), and J. Other (56.69%). In order to complete all the pre-processing steps, the average computation time was 0.0847 sec.

cluttered backgrounds is not considered. The "Other" pre-processing step (56.69%) is dominated by the process of importing the raw image.

Table 2.2 reiterates the fact that pre-processing is the most time-consuming step of the recognition process, whereas the classification scheme is the least costly step. These pre-processing times form a lower bound for the times that would occur in more realistic gesture recognition applications, which would need to implement more

elaborate and robust gesture localization and segmentation routines, not included in these results.

Since real-time applications depend on fast processing, it is apparent that improving the pre-processing performance is paramount. Table 2.2 also shows that the both classification schemes, LDA and logistic regression, produce similar results in accuracy averaged over within-class and out-of-class success rates. This provides some evidence that if the pre-processing is done well, and the best features are extracted from the images, then the classification scheme need not be overly sophisticated.

With the importance of pre-processing in mind, consider a new performance study that aims to emphasize the contributions of the various steps of the pre-processing, and whose results are illustrated in Fig. 2.19. In this experiment, only the PCA feature selection method is implemented because of its sensitivity to variations in translation, rotation, and scale. Also, only the LDA classification scheme, with a one-vs-the-rest comparison style, is used to determine the gesture class of the hand images. This study uses only one well articulated image for each gesture class, there being five different classes in total, as before. These images do have some significant portions of the arm/wrist region showing in the frame of the images.

Recalling the nine pre-processing steps from section 2.2.1, these images are pre-processed in 12 different ways: The first six ways stem from completing all the possible pre-processing steps ("All"), and then, in the other five ways, a single pre-processing step is removed. The last six ways stem from completing as little pre-processing as possible ("None"), and then, in the remaining five ways, a single pre-processing step is implemented. However, even the images that are pre-processed under "None" are converted to grayscale, resized to the desired image resolution, and have the hand segmented from the background. In order to depict the effects of not having certain pre-processing steps, *noise* is added to the images in order emphasize the missing step(s); i.e. when no image cropping is done, the hand size is purposely rescaled in its frame to appear much smaller than the ideally articulated gesture. Other types of

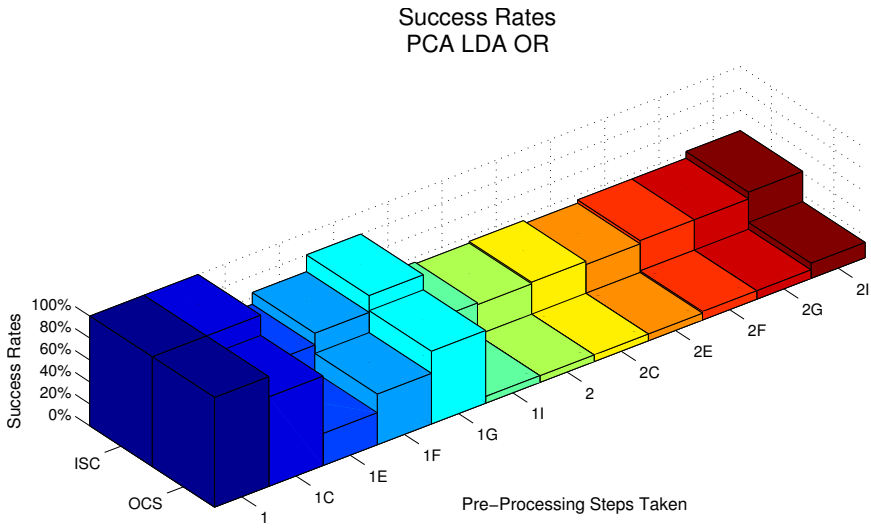


Figure 2.19: The success rates of recognizing 5 hand gestures, just using 1 image for each class (gesture). Within-class success rates (ICS) are calculations of correctly labeling images within their respective classes. Out-of-class (OCS) success rates are determined from correctly not labeling images into classes to which they do not belong. The success rates are averaged over the five images of each pre-processing scenario and over 13 different image resolutions, ranging from 4×4 images to 64×64 images. The alpha-numeric labels correspond to the pre-processing methods as follows: 1. All, 1C. All But Intensity Normalization, 1E. All But Cropping, 1F. All But Arm/Wrist Removal, 1G. All But Centering, 1I. All But Rotation, 2. None, 2C. Only Intensity Normalization, 2E. Only Cropping, 2F. Only Arm/Wrist Removal, 2G. Only Centering, and 2I. Only Rotating.

image noise includes de-centering the hand within the frame of the image, and a 90 degree rotation of the hand counter-clockwise.

The success rates are averaged over the five images of each pre-processing scenario, and over 13 different image resolutions, ranging from 4×4 to 64×64. Figure 2.19 illustrates the results of this test with both the within-class and out-of-class success

rates for all 12 pre-processing scenarios.

As would be expected, the images that were pre-processed with "All" of the possible steps performed with perfect accuracy because the classification was trained (learned) using these images. The success rates of these fully pre-processed images serve as an upper bound to which the other pre-processing methods can be relatively compared against. Likewise, success rates for the images that were pre-processed with "None" of the possible steps, serve as a lower bound to which the other pre-processing methods can be relatively compared against. In this case, the lower bound is about a 40% accuracy. Thus, the actual accuracies are not as meaningful as the relative differences in accuracy between each pre-processing scenario.

Figure 2.19 clearly shows that pre-processing is key to accurately recognizing gestures. The normalization of the pixel intensities seems to be the least important of the pre-processing steps in this study, while rotation seems to be the most important. Some pre-processing steps depend on other steps; for instance, centering the hand within the frame of the image is useless without first removing the excess arm/wrist regions. Not accounting for the PCA method's sensitivity to translation, scale, and rotation variances in the pre-processing causes the recognition rates to suffer.

One way of avoiding the costly pre-processing would be to use translation, scale, and/or rotation invariant feature selection methods. Even still, the pre-processing procedure cannot be completely avoided. For instance, the arm/wrist detection is still needed in order to for any feature selection method to properly identify the true center of the hand. Also, identification, localization, segmentation, and background removal will always be a crucial aspect of pre-processing, and is the most time consuming part as well. Additionally, rotation may still be an issue for some background removal methods to work properly, e.g. the Viola-Jones detector is sensitive to gesture rotations [75]. Finally, sometimes the best features may not come from translation, scale and rotation invariant methods. Referring back to Table 2.2, it is clear that the PCA and GP methods, and their variants, out-performed the IM method in both

accuracy and speed.

2.7 Chapter Discussion

Dynamic mode decomposition has been shown to be a viable and efficient method for time-scale dynamics separation and filtering problems. For the important application of background/foreground separation in videos, the DMD method produces a high quality separation for most realistic and practical videos, the results of which are comparable with the leading competing method found in the RPCA algorithm, except that the DMD method is at least an order of magnitude faster computationally. For other separation tasks, the DMD method has shown a great deal of promise in being able to properly identify and filter dynamics operating on different time-scales.

As with any separation method, including RPCA and DMD, the burden of working with too much data, i.e. high-resolution images and/or many frames per video segment, can be problematic because of reduced computational speeds and limited memory sizes. Nonetheless, the DMD algorithm has shown itself to be robust and efficient enough to produce attractive results in times well below the normal frame acquisition rate of most cameras, allowing for higher pixel resolutions and video segment sizes to be used. For real-time video applications, it makes sense to break the continuous video stream into segments large enough to ensure that there is enough information to complete an adequate background/foreground separation, but small enough to keep the processing times smaller than the data acquisition times. Plus, moving objects that turn, stop, and/or accelerate are better handled by the DMD procedure as individual actions, than in one large video segment.

It is important to understand that the DMD method is not actually performing a true low-rank/sparse separation of the given data, only an approximation to these components. In the two soliton example presented here, the data (dynamics with noise) initially has a rank of 25, the same as the number of steps in time. The RPCA method is able to find a low-rank structure with a rank of 13, and a corresponding

sparse structure with a rank of 25. If the soliton solution without any noise was stationary in time and if the RPCA method made a perfect low-rank/sparse separation, then the low-rank term $\mathbf{L}$ would have had a rank of 1. The DMD method separates the data into approximate low-rank and sparse structures, both with ranks of 25. Note that the ranks, and thus condition numbers, of the approximate low-rank and sparse DMD terms are also skewed by the process of ensuring real-valued, non-negative matrix entries for both matrices, which requires taking magnitudes of complex values and exchanging values between the two matrices.

However, the condition numbers κ to these data matrices suggest that the DMD method is truly approximating the low-rank and sparse components to the original data. The low-rank structure is expected to be *ill-conditioned* in terms of solving linear equation problems, and therefore should have a higher condition number; while the sparse structure is expected to be *well-conditioned*, and thus should have a lower condition number. In this two-soliton with noise problem, the original data has a condition number $\kappa \approx 24.8$, and the RCPA low-rank/sparse separation produces condition numbers of $\kappa \approx 5.23 \cdot 10^{16}$ and $\kappa \approx 2.42$, respectively. Compare this to the DMD low-rank/sparse separation, which produces the condition numbers of $\kappa \approx 203$ and $\kappa \approx 16.96$, respectively. This suggests that if one were looking for an exact low-rank/sparse separation, then one could use RPCA method with the DMD solution as an excellent initial guess to the PCP problem (A.1), potentially speeding up the computational time needed to solve the convex nonlinear optimization of the RPCA routine.

Figure 2.6 demonstrates the fact that the DMD method can be applied iteratively in order to converge toward the true low-rank and sparse components of the given data. The successive DMD iterations are applied to the approximate sparse structure, creating a better approximate sparse structure and adding more data back into the low-rank structure. In the future, finding the exact rate of convergence and understanding the theoretical basis for this convergence will be important steps to-

ward establishing an analytical connection between the RPCA and DMD separation algorithms.

The DMD routine does accumulate error in the last time step (or last frame) of its reconstruction, but it cannot reconstruct every type of dynamics. More analytical work is needed in order to properly understand conditions under which the DMD method will fail at separating low-rank/sparse structures within the data, and/or when the DMD method cannot reconstruct the given data.

It has been established that through clever and thoughtful pre-processing, the recognition algorithms can be greatly improved. When the gesture images are well-normalized, even simple feature selection, statistical learning, and classification techniques are effective enough to produce accurate recognitions in real-time, and can overcome translational, scale, and rotational variances that would otherwise plague the algorithms' performance capabilities. Since feature extraction and classification are computationally quick to accomplish, there is not too much to be gained by improving these processes, especially through more elaborate and sophisticated schemes. But, pre-processing is inherently a more time-consuming procedure, and it holds a great influence over the ability of the feature selection and classification to preform properly. Thus, there is much more to be gained through enhancing pre-processing methods so that they are more robust and efficient.

It was seen that, even at image resolutions as low as 8×8 pixels, accuracies of 99%, using PCA feature selection, and 95%, using generalized projection techniques, could be achieved given the right kind of pre-processing (CPP), and, of course, with a somewhat idealized dataset that starts with dark, uniform backgrounds. It is likely that such accurate results are achieved because the pre-processing is tailored to this particular application and experiment. Not all applications will be able to produce such performance, but the pre-processing schemes can be optimized around the given application.

Chapter 3

BEST GESTURE LEXICONS

By carefully crafting the pre-processing procedure, the recognition process can benefit from faster processing times, greater accuracy, and applicability to a more diverse set of situations. However, computer algorithms alone cannot ensure that a gesture recognition scheme will achieve excellent performance results. It is also important to consider what gestures will be used in a given application; namely, whether those gestures are favorable for both the gesture articulator and the recognition algorithms involved.

Figure 2.12 highlights this idea of choosing the best gestures for a recognition project, out of a potentially vast set of available gestures. One would have to consider how some gestures, e.g. gestures **a**, **0**, and **v**, already have vernacular meanings in certain contexts, i.e. "stop"/letter "a", "zero"/"nothing"/letter "o", and "peace"/"two"/letter "v", respectively. But, it is also important to consider that in order for the selected gestures to be well-distinguished by a computer algorithm, they need to be well-separated in feature space. In particular, if $n = 3$ gestures were required for the given project and one could choose from the $m = 5$ lexicon shown in Fig. 2.12, then one might choose **0**, along with one of either **a** or **e**, and one of either **u** or **v**, depending on the application for which it was intended.

From a computer's perspective, the best gestures are those which are the most easily distinguished through its given feature selection and statistical learning algorithms. For the computer, ergonomic and/or vernacular gestures are irrelevant to its consideration process. Previously, arguments have been made claiming that well-clustered (with-in class) and well-separated (between classes) gestures in feature space

constitute easily distinguishable gestures for computer algorithms. However, it may be argued that with many features, even weak ones, and a sophisticated statistical learning algorithm, a computer can still achieve great recognition accuracies, even when the gestures classes are either not well-separated in feature space or are separated in some complicated, nonlinear fashion. There is validity to this argument, however with some caveats.

As was alluded to previously, one major problem in the recognition field is *over-fitting*. Over-fitting occurs when a statistical learning algorithm learns its training set too well, and thus is able to distinguish between the training set gesture classes with excellent accuracies, but at the cost of not being able to maintain those accuracies in a generalized setting. Over-fitting can occur by simply not having a diverse enough training set, but also because of the complexity and over-sophistication of the learning algorithm. One might think of finding an incredibly complicated separatrix, like the one in Fig. 3.1 (B), that perfectly distinguishes two classes in the training set, but does not do as good of a job distinguishing the test data which may actually have overlapping features from the two classes.

For most recognition applications, one cannot determine how the different classes will render themselves in feature space. And since many sophisticated statistical learning algorithms need some *a priori* information in order for the algorithm to learn a complex, nonlinear separation measure, one is often left to either guess-work or very specific problems and applications where *a priori* information is known.

A good statistical learning algorithm may also be able to learn a nonlinear separation between classes if there are enough training data points available, especially near and well-distributed around the nonlinear separation. However, not every application can provide a sufficient number of training data points, nor would it be easy to guarantee that those points would be near and well-distributed around the critical areas of the feature space. This problem is further exacerbated by having high-dimensional data, which multiplies the number of data points needed in order to properly learn

complicated class separations as the dimension increases. This is another reason for leaving out weak features if possible.

In some recognition applications, especially when there are only a few gestures needed, it may be acceptable to have gesture classes that are spread out in feature space, as long as the classes are separated from one another. However, in general, expansive classes tend to be harder to separate than well-clustered classes [23]. In part, this is because most features tend to yield values in a limited range, and so sparse classes are more likely to overlap in feature space. Also, as more gestures are needed for the given application, it becomes more likely that a spread out gesture class will overlap in feature space with another class, making them difficult to distinguish.

Therefore, having complicated, nonlinear separations between classes is not very desirable, nor is having spread out features in feature space. Thus, the idea of keeping the gestures in feature space well-clustered with-in individual classes and well-separated between different classes remains an excellent indication of how easily a computer will be able to distinguish the classes in a recognition algorithm. This indication remains valid and general across the many applications and specific algorithms used in the recognition process [76].

3.1 Ellipsoidal Distance Ratio

In order to create an objective metric that captures the desirable feature space characteristics for the recognition process of computers, it makes sense to reward gesture classes that are distributed in ellipsoidal-like shapes in feature space. When a gesture's features lie in an ellipsoidal distribution in feature space, they are likely to be both well-clustered and easily separated from other classes. The *ellipsoidal distance ratio metric* (EDRM), to be introduced presently, is one possible metric for gesture class separation that takes advantage of rewarding optimal class feature space distributions. Here, the distinctions between the terms ellipse, ellipsoid, and hyper-ellipsoid are ignored, and the general term ellipsoid is used no matter the dimension of the

object at hand.

The first step to calculating the EDRM between two classes is to find the ellipsoid in the appropriate dimension which best encompasses the feature points for each gesture class, even, of course, if those points are not in a true ellipsoidal distribution. This is accomplished by finding the positive definite matrix E for each class of the general ellipsoid equation: $(\vec{x} - \vec{x}_0)^* E^{-1}(\vec{x} - \vec{x}_0) = 1$, where $(^*)$ is the Hermitian transpose, $\vec{x}_0$ is the center of the ellipsoid, $\vec{x}$ defines a point on the surface of the ellipsoid, and the eigenvectors and eigenvalues of the matrix E are the principal directions of the ellipsoid and the squares of the semi-axis lengths, respectively. The centers of the ellipsoids are the centers of the feature points for each class in feature space.

Assume that the feature data for a given gesture m is in a data structure Y^m such that Y^m has L columns, which corresponds to the number of features extracted from the images, and the ℓ^{th} column has N_m rows, which corresponds to the number of images in gesture class m. Note that the number of features extracted L is also the dimension of the feature space. By the properties of the *singular value decomposition* (SVD), the ellipsoidal matrix for gesture m is $E_m = (Y^m)^* \cdot Y^m$. This works out because the SVD states that any matrix $Z \in \mathbb{R}^{q \times r}$ can be decomposed such that $Z = U\Sigma V^*$, where $U \in \mathbb{R}^{q \times q}$ is unitary, $\Sigma \in \mathbb{R}^{q \times r}$ is diagonal, and $V \in \mathbb{R}^{r \times r}$ is unitary. This implies that $Z \cdot Z^* = U\Sigma^2 U^*$, which is the eigendecomposition of $Z \cdot Z^*$. So if the ellipsoid matrix $E = Z \cdot Z^*$, then the principal directions of the ellipsoid are found in the columns of U and the corresponding squares of the semi-axis lengths are along the main diagonal of Σ^2. Letting $Z = (Y^m)^*$, the ellipsoidal matrix E is guaranteed to have the proper dimensions $(L \times L)$ and proper attributes that best fit the feature data of feature m into an ellipsoid.

Obviously, the more gesture images in each class, the better sampling one will get for the true size and nature of the class in feature space. It would be good to have at least as many images in each gesture class as there are dimensions of the feature

space (L) in order to create the gesture ellipsoids and to calculate the EDRM.

At this point one has the option of rescaling the size of the ellipsoid for each gesture class. This somewhat arbitrary rescaling can have significant effects on the EDRM. Clearly, it is best to choose a consistent rescaling scheme across all the gesture classes in order to maintain fair comparisons of the class sizes and distances in feature space. It is suggested that each ellipsoid's surface be a certain percentage of the distance from the center of the ellipsoid to the most extreme outlier feature point in that gesture class. In this way, it is guaranteed that at least some feature points from each class are on the outside of the ellipsoid of its own class. This method acknowledges the reality that there are almost always impure articulations (outliers) of any gesture that may not fully represent the class. The actual percentage used in this convention can be determined based on how pure the gesture images are for each class. For instance, if all the images used in the dataset are of well-articulated gestures, then one might choose to have the ellipsoid encompass most of the feature points; this is assuming there are few to no outliers, and so a high percentage value is used, e.g. 90% – 100%. Whereas, a noisier sampling of the gestures might assume the existence of outliers in each class, and so a lower percentage of 60% – 80% is used. In what follows, the convention will be set at 65%.

The *ellipsoidal distance ratio* is the shortest distance between two ellipsoids compared against the distance between the ellipsoid centers. Measuring the straight-line distance between ellipsoid centers is trivial. However, the shortest distance between two ellipsoids needs to be formulated as an optimization problem. Suppose there are two classes, $\mathbf{a}$ and $\mathbf{b}$, with ellipsoid surface points $\vec{x}^{\mathbf{a}}$ and $\vec{x}^{\mathbf{b}}$, centers $\vec{x}_0^{\mathbf{a}}$ and $\vec{x}_0^{\mathbf{b}}$, and ellipsoid matrices $E_{\mathbf{a}}$ and $E_{\mathbf{b}}$, respectively. The optimization problem can then be

stated as follows

$$\begin{aligned} &\arg\min \sqrt{\left(\vec{x}^{\mathbf{a}} - \vec{x}^{\mathbf{b}}\right)^* \cdot \left(\vec{x}^{\mathbf{a}} - \vec{x}^{\mathbf{b}}\right)} \\ &\text{subject to } \left(\vec{x}^{\mathbf{a}} - \vec{x}_0^{\mathbf{a}}\right)^* E_{\mathbf{a}}^{-1} \left(\vec{x}^{\mathbf{a}} - \vec{x}_0^{\mathbf{a}}\right) = 1 \ \& \\ &\left(\vec{x}^{\mathbf{b}} - \vec{x}_0^{\mathbf{b}}\right)^* E_{\mathbf{b}}^{-1} \left(\vec{x}^{\mathbf{b}} - \vec{x}_0^{\mathbf{b}}\right) = 1. \end{aligned}$$

This problem effectively says that one wants the shortest distance between the points $\vec{x}^{\mathbf{a}}$ and $\vec{x}^{\mathbf{b}}$, while constraining $\vec{x}^{\mathbf{a}}$ to be on the surface of the ellipsoid of gesture $\mathbf{a}$ and constraining $\vec{x}^{\mathbf{b}}$ to be on the surface of the ellipsoid of gesture $\mathbf{b}$. Without any loss of purpose or accuracy, the optimization problem can be better formulated for numerical and algorithmic stability issues such that

$$\begin{aligned} &\arg\min \frac{\left(\vec{x}^{\mathbf{a}} - \vec{x}^{\mathbf{b}}\right)^* \cdot \left(\vec{x}^{\mathbf{a}} - \vec{x}^{\mathbf{b}}\right)}{2} \\ &\text{subject to } \left[\left(\vec{x}^{\mathbf{a}} - \vec{x}_0^{\mathbf{a}}\right)^* E_{\mathbf{a}}^{-1} \left(\vec{x}^{\mathbf{a}} - \vec{x}_0^{\mathbf{a}}\right) - 1\right]^2 = 0 \ \& \\ &\left[\left(\vec{x}^{\mathbf{b}} - \vec{x}_0^{\mathbf{b}}\right)^* E_{\mathbf{b}}^{-1} \left(\vec{x}^{\mathbf{b}} - \vec{x}_0^{\mathbf{b}}\right) - 1\right]^2 = 0. \end{aligned}$$

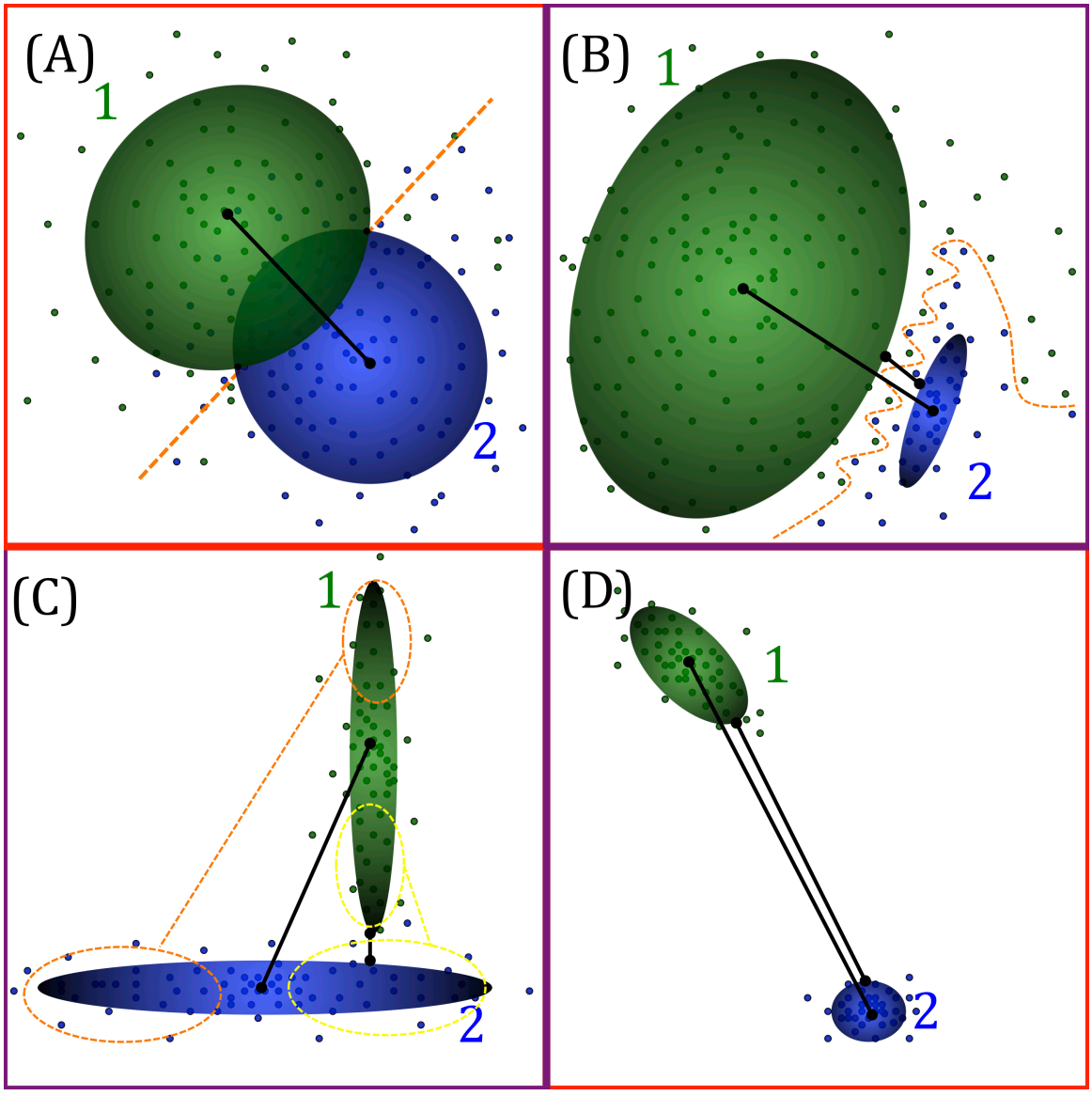
Since, in high-dimensions, these ellipsoids cannot be easily visualized, one needs to take extra precautions to ensure that the ellipsoids do not overlap in feature space. Given the optimal points $\tilde{\vec{x}}^{\mathbf{a}}$ and $\tilde{\vec{x}}^{\mathbf{b}}$, one check that can be made is to ensure that

$$\begin{aligned} &\left(\tilde{\vec{x}}^{\mathbf{b}} - \vec{x}_0^{\mathbf{a}}\right)^* E_{\mathbf{a}}^{-1} \left(\tilde{\vec{x}}^{\mathbf{b}} - \vec{x}_0^{\mathbf{a}}\right) > 1 \ \& \\ &\left(\tilde{\vec{x}}^{\mathbf{a}} - \vec{x}_0^{\mathbf{b}}\right)^* E_{\mathbf{b}}^{-1} \left(\tilde{\vec{x}}^{\mathbf{a}} - \vec{x}_0^{\mathbf{b}}\right) > 1. \end{aligned}$$

These inequalities verify that the optimal point on ellipsoid $\mathbf{b}$ is not inside of the ellipsoid for gesture $\mathbf{a}$, and vice-versa. Also, due to numerical inaccuracies it can be good to verify that

$$\begin{aligned} &\left|1 - \left(\tilde{\vec{x}}^{\mathbf{a}} - \vec{x}_0^{\mathbf{a}}\right)^* E_{\mathbf{a}}^{-1} \left(\tilde{\vec{x}}^{\mathbf{a}} - \vec{x}_0^{\mathbf{a}}\right)\right| < \epsilon \ \& \\ &\left|1 - \left(\tilde{\vec{x}}^{\mathbf{b}} - \vec{x}_0^{\mathbf{b}}\right)^* E_{\mathbf{b}}^{-1} \left(\tilde{\vec{x}}^{\mathbf{b}} - \vec{x}_0^{\mathbf{b}}\right)\right| < \epsilon, \end{aligned}$$

Figure 3.1: In this figure four examples illustrate why ellipsoidal distance ratios (EDRs) are good measures of class separation in feature space. The EDR is the shortest distance between two ellipsoids divided by the distance between the ellipsoid centers (black lines). When two classes overlap (A), so do their corresponding ellipsoids; thus, the EDR is zero. Even when two classes have non-ellipsoid distributions of feature points and are separated by a nonlinear separatrix (B), the EDR still provides a reliable measure metric for closeness. The EDR measure is not easily fooled when many points from the two classes are well separated, but some are not (C) - compare the distances between features points in the upper half of class **1** and in the left half of class **2** to the distances using the lower half of class **1** and the right half of class **2**. Ideally, each class is well-clustered and well-isolated from the other class (D), thus yielding a EDR value near one.


(A)
1
2
(B)
1
2
(C)
1
2
(D)
1
2

where ϵ is some small tolerance for numerical inaccuracy, which may need to grow with the number of dimensions of the problem. These inequalities ensure that the optimal point for ellipsoid **a** is truly on the surface of ellipsoid **a**, and the same for ellipsoid **b**.

Once these points on the two gesture class ellipsoids that minimize the distance between the ellipsoids have been found, one can calculate the *ellipsoidal distance ratio metric* as follows

$$\mathrm{EDRM}_{\mathbf{a},\mathbf{b}} = \frac{\sqrt{\left(\tilde{\vec{x}}^{\mathbf{a}} - \tilde{\vec{x}}^{\mathbf{b}}\right)^* \cdot \left(\tilde{\vec{x}}^{\mathbf{a}} - \tilde{\vec{x}}^{\mathbf{b}}\right)}}{\sqrt{\left(\vec{x}_0^{\mathbf{a}} - \vec{x}_0^{\mathbf{b}}\right)^* \cdot \left(\vec{x}_0^{\mathbf{a}} - \vec{x}_0^{\mathbf{b}}\right)}}.$$

The EDRM values are always between 0 and 1. Larger EDRM values indicate that the two classes that are being compared are further separated and better clustered in feature space. Also note that the EDRM normalizes the scale of the ellipsoids in that larger ellipsoids must be more separated from one another in order to have the same EDRM score as smaller ellipsoids. If the volumes of both ellipsoids go to zero, because of the clustering of feature points, then the EDRM approaches its ideal value of 1 because the shortest distance between the ellipsoid surfaces (numerator) becomes the same as the distance between the ellipsoid centers (denominator).

Figure 3.1 illustrates why the EDRM is a good metric for gesture separation. First note that when feature points from the two given classes significantly overlap (A), the EDR value will always be zero because the class ellipsoids will also overlap and so the distance between the ellipsoid surfaces is zero. In the case that one or both feature point regions for the two classes have non-ellipsoidal distributions (B), the best-fitting ellipsoids to the class regions are determined. Further, in the case that these regions are still fairly well-separated, a EDRM score can be calculated that indicates how well-separated the regions are. In some instances there can be sub-regions of the two classes that are well-separated in feature space, and yet other sub-regions of the two classes in feature space that are nearby each other (C). In this scenario, the EDRM will produce a practical value that accounts for the different sub-regions and their

respective separations. The EDRM is neither misled into being too high nor too low by the positioning of the sub-regions. Ideally, the two classes are well-clustered and well-separated (D).

3.2 Subjective Separation Measures

For many applications, the comfort and ease of articulating a gesture can be an important factor for determining which gestures are more appropriate for use. Moreover, some gestures better suite an application because of the cultural norms and meanings of the gesture already in use in society. In order to incorporate these ergonomic and vernacular reasons into the best gestures decision making process, one would need to subjectively rate all the gestures in the entire lexicon of available gestures.

One method of creating a subjective measure for ranking the suitability of the gestures that is comparable with the EDRM, is by first rating the ergonomic and vernacular quality of each gesture on a scale of $[0, 1]$. A rating of 0 would mean *not suitable* for the given application, and a rating of 1 would mean *most suitable* for the task. The fact that this rating scale is the same as the range of possible EDRM scores is not coincidental.

Note that the EDRM scores are comparisons between two classes, therefore the subjective ratings must also be a comparisons between two classes in order to be able to integrate this subjective measure with the EDRM. This is accomplished by creating a *subjective measure* (SM) for every class pairing that is the average of the numerical ratings that each class in the pair was given.

$$\mathrm{SM}_{\mathbf{a},\mathbf{b}} = \frac{\mathrm{Rating}(\mathbf{a}) + \mathrm{Rating}(\mathbf{b})}{2} \in [0, 1]$$

Table 3.1 is a four class lexicon example of how subjective measures are calculated. Note that the values of this table are subjectively generated. Ultimately, one might refine this process by having a large sample of people vote on their favorite gestures, thus generating average scores of gesture likability across a population.

The subjective measure and the EDRM for each class pairing can be combined into a *total measure* (TM) by a weighting factor $\alpha \in [0,1]$, which determines the relative influence that the subjective measure will have on the total measure.

$$\mathrm{TM}_{\mathbf{a},\mathbf{b}} = \alpha \mathrm{SM}_{\mathbf{a},\mathbf{b}} + (1-\alpha)\mathrm{EDR}_{\mathbf{a},\mathbf{b}} \in [0,1]$$

If any gesture must be included in the best lexicon of size n, then the problem becomes a matter of finding the best lexicon of size $n-1$, where the n^{th} gesture is already determined. In the case that one of p ($p \ll n$) gestures must be included, in the best lexicon of size n, then this constraint can be enforced in the programming of the algorithmic search process. Finally, if there is a gesture or multiple gestures that should never be chosen to be a part of the best gestures set, then these gestures can simply be removed from the entire lexicon of available gestures, and therefore from the process of consideration.

3.3 Best Gestures

Once one has calculated the total measure (TM) for every gesture class pairing in the entire lexicon of available gestures, determining the best gesture subset is a fairly

Table 3.1: Here a lexicon of 4 gestures, labeled **a** – **d**, receives subjective ratings, written in parentheses, which are used to create subjective measures (SM) between each class pairing.

	a (0.9)	**b** (0.2)	**c** (0.7)	**d** (0.5)
a (0.9)	-	0.55	0.80	0.70
b (0.2)	0.55	-	0.45	0.35
c (0.7)	0.80	0.45	-	0.60
d (0.5)	0.70	0.35	0.60	-

straight-forward process. Assume there are a total of m available gestures in our lexicon library, and one desires to know what is the best lexicon of size n, where $n < m$. The best lexicon of size 2 is simply the gesture class pairing with the largest TM value. In order to calculate the best lexicon of size $n \geq 3$, one must first find all the unique combinations of n gestures in the set of m possible gestures. Then, the sum of the TMs from each pairing in every unique combination of n gestures is calculated. The unique grouping of n gestures with the highest total sum TM value constitutes the best lexicon of size n.

Note that this problem is *NP hard* because there are

$$\binom{m}{n} = \frac{m!}{n!(m-n)!}$$

unique combinations of size n, and

$$\binom{n}{2} = \frac{n!}{2!(n-2)!}$$

pairings within each combination of n. Therefore, determining the best gesture set of size $n \geq 3$ requires at least

$$\binom{m}{n} \cdot \binom{n}{2} = \frac{m!}{2!(n-2)!(m-n)!}$$

operations, which grows unsustainably with m, and the problem becomes intractable. However, the process is still manageable when n is small relative to m.

Note that the subjective metric (SM) is set up in such a way that the best lexicon of size n will always include the best lexicon of size $n-1$; thus, one only needs to find the best gesture to add to the $n-1$ already found. This makes the problem much more tractable. However, the objective EDRM does allow for situations where the best lexicon of size n does not include the best lexicon of size $n-1$. This can occur for a variety of reasons including cases where there are two similar gestures,

which of course are nearby in feature space, and one of them is better separated from a few other gestures. But, as the number of gestures included in the best gesture set increases, the other gesture of the two like-gestures becomes the better choice. Another example is when two different gestures are very well-separated from one another in feature space; however, one of the two gestures is fairly close to the other available gestures in the entire lexicon. In this case, the best lexicon of 2 is determined by the two aforementioned well-separated gestures, but the best lexicons of $n \geq 3$ will not include the gesture that is close to the rest of the available gestures. In any case, assuming that the best lexicon of size n includes the best lexicon of size $n-1$ simplifies the search process, but at the cost of potentially not finding the true best lexicon of size n.

3.4 Large Lexicon Example and Results

In order to test this process for determining the best gesture sets, consider a real static hand gesture recognition problem. For this example, consider the Massey University ASL static hand gesture dataset [77, 78] of 36 hand gestures, which has 70 renditions of each gesture class, except for class **t** which has 65 renditions (See Fig. 3.2). The hands have already been isolated within each image and segmented with black backgrounds. The raw images are pre-processed by converting to grayscale, centering the hand within the frame, down-sampling the images to 32×32 (without changing the aspect ratio of the hand), and normalizing the brightest pixel intensity [79]. Feature selection is done by the PCA method and the GP method, both only using 10 features. Statistical learning will be done by the LDA method [55], with a one-vs-the-rest classification style [79]. The training set consists of 20 randomly chosen images from each gesture class, and the test data consists of the remaining 50 images (45 for class **t**) from each class.

For the case where there are no subjective considerations to be incorporated into the decision making process ($\alpha = 0$), the computer will rely solely on the objective

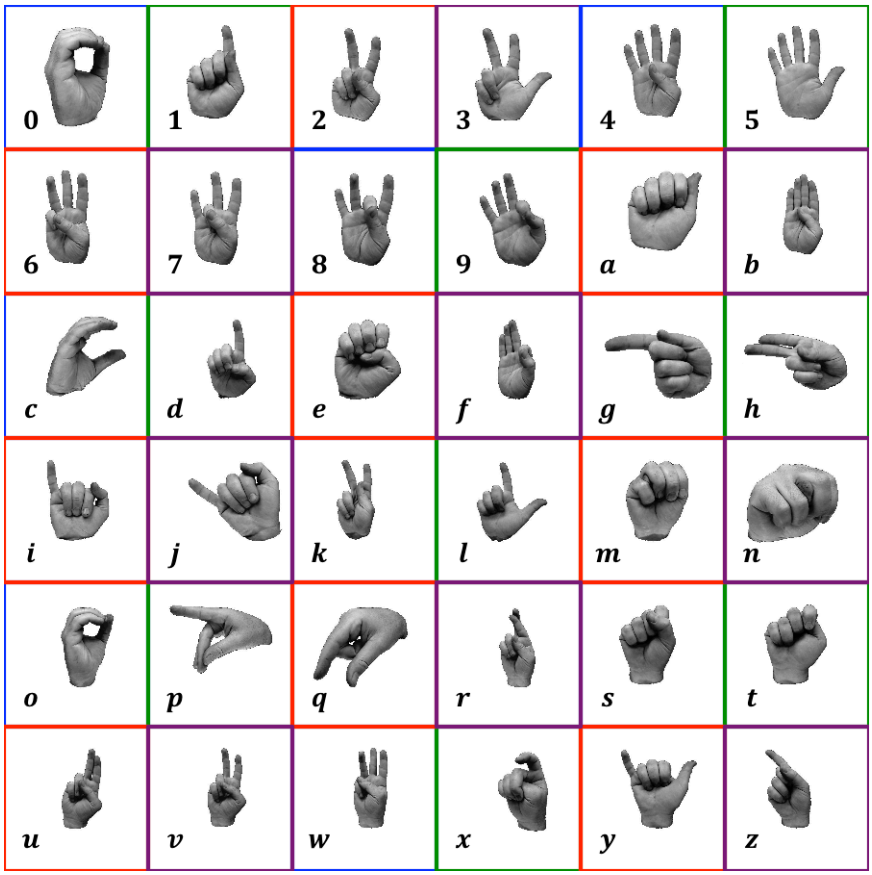


Figure 3.2: The ASL static hand gesture dataset from Massey University [77, 78] is a lexicon of 36 gesture classes, labeled $\mathbf{0-9}$, then $\mathbf{a-z}$.

EDRM to determine the best lexicon of size n. Figure 3.3 illustrates the results of this experiment using both the PCA (left) and GP (right) feature selection methods. Because these methods produce profoundly different features, the best gesture sets that are determined are also different.

Of course, the true best gesture sets, without subjective considerations, are those which produce the highest recognition rates from the given algorithms. Recognition accuracy can take on two forms: (1) the *within-class success rate*, meaning when images are correctly labeled within their respective class, and (2) the *out-of-class*

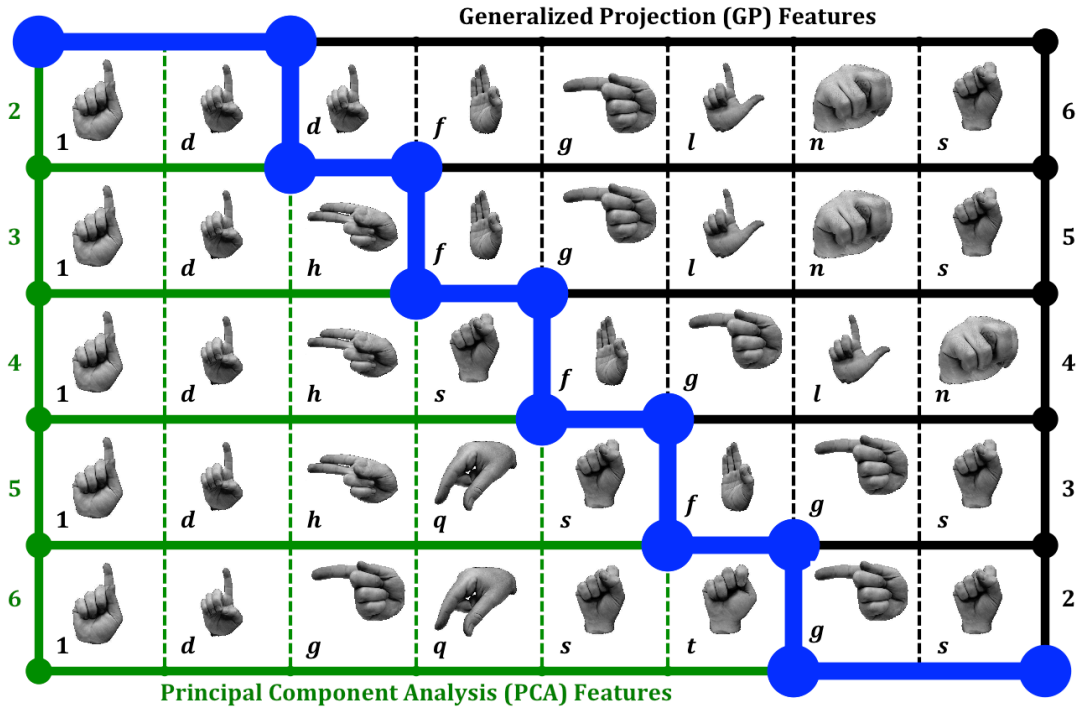


Figure 3.3: These are the best lexicons of sizes $n = 2 - 6$ using the PCA (green) and GP (black) feature extraction methods. The lexicons of size n are read across each row; the blue line separates the two feature selection methods. In this scenario $\alpha = 0$, which means that any subjective considerations for determining better gestures are being ignored and the total measure (TM) comes solely for the EDRM.

success rate, meaning when images are correctly not labeled to belong to classes to which they do not belong. By counting the number of times each image is correctly labeled within its own class and not labeled to belong to another class, the *within-class* and *out-of-class* success rates can be calculated and then averaged together to create an overall average recognition rate. By finding this overall average recognition rate for every possible combination of n gestures, and picking the maximal rate, one thus discovers the corresponding true best gesture set of size n.

Using this prescribed method, all of the best lexicons of size n in Fig. 3.3 that were determined by the EDRM have been confirmed to also be the true best lexicons. For all of the different cases presented in Fig. 3.3, there are multiple lexicons of size n that tied for being the best gesture set according to the average recognition rates (only one is shown), which is not surprising for small lexicons.

One caveat to mention is that the LDA statistical learning method relies on Fisher linear discriminants, which attempt to group like-classes while separating different classes in a projection from feature space onto a line. Since the EDRM rewards pairs of classes that are both well-clustered and well-separated in feature space, the LDA method is more likely to produce good recognition rates on the same classes that bode well under the EDRM. Other statistical learning algorithms may produce slightly different best lexicons, but are still very likely to rate the best lexicons found using the EDRM very highly, for the reasons mentioned earlier in the beginning of Section **??**.

One can incorporate subjective considerations into the TM metric as was described in Section **??**. Table 3.2 lists an example of possible subjective rankings given to each gesture for the purposes of this experiment. Using these rankings, the SM for each class pairing is calculated and incorporated into the TM according to the weighting $\alpha > 0$. Figure 3.4 displays the best lexicons of sizes $2 - 6$ for the four scenarios: (a) $\alpha = 0.25$, (b) $\alpha = 0.50$, (c) $\alpha = 0.75$, and (d) $\alpha = 1$.

3.5 Chapter Discussion

Improving software algorithms is one technique for securing good recognition results in a given application, but it is also important to make sure the recognition problem is *well-posed* for both the gesture articulator and for the underlying computer procedures. Determining a limited number of best gestures in a large lexicon (library) of possible gestures, while account for both subjective and objective considerations is an effective way to ensure that the recognition problem is well-posed. To have

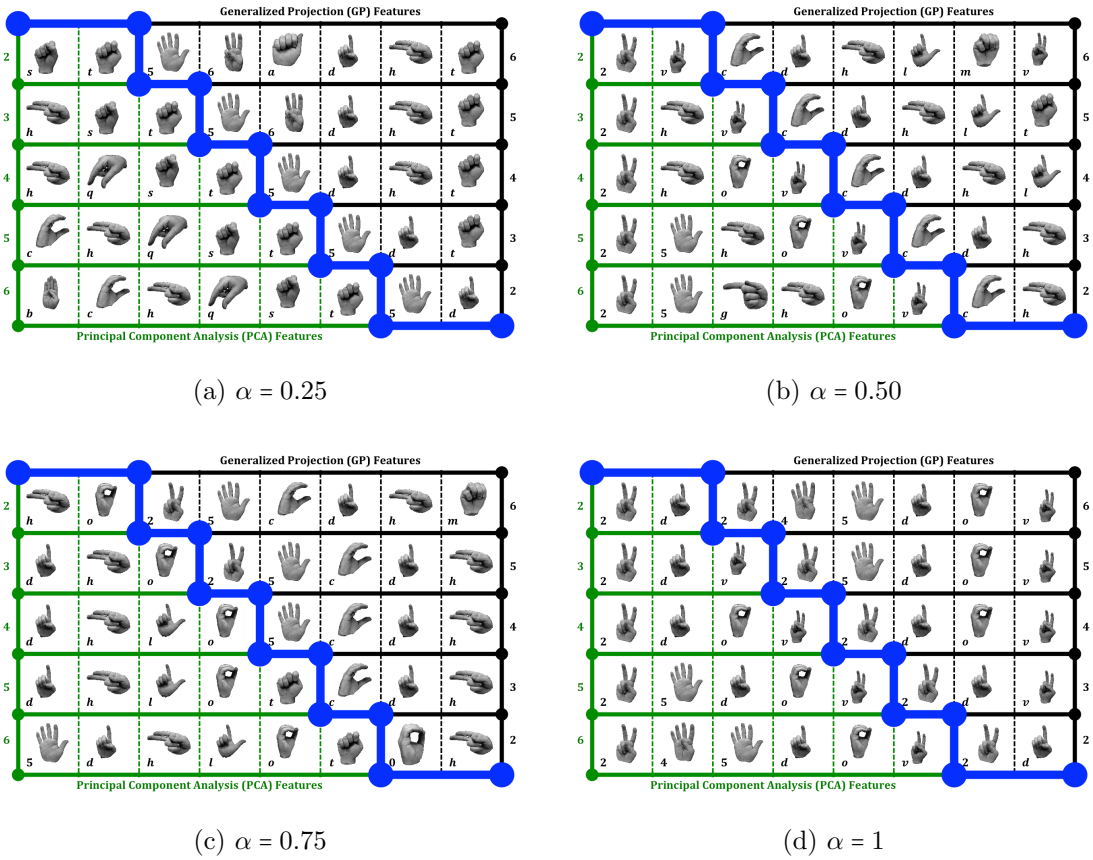
broad technological appeal, such optimal selection must not only be accomplished through an objective algorithm, but must include consumer preference, or subjective constraints, as well.

The manner in which feature data from the gesture images lie in feature space provides excellent clues as to how well a general recognition algorithm will perform. Hence, it is important to choose quality features that are well-clustered near the features of the same class and well-separated from the features of other classes, as is measured by the feature selection weakness (FSW) metric developed in this chapter. Having excellent features diminishes the perils of *over-fitting* the data, and abates the complications of working with high-dimensional feature data. Nonetheless, one often cannot avoid having features that are not well-rendered in feature space, and so some gestures become objectively better for recognition in terms of computational ease.

The *ellipsoidal distance ratio metric* (EDRM) is a measure devised to reward optimal feature clusterings in feature space by assigning a value to each gesture class pairing that accounts for their respective separation compared to their individual clustering sizes. This measure has been shown to match the true average recognition rate results that decide which combination of n gestures is best, which are obtained

Table 3.2: The subjective rankings given to each gesture in the Massey University lexicon are listed, from which the subjective measures are calculated.

0 → 0.97	**1** → 0.10	**2** → 1.00	**3** → 0.30	**4** → 0.95	**5** → 0.96
6 → 0.90	**7** → 0.45	**8** → 0.55	**9** → 0.50	**a** → 0.85	**b** → 0.85
c → 0.95	**d** → 0.99	**e** → 0.85	**f** → 0.60	**g** → 0.90	**h** → 0.95
i → 0.70	**j** → 0.20	**k** → 0.10	**l** → 0.95	**m** → 0.90	**n** → 0.85
o → 0.97	**p** → 0.75	**q** → 0.70	**r** → 0.45	**s** → 0.95	**t** → 0.90
u → 0.95	**v** → 0.98	**w** → 0.90	**x** → 0.55	**y** → 0.30	**z** → 0.30



(a) $\alpha = 0.25$

(b) $\alpha = 0.50$

(c) $\alpha = 0.75$

(d) $\alpha = 1$

Figure 3.4: These are the best lexicons of sizes $n = 2-6$ using the PCA (green) and GP (black) feature extraction methods. The lexicons of size n are read across each row; the blue line separates the two feature selection methods. The relative influence that the subjective considerations for which gestures are better than others is controlled by the weighting factor α. When $\alpha = 1$, the total measure (TM) comes solely from the subjective measure (SM), and is not swayed by the EDRM.

by actually completing entire recognition process. The EDRM value only needs the feature data to provide a well-educated guess as to what is the best lexicon of size n. Given that the EDRM value can easily be combined with subjective considerations

to make a total measure on the quality of a gesture set, it is an attractive means for determining the best gesture lexicons that are most suitable for a given recognition project.

The EDRM is not without some failings. It can suffer from the effects of the *curse of dimensionality*, since the ellipsoid dimensions are determined by the number of features extracted. This means that as the dimensions increase, more data is needed to make accurate predictions about the true recognition rates. Sorting through all the possible combinations of n gestures using the EDRM is an *NP hard* problem, which can quickly become intractable as the size of the entire lexicon of available gestures increases; though, subjective constraints applied to entire set of all viable gestures often can eliminate this problem. Also, a tree-based hierarchy [74] selection scheme can improve the computational efficiency of finding the best gestures.

Plus, the EDRM is designed for filter-based feature selection methods where the features are extracted generally, independent of the classifier. Wrapper-based and embedded feature selection methods are directly tied to the specific classifier and so may not be as generalizable and can be computationally expensive. Along with this, the EDRM method of determining best gestures does not account for more sophisticated learning algorithms that can distinguish classes separated in complicated, nonlinear ways, which may work very well for specific applications.

Chapter 4

CONCLUSION

In this work, methods and algorithms for improving gesture recognition have been introduced and tested. The dynamic mode decomposition (DMD) method for time-scale separation and filtering improves upon the computational efficiency of the robust principal component analysis (RPCA) method for some separation applications, and opens up new applications for the DMD method. It has been demonstrated that pre-processing is key to better gesture recognition performance in terms of robustness, accuracy, and computational efficiency. The ellipsoidal distance ratio metric (EDRM), combined with a simple subjective metric, is able to measure a gesture's likelihood for accurate recognition and its appropriateness for the given application. Finding the best gestures appropriate for the recognition task at hand helps ensure better algorithmic performance and user comfort and suitability.

Now that the DMD method has made real-time background/foreground separation in videos viable, the strategy of optimizing pre-processing becomes much more appealing and feasible for improving gesture recognition performance. Future investigations in the computer vision field can include combining the background subtraction with enhanced pre-processing to make a complete real-time gesture recognition algorithm. Tracking the features of the gestures through time can also lead to gesture anticipation and/or the recognition of dynamic gestures. When dynamics are involved it may be important to examine whether it is better to handle the time component of the recognition process in the feature selection routine or in the statistical learning and classification routines, e.g. using higher order PCA methods [80] for extracting features in time or using the *conditional random field* (CRF) [73] learning scheme to

learn various chains of gestures in time. It would also be worth exploring the idea of using the Fourier frequencies found in the DMD separation as features to dynamic gestures, and as indicators as to when dynamic gestures start and stop, or when a series of static gestures has ended.

The feature selection weakness (FSW) measure presented in this dissertation has been shown to be well-suited for determining which features will be well-clustered and well-separated in feature space, and hence, which features have the strongest influence on accurate gesture recognition. It is also related to the EDRM, where gestures of the same class are rewarded for being well-clustered in feature space, and well-separated from other gesture classes. Subjective measures, determined from ergonomic and vernacular considerations that are mindful of the one who articulates the gestures, and account for ease of articulation, comfort, cultural norms, and suitability to the given application, are easily combined with EDRM value on a consistent scale of $[0, 1]$ in order assess each gesture's quality for the given task. In future investigations, the true predictive power of this total measure on gesture's quality can be examined more thoroughly by implementing this best lexicon finding scheme on more recognition problems and then comparing actual recognition rates to the total measure value.

Future work will focus on using this DMD technique for time-scale separation and filtering in dynamical systems, where it is envisioned that slow time-scale dynamics will be manifested in low magnitude Fourier frequencies of the DMD process, which correspond to DMD terms that can be extracted to make separate solutions of slow and fast time-scales, though it is already known that some separations can made that do not divide Fourier frequencies based on their magnitudes (See Fig. 2.9). A thorough and rigorous analytical study will be needed in order to further identify and understand the conditions under which the DMD method converges to exactly reconstruct the dynamics of a system, or to exactly produce a fast and slow time-scale separation in the system. Such studies may reveal the true reasons for why the DMD error in the foreground video has pixel intensities that are inverted from the

pixel intensities of the moving object(s). The error in the DMD method verses object size, object speed, object acceleration, object intensity/amplitude, number of time snapshots, dimensionality reduction, and other method and input parameters can also benefit from a more in-depth analysis. Other possible applications to the DMD method include compressed sensing, data compression, sparse sampling of features, and normalizing the articulation times for dynamic gestures.

It is also interesting to conjecture on the connection between the proposed DMD technique for performing RPCA, and the matrix completion problem (See [81] and references therein). Potentially, one could use the DMD technique for generating initial conditions for the convex optimization problem of RPCA, thus combining the power and speed of the method with the exact reconstruction ability of RPCA. The possibility of connecting the DMD methodology to the more standard L^1 convex optimization problem of RPCA and the matrix completion problem is also intriguing, and perhaps will allow for even further improvements to be made in background separation methods in terms of both quality and speed. There may also be ties between the DMD method and data compression and/or sparse sensing [9].

BIBLIOGRAPHY

[1] L. G. Shapiro and G. C. Stockman. *Computer Vision*. Prentice Hall, 2001.

[2] T. Morris. *Computer Vision and Image Processing*. Palgrave Macmillan, 2004.

[3] R. Cipolla and A. Pentland. *Computer Vision for Human-Machine Interaction*. Cambridge University Press, 1998.

[4] V. Pavlovic, R. Sharma, and T. Huang. *Visual interpretation of hand gestures for human-computer interaction: A review*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):677–695, 1997.

[5] L. Li, W. Huang, I. Gu, and Q. Tian. *Statistical Modeling of Complex Backgrounds for Foreground Object Detection*. *IEEE Transactions on Image Processing*, 13(11):1459–1472, 2004.

[6] Y. Tian, M. Lu, and A. Hampapur. *Robust and Efficient Foreground Analysis for Real-Time Video Surveillance*. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.*, volume 1, pages 1182–1187, 2005.

[7] L. Maddalena and A. Petrosino. *A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications*. *IEEE Transactions on Image Processing*, 17(7):1168–1177, 2008.

[8] J. He, L. Balzano, and A. Szlam. *Incremental Gradient on the Grassmannian for Online Foreground and Background Separation in Subsampled Video*. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1568–1575, 2012.

[9] E. Candès, X. Li, Y. Ma, and J. Wright. *Robust Principal Component Analysis?* *Computing Research Repository*, abs/0912.3599, 2009.

[10] F. De la Torre and M. Black. *A Framework for Robust Subspace Learning*. *International Journal of Computer Vision*, 54(1-3):117–142, 2003.

[11] P. Huber. *Robust Statistics*. Wiley, 1981.

[12] R. Gnanadesikan and J. Kettenring. *Robust Estimates, Residuals, and Outlier Detection with Multiresponse Data*. *Biometrics*, 28(1):pp. 81–124, 1972.

[13] Q. Ke and T. Kanade. *Robust L_1 norm factorization in the presence of outliers and missing data by alternative convex programming*. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.*, volume 1, pages 739–746, 2005.

[14] M. Fischler and R. Bolles. *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*. *Communications of the Association for Computing Machinery (ACM)*, 24(6):381–395, 1981.

[15] P. Schmid. *Dynamic mode decomposition of numerical and experimental data*. *Journal of Fluid Mechanics*, 656:5–28, 2010.

[16] P. Schmid, L. Li, M. Juniper, and O. Pust. *Applications of the dynamic mode decomposition*. *Theoretical and Computational Fluid Dynamics*, 25(1-4):249–259, 2011.

[17] K. Chen, J. Tu, and C. Rowley. *Variants of Dynamic Mode Decomposition: Boundary Condition, Koopman, and Fourier Analyses*. *Journal of Nonlinear Science*, 22(6):887–915, 2012.

[18] C. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. Henningson. *Spectral analysis of nonlinear flows*. *Journal of Fluid Mechanics*, 641:115–127, 2009.

[19] J. Tu, D. Luchtenberg, C. Rowley, S. Brunton, and J. N. Kutz. *Generalizing dynamic mode decomposition to a larger class of datasets*. *Journal of Computational Dynamics*, 2013. (submitted).

[20] J. N. Kutz. *Data-driven modeling and scientific computing: Methods for Integrating Dynamics of Complex Systems and Big Data*. Oxford Press, 2013.

[21] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld. *Face recognition: A literature survey*. *ACM Comput. Surv.*, 35(4):399–458, 2003.

[22] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. McLachlan, A. Ng, B. Liu, P. Yu, Z. Zhou, M. Steinbach, D. Hand, and D. Steinberg. *Top 10 algorithms in data mining*. *Knowledge and Information Systems*, 14(1): 1–37, 2008.

[23] J. Fan and J. Lv. *A Selective Overview of Variable Selection in High Dimensional Feature Space. Statistica Sinica*, 20(1):101–148, 2010.

[24] H. Wu and A. Sutherland. *Dynamic Gesture Recognition Using PCA with Multi-scale Theory and HMM.* In *In proceedings of Society of Photo-Optical Instrumentation Engineers (SPIE)*, volume 4550, pages 132–139, 2001.

[25] A. Erol, G. Bebis, M. Nicolescu, R. Boyle, and X. Twombly. *Vision-based hand pose estimation: A review. Computer Vision and Image Understanding*, 108: 52–73, 2007.

[26] S. Cobos, M. Ferre, M. Sánchez-Urán, J. Ortego, and R. Aracil. *Human hand descriptions and gesture recognition for object manipulation. Computer Methods in Biomechanics and Biomedical Engineering*, pages 1–13, 2010.

[27] B. Ionescu, D. Coquin, P. Lambert, and V. Buzuloiu. *Dynamic Hand Gesture Recognition Using the Skeleton of the Hand. EURASIP Journal on Advances in Signal Processing*, 13:2101–2109, 2005.

[28] J. Lin, Y. Wu, and T. Huang. *Modeling the Constraints of Human Hand Motion.* In *In IEEE Human Motion Workshop*, pages 121–126, 2000.

[29] X. Zabulis, H. Baltzakis, and A. Argyros. *Vision-based hand gesture recognition for human-computer interaction*, 2009. The Universal Access Handbook. LEA.

[30] P. Viola and M. Jones. *Robust Real-time Object Detection.* In *IEEE ICCV Second Int'l Workshop on Stat. and Comp. Theories of Vision*, volume 20, pages 1254–1259, 2001.

[31] Y. Benezeth, P. Jodoin, B. Emile, H. Laurant, and C. Rosenberger. *Review and Evaluation of Commonly-Implemented Background Subtraction Algorithms.* In *IEEE International Conference on Pattern Recognition*, pages 1–4, 2008.

[32] E. Candès, J. Romberg, and T. Tao. *Robust Uncertainty Principles: Exact Signal Reconstruction From Highly Incomplete Frequency Information. IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[33] *Advanced Video and Signal based Surveillance Datasets*, 2007. URL `http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html`. i-Lids Dataset for AVSS London 2007.

[34] P. Soille. *Morphological Image Analysis: Principles and Applications*. Springer-Verlag, New York, second edition, 2010.

[35] P. Dreuw. *RWTH German Fingerspelling Database*, 2005. URL `http://www-i6.informatik.rwth-aachen.de/~dreuw/fingerspelling.php`.

[36] P. Dreuw, T. Deselaers, D. Keysers, and H. Ney. *Modeling Image Variability in Appearance-Based Gesture Recognition*. In *ECCV Workshop on Statistical Methods in Multi-Image and Video Processing*, pages 7–18, 2006.

[37] *Cambridge Hand Gesture Data set*, 2007. URL `http://www.iis.ee.ic.ac.uk/icvl/ges_db.htm`.

[38] S. Marcel. *Hand Posture and Gesture Datasets*, 2001. URL `http://www.idiap.ch/resource/gestures/`.

[39] H. Birk, T. Moeslund, and C. Madsen. *Real-Time Recognition of Hand Alphabet Gestures Using Principal Component Analysis*. In *10th Scandinavian Conference on Image Analysis*, 1997.

[40] R. Lockton and A. W. Fitzgibbon. *Real-time gesture recognition using deterministic boosting*. In *Proceedings, British Machine Vision Conference*, 2002.

[41] M. Turk and A. Pentland. *Face Recognition Using Eigenfaces*. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[42] M. Turk and A. Pentland. *Eigenfaces for Recognition*. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[43] D. Zhang and Z.-H. Zhou. $(2D)^2PCA$*: Two-directional two-dimensional PCA for efficient face representation and recognition*. *Nanocomputing*, 69:224–231, 2005.

[44] G. C. Feng and P. C. Yuen. *Variance projection function and its application to eye detection for human face recognition*. *Pattern Recognition Letters*, 19: 899–906, 1998.

[45] Z. Zhou and X. Geng. *Projection functions for eye detection*. *Journal of Pattern Recognition*, 37(5):1049–1056, 2004.

[46] G. Feng and P. Yuen. *Multi-cues eye detection on gray intensity image. Journal of Pattern Recognition*, 34(5):1033 – 1046, 2001.

[47] R. Brunelli and T. Poggio. *Face Recognition: Features vs Templates. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.

[48] M. Hu. *Visual Pattern Recognition by Moment Invariants. IRE Transactions Information Theory*, IT-8:179–187, 1962.

[49] J. Flusser. *On the Independence of Rotation Moment Invariants. Pattern Recognition*, 33:1405–1410, 2000.

[50] J. Flusser. *Moment Invariants in Image Analysis. World Academy of Science, Engeering & Technology*, 11:376–381, 2005.

[51] J. Flusser and T. Suk. *Rotation Moment Invariants for Recognition of Symmetric Objects. IEEE Transactions on Image Processing*, 15:3784–3790, 2006.

[52] D. Donoho. *High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality.* In *Math Challenges of the 21st Century.* Stanford University, Aide-Memoire, Los Angeles, CA, 2000. URL `http://www-stat.stanford.edu/~donoho/Lectures/AMS2000/AMS2000.html`.

[53] Y. Freund and R. Schapire. *A Desicion-Theoretic Generalization of On-Line Learning and an Application to Boosting.* In *Computational Learning Theory*, volume 904 of *Lecture Notes in Computer Science*, pages 23–37. Springer Berlin Heidelberg, 1995.

[54] R. Schapire and Y. Singer. *Improved Boosting Algorithms Using Confidence-rated Predictions. Machine Learning*, 37(3):297–336, 1999.

[55] S. Mika, G. R atsch, J. Weston, B. Sch olkopf, and K. M uller. *Fisher Discriminant Analysis with Kernels.* In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, pages 41–48, 1999.

[56] Q. Gu, Z. Li, and J. Han. *Generalized Fisher Score for Feature Selection.* In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, 2011.

[57] D. Koller and M. Sahami. *Toward Optimal feature Selection.* Technical Report, Standford InfoLab, 1996. URL `http://ilpubs.stanford.edu:8090/208/`.

[58] M. Robnik-Šikonja and I. Kononenko. *Theoretical and Empirical Analysis of ReliefF and RReliefF*. *Machine Learning*, 53(1-2):23–69, 2003.

[59] X. He, D. Cai, and P. Niyogi. *Laplacian Score for Feature Selection*. In *Advances in Neural Information Processing Systems 18*, pages 507–514. MIT Press, Cambridge, MA, 2006.

[60] L. Song, A. Smola, A. Gretton, K. Borgwardt, and J. Bedo. *Supervised feature selection via dependence estimation*. In *Proceedings of the 24th International Conference on Machine Learning*, pages 823–830, 2007.

[61] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan. *Trace Ratio Criterion for Feature Selection*. In *Proceedings of the 23rd Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, pages 671–676, 2008.

[62] P. Lisboa and A. Mehri-Dehnavi. *Sensitivity Methods for Variable Selection Using the MLP*. In *International Workshop on Neural Networks for Identification, Control, Robotics and Signal/Image*, pages 330–338, 1996.

[63] Y. Lu, I. Cohen, X. Zhou, and Q. Tian. *Feature Selection Using Principal Feature Analysis*. In *Proceedings of the 15th Annual International Association of Computing Machinery Multimedia Conference*, pages 301–304, 2007.

[64] S. Menard. *Logistic regression: From Introductory to Advanced Concepts and Applications*. SAGE Pub., Inc., 2010.

[65] S. Haykin. *Neural Networks: a comprehensive foundation*. Macmillan, New York, 1994.

[66] V. Vapkin. *Statistical Learning Theory*. Wiley & Sons, 1998.

[67] Y. Freund and R. Schapire. *A Desicion-Theoretic Generalization of On-Line Learning and an Application to Boosting*. *Journal of Computer and System Sciences*, 55:119–139, 1997.

[68] Y. Freund and R. Schapire. *A Short Introduction to Boosting*. *Journal of Japanese Society for Artificial Intelligence*, 14:771–780, 1999.

[69] H. Cooper and R. Bowden. *Large Lexicon Detection of Sign Language*. *Human-Computer Interaction, Lecture Notes in Computer Science*, 4796:88–97, 2007.

[70] M. Russell and R. Moore. *Explicit Modelling of State Occupancy in Hidden Markov Models for Automatic Speech Recognition*. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 5–8, 1985.

[71] F. Samaria and S. Young. *HMM-based architecture for face identification*. *Image and Vision Computing*, 12:537–543, 1994.

[72] A. Nefian and M. Hayes III. *Hidden Markov Models for face identification*. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 2721–2724, 1998.

[73] J. Lafferty, A. McCallum, and F. Pereira. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*. *Proceedings of the International Conference on Machine Learning*, pages 282–289, 2001.

[74] T. Khoshgoftaar, E. Allen, L. Bullard, R. Halstead, and G. Trio. *A Tree-Based Classification Model for Analysis of a Military Software System*. In *IEEE Comp. Soc. Proc. of the IEEE High-Assurance Systems Engineering Workshop*, pages 244–251, 1997.

[75] M. Kölsch and M. Turk. *Analysis of Rotational Robustness of Hand Detection with a Viola-Jones Detector*. In *Proceedings of the 17th International Conference on Pattern Recognition*, 2004.

[76] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta. *A survey of kernel and spectral methods for clustering*. *Pattern Recognition*, 41(1):176 – 190, 2008.

[77] A. Barczak, N. Reyes, M. Abastillas, A. Piccio, and T. Susnjak. *A New 2D Static Hand Gesture Colour Image Dataset for ASL Gestures*. *Research Letters in the Information and Mathematical Sciences*, 15:12–20, 2011. URL `http://www.massey.ac.nz/massey/fms/Colleges/College%20of%20Sciences/IIMS/RLIMS/Volume%2015/GestureDatasetRLIMS2011.pdf`.

[78] A. Barczak, N. Reyes, M. Abastillas, A. Piccio, and T. Susnjak. *Massey University ASL Hand Gesture Dataset*, 2012. URL `http://www.massey.ac.nz/~albarcza/gesture_dataset2012.html`.

[79] J. Grosek, P. Shi, and J. N. Kutz. *Enhanced Gesture Recognition Performance through Improved Pre-Processing*. *International Journal of Computer Applications*, 62(9):1–8, 2013.

[80] L. Eldén. *Matrix Methods in Data Mining and Pattern Recognition*. SIAM, Philadelphia, 2007.

[81] E. Candès and B. Recht. *Exact Matrix Completion via Convex Optimization*. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[82] D. Soh and J. Koplow. *Analysis of Spectral Broadening of Incoherent Light in Optical Fibers with Nonzero Dispersion*. Technical report, Sandia National Laboratories, Livermore, CA, 2010.

[83] Z. Lin, M. Chen, and Y. Ma. *The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices*. Technical Report, University of Illinois at Urbana-Champaign, 2011. URL `http://arxiv.org/abs/1009.5055?context=math`.

[84] M. Pantic and L. Rothkrantz. *Automatic Analysis of Facial Expressions: The State of the Art*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1424–1445, 2000.

[85] Y. Kang. *Calculations and Measurements of Raman Gain Coefficients of Different Fiber Types*. Master's thesis, Electrical Engineering, Virginia Polytechnic Institute and State University, Dec. 2002.

[86] Y. Chen. *Combined processes of stimulated Raman scattering and four-wave mixing in optical fibers*. *Journal of the Optical Society of America B*, 7(1): 43–52, Jan. 1990.

[87] Y. Prior. *A Complete Expression for the Third-Order Susceptibility* ($\chi^{(3)}$)*-Perturbative and Diagrammatic Approaches*. *IEEE Journal of Quantum Electronics*, QE-20(1):37–42, Jan. 1984.

[88] R. Stolen, C. Lee, and J. Clinton. *Development of the stimulated Raman spectrum in single-mode silica fibers*. *Journal of the Optical Society of America B*, 1(4):652–657, Aug. 1984.

[89] A. Boskovic, S. Chernikov, J. Taylor, L. Gruner-Nielson, and O. Levring. *Direct continuous-wave measurement of* n_2 *in various types of telecommunication fiber at 1.55* μm. *Optics Letters*, 21(24):1966–1968, Dec. 1996.

[90] *Dispersion Equations*, 2011. URL `www.cvimellesgriot.com/products/Documents/Catalog/Dispersion_Equations.pdf`.

[91] J. Grosek and J. N. Kutz. *Dynamic Mode Decomposition for Real-Time Background/Foreground Separation in Video. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013. (submitted).

[92] Y. Yatsenko and A. Mavritsky. *D-scan measurement of nonlinear refractive index in fibers heavily doped with GeO_2. Optics Letters*, 32(22):3257–3259, Nov. 2007.

[93] D. Soh, J. Koplow, Moore S., K. Schroder, and W. Hsu. *The effect of dispersion on spectral broadening of incoherent continuous-wave light in optical fibers. Optics Express*, 18(21):22393–22405, Oct. 2010.

[94] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma. *Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 61, 2009.

[95] T. Erdogan. *Fiber Grating Spectra. Journal of Lightwave Technology*, 15(8): 1277–1294, Aug. 1997.

[96] J. Nilsson and G. van der Westhuizen. *Fiber Optical Parametric Oscillator and Amplifier for High Power, High Efficiency Operation at Around 0.8 μm.* Progress report, University of Southampton, UK, Nov. 2008.

[97] G. van der Westhuizen and J. Nilsson. *Fiber Optical Parametric Oscillator for Large Frequency-Shift Wavelength Conversion.* Draft, Southampton, UK, Dec. 2010.

[98] C. Thiel. *Four-Wave Mixing and its Applications.* Technical report, Montana State University, 2008. URL `www.physics.montana.edu/students/thiel/docs/FWMixing.pdf`.

[99] L. Henry, J. Grosek, G. Moore, and T. Shay. Generation of 50w of 1178 nm via amplification of the second stokes. In *Lasers, Sources, and Related Photonic Devices*. Optical Society of America, 2012. URL `http://www.opticsinfobase.org/abstract.cfm?URI=FILAS-2012-FW3C.4`.

[100] V. Grigor'yants, V. Smirnov, and Y. Chamorovskiĭ. *Generation of wide-band optical continuum in fiber waveguides. Kvantovaya Elektron (Moscow)*, 9:1322–1331, July 1982.

[101] L. Henry and J. Grosek. *Investigation of the low power stage of a 1178 nm Raman system.* In *Society of Photo-Optical Instrumentation Engineers (SPIE)*

LASE Conference on Laser Resonators, Microresonators, and Beam Control XVI, 2013. (submitted).

[102] *Low-Rank Matrix Recovery and Completion via Convex Optimization Sample Code*, 2013. URL `http://perception.csl.illinois.edu/matrix-rank/sample_code.html`. Perception and Decision Lab, University of Illinois at Urbana-Champaign, and Microsoft Research Asia, Beijing, Copyright 2009.

[103] J. Fleming. *Material Dispersion in Lightguide Glasses*. *Electronics Letters*, 14 (11):326–328, May 1978.

[104] D. Soh. *Modeling and Simulations on Raman Fiber Laser for Spectral Beam Combining and Consolidation method*. Technical report, Sandia National Laboratories, Livermore, CA, Mar. 2010.

[105] J. Lee and T. Kunii. *Model-Based Analysis of Hand Posture*. *IEEE Computer Graphics and Applications*, pages 77–86, 1995.

[106] R. Sammut and S. Garth. *Multiple-frequency generation on single-mode optical fibers*. *Journal of the Optical Society of America B*, 6(9):1732–1735, Sept. 1989.

[107] V. Vapkin. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[108] G. Agrawal. *Nonlinear Fiber Optics*. Academic Press, San Diego, 4$^{\text{th}}$ edition, 2007.

[109] R. Boyd. *Nonlinear Fiber Optics*. Academic Press, San Diego, 3$^{\text{rd}}$ edition, 2008.

[110] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, Philadelphia, 1997.

[111] J. Hagen, R. Engelbrecht, O. Welzel, A. Siekiera, and B. Schmauss. *Numerical Modeling of Intracavity Spectral Broadening of Raman Fiber Lasers*. *IEEE Photonics Technology Letters*, 19(21):1759–1761, Nov. 2007.

[112] S. Miller and A. Chynoweth. *Optical Fiber Telecommunications*. Academic Press, New York, 1979.

[113] A. Yariv and P. Yeh. *Optical Waves in Crystals: Propagation and Control of Laser Radiation*. John Wiley & Sons, Inc., New Jersey, 2003.

[114] R. Stolen and J. Bjorkholm. *Parametric Amplification and Frequency Conversion in optical Fibers. IEEE Journal of Quantum Electronics*, QE-18(7): 1062–1072, July 1982.

[115] S. Garth. *Phase matching the stimulated four-photon mixing process on single-mode fibers operating in the 1.55-μm region. Optics Letters*, 13(12):1117–1119, Dec. 1988.

[116] Y. Shen. *The Principles of Nonlinear Optics*. John Wiley & Sons, Inc., New York, 2003.

[117] R. Coso and J. Solis. *Relation between nonlinear refractive index and third-order susceptibility in absorbing media. Journal of the Optical Society of America B*, 21(3):640–644, Mar. 2004.

[118] J. Chee and J. Liu. *Raman-assisted parametric frequency and polarization conversion in a birefringent fiber. Optics Letters*, 14(15):820–822, Aug. 1989.

[119] J. W. Nicholson, M. F. Yan, P. Wisk, J. Fleming, F. DiMarcello, E. Monberg, T. Taunay, C. Headley, and D. J. DiGiovanni. *Raman fiber laser with 81 W output power at 1480 nm. Optical Letters*, 35(18):3069–3071, 2010.

[120] C. Denman, P. Hillman, G. Moore, J. Telle, J. Preston, J. Drummond, and R. Fugate. *Realization of a 50-watt facility-class sodium guidestar pump laser.* In *In proceedings of Society of Photo-Optical Instrumentation Engineers (SPIE)*, volume 5707, pages 46–49, 2005.

[121] K. Peng, L. Chen, S. Ruan, and G. Kukharev. *A Robust and Efficient Algorithm for Eye Detection on Gray Intensity Face. Springer-Verlay, Berlin*, pages 302–308, 2005.

[122] L. Henry, T. Shay, G. Moore, and J. Grosek. *Seeded Raman amplifier for applications in the 1100 - 1500 nm spectral region*, 2013. URL `http://patft.uspto.gov/netahtml/PTO/srchnum.htm`. US Patent No. 8472486.

[123] J. Grosek and J. N. Kutz. *Selecting a Small Set of Optimal Gestures from an Extensive Lexicon. Image and Vision Computing*, 2013. (submitted).

[124] J. Bouteiller. *Spectral Modeling of Raman Fiber Lasers. IEEE Photonics Technology Letters*, 15(12):1698–1700, Dec. 2003.

[125] W. Wadsworth, N. Joly, J. Knight, T. Birks, F. Biancalana, and P. Russell. *Supercontinuum and four-wave mixing with Q-switched pulses in endlessly single-mode photonic crystal fibres*. *Optics Express*, 12(2):299–309, Jan. 2004.

[126] I. Dajani, C. Zeringue, T. Bronder, T. Shay, A. Gavrielides, and C. Robin. *A theoretical treatment of two approaches to SBS mitigation with two-tone amplification*. *Optics Express*, 16(18):14233–14247, Sept. 2008.

[127] R. Hellwarth. *Third-Order Optical Susceptibilities of Liquids and Solids*. *Progress in Quantum Electronics*, 5:1–68, 1977.

[128] E. Golovchenko and A. Pilipetskii. *Unified analysis of four-photon mixing, modulational instability, and stimulated Raman scattering under various polarization conditions in fibers*. *Journal of the Optical Society of America B*, 11(1): 92–101, Jan. 1994.

[129] Y. Wu and T. Huang. *Vision-Based Gesture Recognition: A Review*. *Lecture Notes in Computer Science, Gesture Workshop*, pages 1–10, 1999.

[130] P. Wizinowich, D. Mignant, A. Bouchez, R. Campbell, J. Chin, A. Contos, M. van Dam, S. Hartman, E. Johansson, R. Lafon, H. Lewis, P. Stomski, D. Summers, C. Brown, P. Danforth, C. Max, and D. Pennington. *The W. M. Keck Observatory Laser Guide Star Adaptive Optics System: Overview*. *Publications of the Astronomical Society of the Pacific*, 118(840):297–309, 2006.

[131] Y. Feng, L. Taylor, and D. B. Calia. *25 W Raman-fiber-amplifier-based 589 nm laser for laser guide star*. *Optical Express*, 17(21):19021–19026, 2009.

[132] L. Taylor, Y. Feng, and D. B. Calia. *50W CW visible laser source at 589nm obtained via frequency doubling of three coherently combined narrow-band Raman fibre amplifiers*. *Optical Express*, 18(8):8540–8555, 2010.

[133] L. Henry, J. Grosek, G. Moore, and T. Shay. *1121 nm resonator properties and impact on the design of a 1178 nm sodium guidestar laser*. In *In proceedings of Society of Photo-Optical Instrumentation Engineers (SPIE)*, volume 8236, pages 823605 1–13, 2012. URL `http://dx.doi.org/10.1117/12.906867`.

Appendix A

CALCULATIONS, EQUATIONS, & ALGORITHMS

A.1 Separation Methodologies

A.1.1 RPCA Theory

Given a collection of data from a potentially complex, nonlinear system, the RPCA method will seek out any sparse nature within the data, while simultaneously fitting the remaining entries to a low-rank basis. As long as the given data is truly of this nature, in that it does lie on a low-dimensional subspace and has a sparse component, then the RPCA algorithm is guaranteed to perfectly separate the given data $\mathbf{X}$ according to equation (2.1), as has been proven by Candès et al. [9].

The key to the RPCA algorithm is formulating the problem into a tractable, non-smooth convex optimization problem known as *principal component pursuit* (PCP):

$$\begin{aligned} \arg\min \quad & \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 \\ \text{subject to} \quad & \mathbf{X} = \mathbf{L} + \mathbf{S} \end{aligned} \tag{A.1}$$

Here PCP is minimizing the weighed combination of the nuclear norm: $\|\mathbf{M}\|_* := \operatorname{trace}\left(\sqrt{\mathbf{M}^*\mathbf{M}}\right)$ and the L^1-norm: $\|\mathbf{M}\|_1 := \sum_{ij} |m_{ij}|$. The scalar regularization parameter is nonnegative: $\lambda \geq 0$. From the optimization problem (A.1), it can be seen that as $\lambda \to 0$, the low-rank structure will incorporate all of the given data: $\mathbf{L} \to \mathbf{X}$, leaving the sparse structure devoid of anything. It is also true that as λ increases, the sparse structure will embody more and more the original data matrix: $\mathbf{S} \to \mathbf{X}$, as the low-rank structure will commensurately approach the zero matrix [9, 20]. Compare PCP to the traditional PCA method by comparing the optimization problems (A.1) and (A.9).

Effectively, λ controls the dimensionality of the low-rank subspace; however, one does not need to know the rank of $\mathbf{L}$ *a priori*. Candès et al. [9] have shown that the choice

$$\lambda = \frac{1}{\sqrt{\max(n, m)}}, \tag{A.2}$$

where $\mathbf{X}$ is $n \times m$, has a high probability of success at producing the correct low-rank and sparse separation provided that the matrices $\mathbf{L}$ and $\mathbf{S}$ are incoherent, which is the case for many practical applications. Some fine tuning of λ may yield slightly improved results [20].

Though there are multiple methods that can solve the convex PCP problem, the *augmented Lagrange multiplier* (ALM) method stands out as a simple and stable algorithm with robust, efficient performance characteristics. The ALM method is effective because it achieves high accuracies in fewer iterations when compared against other competing methods [9]. Plus, there is an *inexact* ALM variant [94] to the *exact* ALM method [83], which is able to converge in even fewer iterations at the cost of weaker guaranteed convergence criteria. Matlab code that implements these methods, along with a few other algorithms, can be downloaded from the University of Illinois Perception and Decision Lab website [102], and is the very same code implemented throughout this dissertation.

A.1.2 DMD Theory

Given data that is collected in regularly spaced time intervals, the DMD method will approximate the low-dimensional modes of the linear, time-independent *Koopman operator* in order to estimate the potentially nonlinear dynamics of the system. This process is distinct from linearizing the dynamics. The dynamic information is resolved through a similar process as is found in the *Arnoldi algorithm* [110, 20].

Consider having n data points collected at a time, with a total of m samplings in time, evenly spaced by Δt, when the data is collected. Let $\mathbf{x}_j$ be a vector of the

n data points collected at time t_j, $j = 1, 2, \ldots, m$. The data can be grouped into matrices as follows...

$$\mathbf{X}_1^{m-1} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \ldots & \mathbf{x}_{m-1} \end{bmatrix}$$
$$\mathbf{X}_2^{m} = \begin{bmatrix} \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \ldots & \mathbf{x}_{m} \end{bmatrix}$$

The Koopman operator $\mathbf{A}$ maps the data at time j to the data at time $j+1$ such that $\mathbf{x}_{j+1} = \mathbf{A}\mathbf{x}_j$. The DMD algorithm will estimate the Koopman operator $\mathbf{A}$ that best represents the data in $\mathbf{X}_1^{m-1}$ such that the columns of

$$\mathbf{X}_1^{m-1} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{A}\mathbf{x}_1 & \mathbf{A}^2\mathbf{x}_1 & \ldots & \mathbf{A}^{m-2}\mathbf{x}_1 \end{bmatrix}$$

form a Krylov space. Thus, $\mathbf{A}\mathbf{X}_1^{m-1} = \mathbf{X}_2^m$. The last data point $\mathbf{x}_m$ is determined by

$$\mathbf{x}_m = \sum_{j=1}^{m-1} k_j \mathbf{x}_j + \mathbf{r},$$

where the k_j's are the coefficients of the Krylov space basis vectors and the residual error $\mathbf{r}$ is orthogonal to the Krylov space [20].

Notice that $\mathbf{X}_2^m = \mathbf{X}_1^{m-1}\mathbf{S} + \mathbf{r} \cdot \mathbf{e}_{m-1}^*$, where $[^*]$ is the conjugate transpose operator, $\mathbf{e}_{m-1}$ is the $(m-1)^{\text{th}}$ unit vector, and

$$\mathbf{S} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & k_1 \\ 1 & 0 & \cdots & 0 & 0 & k_2 \\ 0 & 1 & \ddots & 0 & 0 & k_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & k_{m-2} \\ 0 & 0 & \cdots & 0 & 1 & k_{m-1} \end{bmatrix}$$

is the $n \times (m-1)$ *companion* matrix to $\mathbf{A}$. Also, note that the *singular value decomposition* (SVD) of the matrix $\mathbf{X}_1^{m-1}$ can be used for dimensionality reduction, and for the ease and convenience of working with unitary and diagonal matrices. Thus, $\mathbf{X}_1^{m-1} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$, where $\mathbf{U} \in \mathbb{C}^{n \times \ell}$ is unitary, $\mathbf{\Sigma} \in \mathbb{C}^{\ell \times \ell}$ is diagonal, and $\mathbf{V} \in \mathbb{C}^{(m-1) \times \ell}$

is unitary. The parameter ℓ is chosen so as to reduce rank of $\mathbf{X}_1^{m-1}$ as much as possible, while still capturing the fundamental structure and dynamics of the system represented by the data in $\mathbf{X}_1^{m-1}$. Thus, assuming that the residual error is small ($\|\mathbf{r}\| \ll 1$), one can estimate

$$\begin{aligned}\mathbf{X}_2^m &\approx \mathbf{X}_1^{m-1}\mathbf{S} \\ \mathbf{X}_2^m &\approx \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*\mathbf{S} \\ \mathbf{S} &\approx \mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{U}^*\mathbf{X}_2^m.\end{aligned}$$

Using the similarity transform $\mathbf{V}\boldsymbol{\Sigma}^{-1}$, the matrix

$$\tilde{\mathbf{S}} \approx \mathbf{U}^*\mathbf{X}_2^m\mathbf{V}\boldsymbol{\Sigma}^{-1} = \mathbf{U}^*\mathbf{A}\mathbf{U}$$

can be derived, which is mathematically similar to the matrix $\mathbf{S}$ [20]. Note that this transformation effectively projects the Koopman operator $\mathbf{A}$ onto a lower-dimensional *proper orthogonal decomposition* (POD) basis. This adds stability to this algorithm, especially considering that $\mathbf{X}_1^{m-1}$ can be rank-deficient. The Koopman operator is never actually calculated because all of the dynamics are extracted from the data itself, allowing for *equation-free* or *model-free* applications. However, since the DMD method does not attempt to orthogonalize the Krylov space, as is done in the Arnoldi iteration, there is also a reduction in algorithmic stability and convergence properties of the DMD method [15].

Essential to the DMD method is the idea that because $\mathbf{A}\mathbf{X}_1^{m-1} = \mathbf{X}_2^m \approx \mathbf{X}_1^{m-1}\mathbf{S}$, then some of the eigenvalues of the matrix $\mathbf{S}$ approximate the eigenvalues of the Koopman operator $\mathbf{A}$, similar to the calculation done in the Arnoldi algorithm to get the *Ritz values* and *Ritz vectors* [15, 18]. In fact, the Arnoldi iteration can be recovered by a *QR-factorization*, $\mathbf{X}_1^{m-1} = \mathbf{Q}\mathbf{R}$, and by the relation $\mathbf{H} = \mathbf{R}\mathbf{S}\mathbf{R}^{-1}$, where $\mathbf{H}$ is an upper Hessenberg matrix [110]. In practice, the eigenvalues can be determined through the matrix $\tilde{\mathbf{S}}$ because it is similar to the matrix $\mathbf{S}$. These eigenvalues regulate the time dynamics of the Koopman operator as it pushes the reconstructed data

forward in time by Δt. Thus, solving the eigenvalue problem:

$$\tilde{\mathbf{S}}\mathbf{w}_j = \mu_j \mathbf{w}_j, \;\; j = 1, 2, \ldots, \ell \tag{A.3}$$

will yield the DMD eigenvalues μ_j and eigenvectors $\mathbf{w}_j$. From here the j^{th} DMD basis function mode comes from the eigenvectors of the Koopman operator $\mathbf{A}$:

$$\boldsymbol{\varphi}_j = \mathbf{U}\mathbf{w}_j, \tag{A.4}$$

each of which can be put into the columns of a matrix $\boldsymbol{\Phi}$ [15]. If the residual is zero ($\mathbf{r} \equiv \mathbf{0}$), then the linear Koopman operator should have the exact same eigenvalues to those of the matrix $\tilde{\mathbf{S}}$:

$$\mathbf{A}\,(\mathbf{UW}) = (\mathbf{UW})\,\boldsymbol{\Upsilon},$$

where $\boldsymbol{\Upsilon}$ is the diagonal matrix of the eigenvalues μ_j, $\mathbf{W}$ is the matrix of eigenvectors $\mathbf{w}_j$, and $\boldsymbol{\Phi} = \mathbf{UW}$. For the benefit of predicting the time dynamics, the DMD eigenvalues can be converted to Fourier modes by

$$\omega_j = \frac{\ln(\mu_j)}{\Delta t}. \tag{A.5}$$

The real part of ω_j regulates the growth or decay of the DMD basis function modes, while the imaginary part of ω_j drives oscillations in the DMD modes. Clearly, any unwarranted growth or decay in the Fourier modes will eventually limit the range in time that the model can credibly be predictive [20].

Therefore, the DMD reconstruction of the data $\mathbf{x}_{\text{DMD}}$ at time t for any time after the initial data vector $\mathbf{x}_1$ was collected ($t_1 = 0$) is given by

$$\mathbf{x}_{\text{DMD}}(t) = \sum_{j=1}^{\ell} b_j \boldsymbol{\varphi}_j e^{\omega_j t} = \boldsymbol{\Phi}\boldsymbol{\Omega}^t \mathbf{b}, \tag{A.6}$$

where

$$\boldsymbol{\Omega} = \begin{bmatrix} e^{\omega_1} & 0 & \cdots & 0 \\ 0 & e^{\omega_2} & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{\omega_\ell} \end{bmatrix},$$

and the vector $\mathbf{b}$ contains the initial amplitudes for the modes. At the time that the first snapshot of data was taken ($t_1 = 0$), equation (A.6) reduces to

$$\mathbf{x}_{\mathrm{DMD}}(t_1) = \mathbf{x}_1 = \mathbf{\Phi}\mathbf{b}; \tag{A.7}$$

hence,

$$\mathbf{b} = \left(\mathbf{\Phi}^*\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^*\mathbf{x}_1 \text{ [20]}. \tag{A.8}$$

Note that the DMD method demands that data is collected over evenly spaced time intervals Δt, but it cannot know for sure whether this was done in practice or not; it must deal with the data that it is given, assuming regular time intervals. This implies that there must be certain types of dynamics in time that the DMD method cannot model well even when the data is collected in periodic intervals. This is because there can be one-to-one mappings from certain types of dynamics taken at irregular time intervals to transformed dynamics taken at evenly spaced intervals, which can be accomplished through some kind of frame-of-reference transform.

Also, it is important to consider that, in general, the DMD method only provides *limited*, future (long-time) state predictions to the given system, as often approximations to the eigenvalues leads to physically unrealistic growth or decay. But for short-time or moderate-time future state predictions, the approximation to the eigenvalues can be reasonable. Of course, the DMD algorithm most reliably reconstructs the data that it is given with the dimensionality reduction determined by the parameter ℓ, accumulating any error in the final time step $\mathbf{x}_m$.

A.2 Gesture Recognition Algorithms

A.2.1 PCA Based Feature Selection

Principal component analysis (PCA) is well-known for its ability in dimensionality reduction tasks. Mathematically, PCA decomposes the original image space into a set of orthogonal, low-rank basis vectors. The set of basis vectors are sorted in descending

order of their *energy*, which reflects the average magnitude of its weight among all images. If the first several principal components contain nearly all of the energy, the images can be well approximated in the space spanned by the first few principal components, thus reducing the dimensionality of the problem. Such dimensionality reduction is highly desired in practice as it can be used reducing the computational strain of the given algorithms, while retaining all the pertinent information needed for processing.

PCA can be seen as an optimization problem where the best low-rank estimate $\mathbf{L}$ of the matrix $\mathbf{X}$ is sought such that that rank of $\mathbf{L}$ is no larger than ℓ:

$$\begin{aligned} &\arg\min \quad \|\mathbf{X} - \mathbf{L}\|_2 \\ &\text{subject to} \quad \text{rank}(\mathbf{L}) \leq \ell \end{aligned} \tag{A.9}$$

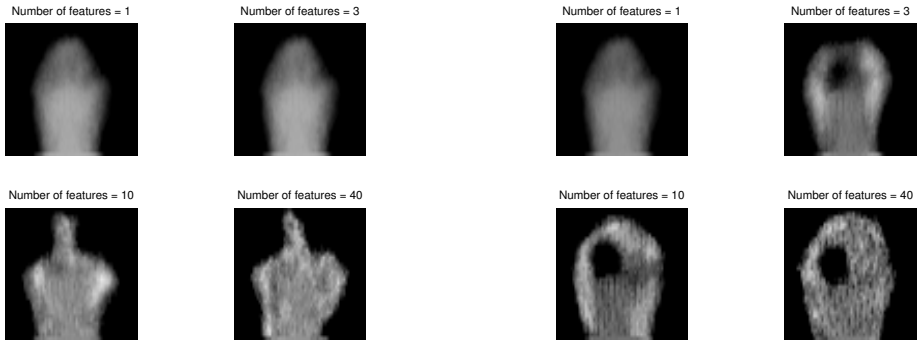
Here $\|\mathbf{Y}\|_2$ is the 2-norm of the matrix $\mathbf{Y}$, which is the largest singular value of $\mathbf{Y}$ [9].

In terms of feature selection, PCA is applied to like-gestures from a training set of images, producing the principal components of those similar images. The linear coefficients associated with those principal components are then used as features that describe the images from which they were calculated. In fact, PCA has been used previously [24, 39, 40] as a feature selection method for recognizing gestures. Real-time gesture recognition methods for the American Sign Language (ASL) alphabet [39, 40] have taken advantage of low-dimensionality through PCA and eigenfaces, which are well-known in the face recognition field [42, 41].

After pre-processing, each image is reshaped into a long column vector. The training set is stored as a $n_{\text{pix}} \times N_{\text{train}}$ matrix $\mathbf{X}$, where n_{pix} is the number of pixels per pre-processed image and N_{train} is the number of images in the training set. In a standard PCA, the mean of each column is subtracted out, producing $\bar{\mathbf{X}}$, and apply the *singular value decomposition* (SVD), as follows...

$$\bar{\mathbf{X}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*. \tag{A.10}$$

The resulting unitary matrix $\mathbf{U} \in \mathbb{C}^{n_{\text{pix}} \times \ell}$ contains the principal components of the



(a) One finger gesture.

(b) O-shape gesture.

Figure A.1: The principal components of the hand gesture with just the index finger extended (a) and the O-shaped hand gesture (b) as a function of the number of features determined from the PCA analysis. The number of principal components used are $\ell = 1, 3, 10$, and 40. Note that approximately 10 features are required to easily resolve gesture (a), whereas only 3 features are needed to visually identify gesture (b).

system in its columns. Entries of the diagonal matrix $\mathbf{\Sigma} \in \mathbb{C}^{\ell \times \ell}$ contain the singular values corresponding to those principal components. The weight coefficients of each image on the principal components are contained in the columns of the unitary matrix $\mathbf{V} \in \mathbb{C}^{\ell \times N_{\text{train}}}$, and $[^*]$ is the conjugate transpose. The parameter ℓ determines the dimensionality reduction, or, in the case of feature selection, determines the number of features to be extracted. Such a decomposition is guaranteed to exist for any matrix $\mathbf{X}$ [110]. The features are extracted as the columns of $\mathbf{V}$ that correspond to the ℓ-most energetic principal components.

In Fig. A.1, the reconstructed images of two typical hand gestures are depicted using $\ell = 1, 3, 10$, and 40 of the most significant principal components. In general, the first few principal components can reconstruct the general shape of the gesture, but

not so well as to be able to distinguish between the different gestures. It has been discovered empirically that choosing about 10 principal components, or 10 features to be extracted, seems to be sufficient in order to reconstruct most hand gestures with sufficient distinction so as to make them clearly recognizable from one another (See objects (h)-(l) in the middle panel of Fig. 2.10).

In the training process of the gesture recognition scheme, the goal is not to extract features, but to build the projection matrix $\mathbf{P} \in \mathbb{R}^{\ell \times n_{\text{pix}}}$ that will project test images $\tilde{\mathbf{X}} \in \mathbb{R}^{n_{\text{pix}} \times 1}$ onto the matrix $\mathbf{V}$ where the features are stored. This is accomplished by rearranging equation (A.10) such that

$$\begin{aligned} \mathbf{P}\tilde{\mathbf{X}} &= \mathbf{V}^* \\ \mathbf{P} &:= \mathbf{\Sigma}^{-1}\mathbf{U}^* \end{aligned} \tag{A.11}$$

Thus, in the recognition application, each the pre-processed gesture image is vectorized as $\tilde{\mathbf{X}} \in \mathbb{R}^{n_{\text{pix}} \times 1}$, and the features of the gesture are calculated according to equation (A.11), and stored in the vector $\mathbf{V}^* \in \mathbb{R}^{\ell \times 1}$.

One drawback to the PCA feature extraction technique is that the *energies* of the PCA components have a fat-tailed distribution, thus potentially preventing an efficient dimensionality reduction. This is because the PCA representation of the image is too sensitive to slight variations in the shape and positioning of the gesture within the image frame. For human beings, a slight translation, dilation, or rotation of a gesture does not effect a person's recognition capability; however, these variations translate into large variations of pixel intensities in the regions of the image that change. This will inevitably generate a component in the PCA analysis which is purely noise. Or said in a different way, it will take more features to adequately resolve the gestures (See Fig. A.1).

There have been attempts to robustify the PCA method [9, 10, 11, 12, 13, 14] and to draw upon 2D correlations between pixels, in a 2D-PCA method [43], in order to improve the quality of the features that get selected through the PCA process.

In spite of these improvements, the PCA method and its variants have limitations associated with its robustness in real-world applications. This dissertation will show that improved pre-processing techniques can avoid these limitations and keep the PCA algorithm as a viable and efficient feature selection method.

A.2.2 Generalized Projections

An alternative feature selection method is called *generalized projections* (GP). This method is motivated by the multi-view orthographic projections in engineering, and has traditionally been implemented on eye tracking algorithms [44, 45, 46, 47]. Like the PCA feature extraction method, generalized projections are also easily understood and implemented, and are computationally efficient.

A generalized projection is constructed from a mixing of integral and variance projections. Suppose a given gesture lies inside of a canvas domain, e.g. an image frame, such that the gesture extends across the region of $[x_1, x_2] \times [y_1, y_2]$, where the intensity of any particular pixel in this domain is $I(x, y)$. Then vertical and horizontal integral projections are given by

$$\mathrm{IPF_v}(x) = \frac{1}{y_2 - y_1} \sum_{y_i=y_1}^{y_2} I(x, y_i)$$
$$\mathrm{IPF_h}(y) = \frac{1}{x_2 - x_1} \sum_{x_i=x_1}^{x_2} I(x_i, y)$$

and the vertical and horizontal variance projections are given by

$$\mathrm{VPF_v}(x) = \frac{1}{y_2 - y_1} \sum_{y_i=y_1}^{y_2} \left[I(x, y_i) - \mathrm{IPF_v}(x)\right]^2$$
$$\mathrm{VPF_h}(y) = \frac{1}{x_2 - x_1} \sum_{x_i=x_1}^{x_2} \left[I(x_i, y) - \mathrm{IPF_h}(y)\right]^2 .$$

The integral and variance projections are mixed together with the parameter $\alpha \in [0, 1]$ such that

$$\mathrm{GPF_v}(x) = (1 - \alpha)\mathrm{IPF_v}(x) + \alpha\mathrm{VPF_v}(x)$$
$$\mathrm{GPF_h}(y) = (1 - \alpha)\mathrm{IPF_h}(y) + \alpha\mathrm{VPF_h}(y) .$$

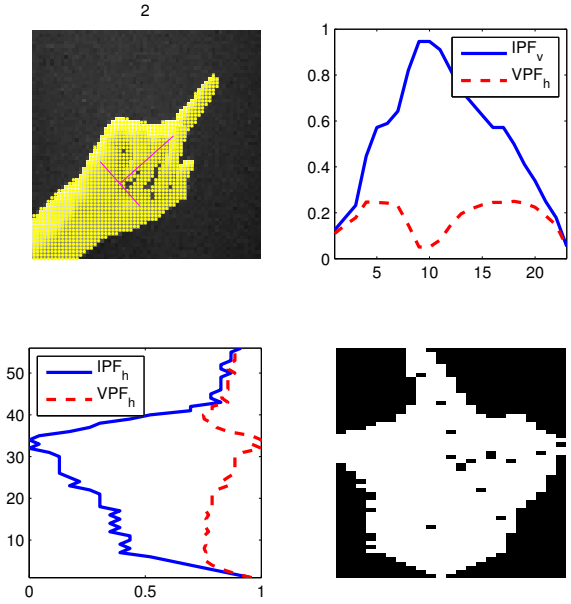


Figure A.2: Original hand shape with the background subtracted, and the principal direction indicated in red, is depicted in the top left panel. The vertical integral and variance projections are computed and shown in the top right panel, while the horizontal integral and variance projections are computed and shown in the bottom left panel. The lower right panel shows the image after pre-processing, which is used to compute the projections.

Figures A.2 and A.3 show the integral and variance projections applied to hand gesture images. One can see in these figures that both the integral and variance projections contain relevant information about the shape of the hand, suggesting that mixing the information can increase the distinguishing power of the resulting features that are extracted from the image.

Features are extracted by selecting a desired number of vertical and horizontal generalized projection values, and accumulating those values into a single feature

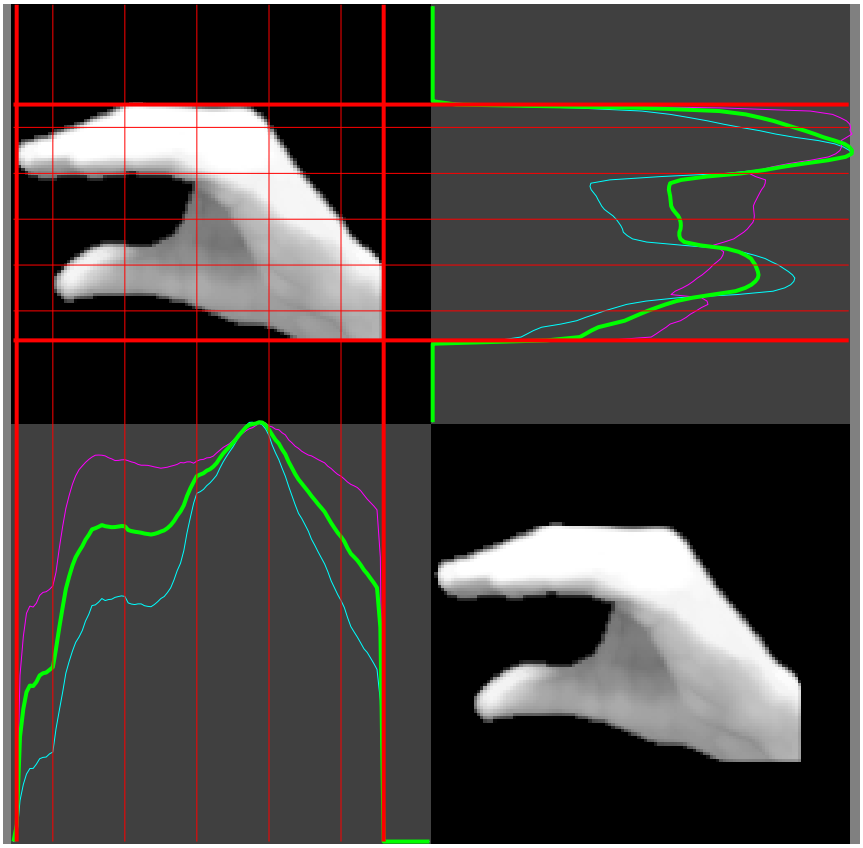


Figure A.3: This figure depicts the generalized projections feature selection routine for another hand gesture. The pre-processed hand gesture (top left and bottom right panels) has five features extracted in both the horizontal and vertical directions. The vertical integral (cyan) and variance (magenta) projections are computed and shown in the bottom left panel, while the horizontal integral (cyan) and variance (magenta) projections are computed and shown in the top right panel. These projections are mixed together ($\alpha = 0.5$) to produce the generalized projections (green).

vector (See objects (m) and (n) in the middle panel of Fig. 2.10). This is accomplished by dividing the (vertical or horizontal) side length of the domain into a desired number of bins, and averaging the generalized projection values within those bins, producing

a single value for each bin. If one desires ℓ features per gesture, then the feature vector $\mathbf{f}$ would look like

$$\mathbf{f} = \Big[\mathrm{GPF_h}(y_1) \cdots \mathrm{GPF_h}(y_{\ell/2})\ \mathrm{GPF_v}(x_1) \cdots \mathrm{GPF_v}(x_{\ell/2})\Big]^*,$$

where $[^*]$ is the transpose, and $\ell/2$ projection bins are sampled both vertically and horizontally at $\{x_1, x_2, \ldots, x_{\ell/2}\}$ and $\{y_1, y_2, \ldots, y_{\ell/2}\}$, respectively. These projections, sampled on a coarse grid, provide a low-dimensional representation of the gesture image. Also, as desired, the generalized projections for different gestures yield distinct feature vectors that can be readily distinguished from one another in a gesture recognition scheme.

A variation of the generalized projections method can be created by taking the derivatives (or gradients) of the integral and variance projection functions, and then mixing them together into *derivatives of generalized projection* (DGP) functions. The derivative would effectively gather information on the rates at which the integral and variance projection values are changing as one scans across the sides of the domain. This information can also be quite unique for each type of gesture, thus also qualifying as excellent candidates for gesture features.

The generalized projections method, and its derivative variant, have the benefit of being translationally and scale invariant, which makes them fairly robust to some of the common types of gesture *noise* that occurs in images. However, these projection methods are still quite sensitive to rotational variations of the gesture within their image frames. This inspires yet another variation to the generalized projections method that takes the integral and variance projections around concentric circles, centered in the middle of the gesture, which is determined by weighted least-squares of the pixel intensities. Here one must be careful to maintain consistency in the projection values by a normalizing the number of pixels in each circle. Also, it is important to start the projection at the beginning of an intersection between the circle and the getsure region if possible, in order to maintain consistent variance projection values.

Again, feature values are averages over bins that consist of several concentric circles. This *circular generalized projections* (CGP) method, also called *signatures*, can be designed to be translationally, scale, and rotationally invariant, but can be limited by very low resolution images. Circular projections can also include other important feature data such as the number of times the circles intersect with the gesture region, and the normalized widths of those intersection regions.

A.2.3 Image Moments

Related to the generalized projections is another feature extraction method called *image moments* (IM) [48, 49, 50, 51], or sometimes called *invariant moments*. Normalized and Hu-image moments are all translation, rotation, and scale invariant. One of the image moments is literally the moment of inertia of the two-dimensional image shape; the other image moments are related.

There are 11 independent moment invariants that are recommended by Flusser [49] for the task of 2D object recognition:

$$\begin{aligned}
\psi_1 &= c_{11} & \psi_7 &= c_{22} \\
\psi_2 &= c_{21}c_{12} & \psi_8 &= \mathrm{Re}\left(c_{31}c_{12}^2\right) \\
\psi_3 &= \mathrm{Re}\left(c_{20}c_{12}^2\right) & \psi_9 &= \mathrm{Im}\left(c_{31}c_{12}^2\right) \\
\psi_4 &= \mathrm{Im}\left(c_{20}c_{12}^2\right) & \psi_{10} &= \mathrm{Re}\left(c_{40}c_{12}^4\right) \\
\psi_5 &= \mathrm{Re}\left(c_{30}c_{12}^3\right) & \psi_{11} &= \mathrm{Im}\left(c_{40}c_{12}^4\right) \\
\psi_6 &= \mathrm{Im}\left(c_{30}c_{12}^3\right) & &.
\end{aligned}$$

Given the image pixel coordinates (x, y) and the corresponding pixel intensities $I(x, y)$, the complex moment c_{pq} is given by

$$c_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x+iy)^p(x-iy)^q I(x,y)\; dxdy,$$

where i is the imaginary number. The resulting image moments feature vector $\mathbf{f}$

would look like

$$\mathbf{f} = \left[\psi_1 \ \psi_2 \ \cdots \ \psi_{11}\right]^*,$$

where $[^*]$ is the transpose.

Like the PCA and generalized projections feature selection methods, and their variants, the image moments method can suffer greatly from not fully and properly segmenting the gesture from its background. Any erroneous background pixels that are not eliminated by the end of the pre-processing step become a type of noise in the feature extraction process that can skew the feature values, broadening the regions that each gesture occupies in feature space and making distinct gestures overlap in feature space (See Fig. 2.12). This will then translate into worse successful recognition rates.

A.2.4 Linear Discriminant Analysis

Two-class linear discrimination analysis (LDA) is an optimization that attempts to maximize the inter-class separation, while minimizing the intra-class variation, for two different types of gestures. This optimization has an analytic solution that requires a generalized eigenvalue problem to be solved. Ultimately, LDA will find a projection vector for the two classes that are being compared, which projects the gesture features to a scalar value. This scalar value, compared against a chosen threshold value, is then used to discriminate between the two classes. In reality, the method described here is called *Fisher's linear discriminant analysis*, which is a generalization of linear discriminant analysis. However, these terms have often been a used synonymously throughout the literature, and in this dissertation, the term linear discriminant analysis will always be used [55, 20].

Consider having features vectors (points) $\mathbf{f}$ that belong to one of two classes; $\mathbf{f} \in C_k, \ k = 1, 2$, there being n_1 feature points in C_1 and n_2 feature points in C_2. The number of features collected per gesture (ℓ) determines the dimensionality of the

feature points: $\mathbf{f} \in \mathbb{R}^\ell$. The center of mass and the covariance of the feature vectors for each class are given by

$$\mu_1 = \frac{1}{n_1} \sum_{\mathbf{f} \in C_1} \mathbf{f} \qquad \mathbf{\Sigma}_{C_1} = \frac{1}{n_1} \sum_{\mathbf{f} \in C_1} \left[(\mathbf{f} - \mu_1) (\mathbf{f} - \mu_1)^* \right]$$
$$\mu_2 = \frac{1}{n_2} \sum_{\mathbf{f} \in C_2} \mathbf{f} \qquad \mathbf{\Sigma}_{C_2} = \frac{1}{n_2} \sum_{\mathbf{f} \in C_2} \left[(\mathbf{f} - \mu_2) (\mathbf{f} - \mu_2)^* \right] \text{ [55].}$$

In one view of LDA, the goal can be thought of as an attempt to find a common weighting for the feature vectors of both classes that maximizes a signal-to-noise ratio, which measures the ratio of the weighted variation between the classes to the weighted variation within the classes. This weighting can be put into the vector $\mathbf{w} \in \mathbb{R}^\ell$, which, for now, can be thought of having the property $\sum_j w_j = 1$. In reality, this constraint is not needed, but it does make the notation easier. The weighted averages of the feature data for both classes are $\mathbf{w}^* \mu_1$ and $\mathbf{w}^* \mu_2$, and weighted covariances of the feature points from both classes are $\mathbf{w}^* \mathbf{\Sigma}_{C_1} \mathbf{w}$ and $\mathbf{w}^* \mathbf{\Sigma}_{C_2} \mathbf{w}$ [55].

The variation between the classes $\mathbf{S}_\mathrm{B}$ and the variation within the classes $\mathbf{S}_\mathrm{W}$ are given by

$$\mathbf{S}_B = (\mu_2 - \mu_1)(\mu_2 - \mu_1)^*$$
$$\mathbf{S}_W = \mathbf{\Sigma}_{C_1} + \mathbf{\Sigma}_{C_2} = \sum_{k=1}^{2} \sum_{\mathbf{f} \in C_k} \left[(\mathbf{f} - \mu_k)(\mathbf{f} - \mu_k)^* \right] .$$

Reinserting the weightings, the signal-to-noise ratio is given by the Rayleigh quotient:

$$R(\mathbf{S}_B, \mathbf{S}_W; \mathbf{w}) = \frac{\mathbf{w}^* \mathbf{S}_\mathrm{B} \mathbf{w}}{\mathbf{w}^* \mathbf{S}_\mathrm{W} \mathbf{w}}$$

Maximizing this ratio leads to the LDA optimization problem,

$$\arg\max_{\mathbf{w}} \frac{\mathbf{w}^* \mathbf{S}_\mathrm{B} \mathbf{w}}{\mathbf{w}^* \mathbf{S}_\mathrm{W} \mathbf{w}},$$

which can be shown to be equivalent to solving the generalized eigenvalue problem

$$\mathbf{S}_\mathrm{B} \mathbf{w} = \lambda \mathbf{S}_\mathrm{B} \mathbf{w} \tag{A.12}$$

for the largest eigenvalue λ and its corresponding eigenvector $\mathbf{w}$ [20].

Another way to view LDA is to think of the vector $\mathbf{w}$ as an orthogonal vector to a discriminant hyperplane that separates the two classes in the best way possible such that the feature points from these classes, projected orthogonally onto this hyperplane, will be as clustered within their respective classes and spaced apart from the other class as much as possible. Thus, and given feature $\mathbf{f}$ will be projected onto the hyperplane by $\mathbf{w}^*\mathbf{f} = c$, where $c \in \mathbb{R}$ is a constant that describes the position on the hyperplane when viewed from the side, as if the hyperplane were a line. Based upon the clusterings of the values of c for each class, a threshold value can be set that will discriminate between the two classes. Since the values of c are shifted about based upon the scaling of the weighting vector $\mathbf{w}$, then one can always design the threshold to be at zero, where $c > 0$ means the feature point belongs to C_1 and $c \leq 0$ means the feature vector belongs to C_2 (See object (o) in the right panel of Fig. 2.10).

In the statistical learning step of the recognition process, the goal is to *train* $\mathbf{w}$ and to set a reasonable threshold that distinguishes the two classes. Or, in other words, the projection vector $\mathbf{w}$ is found by solving the generalized eigenvalue equation (A.12) for the eigenvector that corresponds to the maximal eigenvalue. In the testing or application phase, feature vectors $\mathbf{f}$ are dotted with $\mathbf{w}$ to produce constants c, which can then be compared to the thresholds in order to classify the gestures that correspond to the feature points.

When there are multiple classes $(C_k,\ k = 1, 2, \ldots, K)$, then one can either implement a pairwise or one-versus-the-rest testing strategy. The pairwise testing strategy implements the LDA classification scheme on every possible pairing of two classes, searching for the one class to which the given gesture's feature point belongs. The one-vs-the-rest testing strategy only needs to compare class j (C_j) verses the rest of the classes $(C_{k\neq j})$ for each class j. Also, one has the option of using a *multi-class LDA* statistical learning and classification method, which can distinguish between many classes instead of just two. The multi-class LDA scheme is outside to scope of

this dissertation, and so is left without any explanation.

A.2.5 Logistic Regression

Like most statistical learning methods, logistic regression relies on an iterative optimization procedure to learn the best parameters for distinguishing between two gesture classes. This method projects gesture feature values onto a probabilistic scale $[0, 1]$ that measures the probability of a gesture being in a certain class. The logistic regression is closely related to LDA, in the sense that they both use a hyperplane to separate the two classes. However, since the logistic regression provides a measure of the resemblance between the given gesture and the two possible classes, it is more powerful qualitatively. The fact that the logistic regression algorithm must solve a convex optimization problem using an iterative numerical solver means that it may not converge properly, depending on the given gesture data, the initial guess to the solution, and to the numerical instabilities of working with exponential functions. This is compared with the LDA method which has an analytical solution to its optimization problem, which guarantees the best solution possible to given the gesture data.

Two-class logistic regression uses the logistic function $p(z)$ to measure the likelihood of an object being in class 1 (C_1):

$$p(z) = \frac{1}{1 + \exp(-z)} \tag{A.13}$$

where z is the *logit*, defined as the logarithm of the odds that the given gesture belongs to C_1, and not to C_2. If one assumes that z is linearly related to the independent predictor variables $f_1, \cdots, f_\ell$, then $p(z)$ is the probability of being in C_1 such that

$$z = \beta_0 + \sum_{j=1}^{\ell} \beta_j f_j.$$

In the gesture recognition problem, $\mathbf{f} = [f_1 \ f_2 \ \cdots \ f_\ell]^*$ is the feature vector, and $\boldsymbol{\beta} = [\beta_0 \ \beta_1 \ \cdots \ \beta_\ell]$ are the regression coefficients to be learned from the training data.

Thus, $z = \boldsymbol{\beta}^* \tilde{\mathbf{f}}$, where $\tilde{\mathbf{f}} = [1 \ \mathbf{f}]^*$. From the geometric point of view, z is actually the distance from the object to a class separating hyperplane. If $z = 0$, then $p(0) = 0.5$, meaning that the gesture has a 50-50 chance of being in class 1 and not in class 2 [64].

To compute the best set of β_j's, one can define the *log-likelihood function*

$$\mathrm{LL}(\boldsymbol{\beta}) = \sum_{j=1}^{N} \left[\ln \left(p(\mathbf{f}_j | \boldsymbol{\beta}) \right) \right]$$

where $\mathbf{f}_j$ contains the features that correspond to the j^{th} gesture, there being N total gestures in C_1 and C_2. The conditional probability

$$p(\mathbf{f}_j | \boldsymbol{\beta}) = \begin{cases} \left(1 + \exp(-\boldsymbol{\beta}^* \tilde{\mathbf{f}}_j)\right)^{-1}, & \text{if } \mathbf{f}_j \in C_1 \\ 1 - \left(1 + \exp(-\boldsymbol{\beta}^* \tilde{\mathbf{f}}_j)\right)^{-1} & \text{if } \mathbf{f}_j \in C_2 \end{cases}.$$

The parameters in vector $\boldsymbol{\beta}$ are found by solving

$$\arg\max_{\boldsymbol{\beta}} \mathrm{LL}(\boldsymbol{\beta})$$

This convex optimization problem can be solved using most optimization solvers. Once the $\boldsymbol{\beta}$ regression coefficients are *trained*, a threshold probability can be set to discriminate between the two classes. In the testing or application phase of the recognition process, the probability that a given feature belongs to C_1 is given by equation (A.13), where $z = \boldsymbol{\beta}^* \tilde{\mathbf{f}}$, and this probability can be compared to the threshold in order to classify the gesture [64] (See object (p) in the right panel of Fig. 2.10).

There is a multi-class logistic regression method for recognition problems with more than just two types of gestures [64]. However, such multi-class methods do not necessarily improve the performance of the gesture recognition process, and do not scale well with increasing numbers of gesture classes being considered. In this dissertation, pairwise and one-versus-the-rest classification strategies are implemented instead.

Appendix B

RAMAN LASER/AMPLIFIER MODEL

In the fields of photonics and modern electro-optics, specialized lasers and light amplifiers for specific applications are still in high demand. In general, there is a lack of high-power lasers in the 1100-1500 nm range, and there are many applications where such lasers would be useful, e.g. remote water sensing and telecommunications [122]. Another application is found in exciting sodium atoms at 589 nm, usually after a frequency-doubling technique as been applied, in the upper atmosphere in order to create an artificial guide-star that can then be used to calibrate ground-based telescopes. Atmospheric turbulence and effects from Earth's rotation can be accounted for with adaptive optics; however, one ideally would like to calibrate the telescope lens in the same region of the sky as where the target of observation resides. When there are no natural stars that can be used for calibration, then an artificial bright spot (star) can be manufactured by exciting sodium atoms. There have been many proposals for a lasers that can perform this function, each with their trade-offs [130, 131, 132, 120, 119]. In this appendix chapter, a novel laser/amplifier design, which has recently been patented [122] and is currently being built, originally intended for this guide-star application, will be discussed, along with some strategies for modeling the laser/amplifier output on a computer.

B.1 Model Overview

The fiber laser/amplifier system that has been proposed uses only passive, single-mode, polarization maintaining (PM), step-index silica fiber that is doped with Germanium dioxide (GeO_2). It is assumed that the fiber medium is isotropic and homo-

geneous, meaning that μ and ϵ are scalar and constant throughout the fiber. This is known to be a fairly accurate approximation for most silica fibers. Also, the light is assumed to be linearly polarized, and does not change polarization along the length of the fiber. The fiber laser/amplifier consists of two sections, both of which have similar designs and are fashioned to amplify the signal power for the desired output wavelength of 1178 nm.

The first section of the system uses a 10/125 fiber, meaning that it has a 10 μm core diameter and a 125 μm cladding diameter. This section is seeded with a small (0.05 Watt) power at 1178 nm and is co-pumped pump at 1069 nm. The 1069 nm pump will undergo SRS, which creates a frequency shifted Stokes line at 1121 nm. The light at 1121 nm will resonant in a cavity of length L, with high reflectors at both ends of the cavity. There is also a highly reflective FBG at the end of the cavity for any residual 1069 nm pump power that did not convert to 1121 nm. A second Stokes line at 1178 nm is created from the 1121 nm light through SRS. Since the 1178 nm seed is narrow, SBS effects are taken into account at this wavelength, which may steal away some of the power from the desired 1178 nm output. Finally, FWM effects are calculated for both the 1121 nm and 1178 nm light in order to determine if spectral broadening in the fiber will limit the laser/amplifier's capabilities.

The main idea for the first section of the fiber laser/amplifier system is to amplify the 1178 nm seed in order to create a more powerful seed for the second section of the system. In the second section of this system, a 20/400 fiber is used because it can handle greater amounts of power than the 10/125 fiber. However, the setup for the second section of the system is identical to the first section, with a 1178 nm seed, co-pumped at 1069 nm, and with a 1121 nm cavity. A maximal output of power at 1178 nm from this second section is the ultimate goal of the entire fiber laser/amplifier system. SBS mitigation techniques may be needed in oder to suppress the total SBS gain across the fiber.

Since the 1121 nm light will be trapped in a Fabry-Pérot cavity, the power levels at

this wavelength are expected to be high. This effectively extends the interaction length with the 1178 nm light through Raman gain. However, higher powers and longer interaction lengths also help stimulate FWM within the frequency bandwidth [111]. If the 1121 nm light manages to broaden wider in frequency than the FBG can effectively reflect, then fairly significant amounts of power can be lost and the performance of the laser/amplifier will be reduced, if not terminated by excessive amounts of power feeding back into the optical elements. FWM can also effect the final output of the 1178 nm light if high enough powers are achieved, though SBS will probably be problematic before the FWM in terms of 1178 nm power.

B.2 Theoretical Background

It is well known that light is governed by Maxwell's equations. However, when one simplifies the physics to the case of time-harmonic (monochromatic) light in a isotropic and homogeneous medium with very weak magnetic permeability, then one can readily derive Helmholtz's equation for the electric field:

$$\left(\nabla^2 + k^2\right)\mathcal{E} = \vec{0}, \tag{B.1}$$

where $k^2 = \omega^2\mu\epsilon$ and $\mathcal{E}$ is the electric field vector [112, 108, 109].

One of the first common assumptions that is made in modeling laser/amplifier systems is that the electric field is separable into its radial, azimuthal, and longitudinal components. This assumption has been shown to yield fairly accurate results for certain laser/amplifier systems; but it should be noted that this assumption is not physically nor mathematically correct when accounting for all the nonlinear interactions that occur in the fiber. Specifically, some mode coupling is neglected when the electric field is assumed to be separable, which leads to less loss being seen in the fiber than would physically occur [112, 108].

Nonetheless, using the separable assumption, one can derive the linearly polarized LP_{01}, transverse modal distribution of the amplitude for the j^{th} frequency component

in a single-mode, step-index fiber using Helmholtz's equation (B.1), which yields

$$F_j(r,\theta) = \begin{cases} C_0 J_0(\kappa_j r), & \text{for } r \in [0,a] \\ C_0 \frac{J_0(\kappa_j a)}{K_0(\zeta_j a)} K_0(\zeta_j r), & \text{for } r \in [a,b] \end{cases} \quad [112, 108]. \tag{B.2}$$

Here cylindrical coordinates (r, θ, z) are used, a is the radius of the core region of the fiber, and b is the radius of the cladding region of the fiber. The constant C_0 is the amplitude parameter, which can be ignored since it will be cancelled out in all relevant calculations. Note that $\kappa_j = \sqrt{k_0^2 n_{\text{core}_j}^2 - \beta_j^2}$, $\zeta_j = \sqrt{\beta_j^2 - k_0^2 n_{\text{clad}_j}^2}$, and β_j is the propagation constant for the j^{th} spectral component, which must satisfy $k_0 n_{\text{clad}_j} < \beta_j < k_0 n_{\text{core}_j}$ in order to have a guided mode instead of an attenuated mode. The propagation constant can be found by solving

$$\kappa_j a \frac{J_1(\kappa_j a)}{J_0(\kappa_j a)} = \zeta_j a \frac{K_1(\zeta_j a)}{K_0(\zeta_j a)} \tag{B.3}$$

for β_j [112, 108].

Marcuse discovered that the transverse modal distribution of the amplitude for the LP_{01} mode can be approximated fairly accurately by a Gaussian function:

$$F_j(r,\theta) = e^{-\frac{r^2}{w_j^2}} \tag{B.4}$$

for $r \in [0, +\infty)$, where the width parameter of the j^{th} frequency compnent w_j is given by the formula

$$w_j \approx a\left(0.65 + \frac{1.619}{V_j^{\frac{3}{2}}} + \frac{2.879}{V_j^6}\right),$$

which is based off of a best fit approximation of the Gaussian function and is accurate in the interval $1.2 < V_j < 2.4$. The normalized frequency V_j for the j^{th} spectral component is given by $V_j = a\sqrt{\kappa_j^2 + \zeta_j^2} = k_0 a \sqrt{n_{\text{core}_j}^2 - n_{\text{clad}_j}^2}$. Single-mode propagation occurs when $V_j < 2.405$ [114, 108].

Although Marcuse's Gaussian function can help simplify the mathematics of a numerical model, it is still necessary to find the propagation constants β_j for each

frequency component. Therefore, not too much work can be saved by using the Gaussian approximation as compared to the analytical Bessel function solution found in equation (B.2).

Related to the transverse modal distribution is the power-filling factor. This factor accounts for the amount of power that is lost to the unguided cladding region of a step-index fiber. The power-filling factor of the core region is calculated to be the percent volume that the transverse modal distribution occupies the core region of the fiber:

$$\Gamma = \frac{\int_0^{2\pi}\int_0^{a}|F|^2 r dr d\theta}{\int_0^{2\pi}\int_0^{\infty}|F|^2 r dr d\theta}. \tag{B.5}$$

It is more mathematically convenient to use Marcuse's Gaussian approximation to the transverse modal distribution to calculate the power-filling factor as compared to the Bessel function solution [108].

The governing equation for light propagation in the fiber is given by

$$\nabla \times \nabla \times \mathcal{E} = -\frac{1}{c^2}\frac{\partial^2 \mathcal{E}}{\partial t^2} - \mu_0 \frac{\partial^2 \mathbf{P}}{\partial t^2}.$$

However, in an isotropic, homogeneous medium $\nabla \times \nabla \times \mathcal{E} = -\nabla^2 \mathcal{E}$, and so

$$\vec{\nabla}^2 \mathcal{E} - \frac{1}{c^2}\frac{\partial^2 \mathcal{E}}{\partial t^2} = \mu_0 \frac{\partial^2 \mathbf{P}}{\partial t^2}, \tag{B.6}$$

where $\mathcal{E}$ is the electric field vector and $\mathbf{P}$ is the induced polarization vector. It is assumed that the induced polarization can be broken into its linear and nonlinear parts, which is justified by the fact that the nonlinear polarization is a small perturbation to the entire polarization, $\mathbf{P} = \mathbf{P}_{\mathrm{L}} + \mathbf{P}_{\mathrm{NL}}$, because the changes to the nonlinear index of refraction are many orders of magnitude smaller than the actual index of refraction. Also, the slowly-varying amplitude approximation will be invoked because most energy transferring effects occur over distances much larger than the wavelengths involved in the nonlinear processes. This approximation assumes $\left|\frac{\partial^2 \mathcal{E}}{\partial z^2}\right| \ll \left|k\frac{\partial \mathcal{E}}{\partial z}\right|$ [116, 108].

B.2.1 GeO_2 Doping and Chromatic Dispersion

As was stated earlier, the optical fibers that are employed in this laser/amplifier system will be doped with GeO_2. For the purposes of a computational model, it will be assumed that the GeO_2 doping is constrained solely to the fiber core and is uniform throughout the core region, even though it is not likely to be true. This is a reasonable assumption because the doping of every fiber is likely to be different and the inconsistencies in the doping concentrations throughout the fiber are nearly impossible to empirically determine accurately. Also, this assumption has the advantage of being mathematically simple to work with.

Having a GeO_2 doping in the fiber effects several parameters that control important aspects of the light propagation and nonlinear interactions that occur in the system. For instance, the nonlinear index of refraction depends on the GeO_2 doping concentration as follows

$$n^{\mathrm{NL}} \approx 2.16 \cdot 10^{-20} + 3.30 \cdot 10^{-22} \cdot \chi_{\mathrm{GeO_2}},$$

where $\chi_{\mathrm{GeO_2}}$ is the GeO_2 mole concentration in percent [89, 92].

Another set of fiber parameters that are directly affected by the GeO_2 doping are the Raman gain coefficients, which couple together the various Stokes frequencies and allow for energy transfer between them. These Raman gain coefficients depend both on the GeO_2 concentration and on the given frequency of light. The Raman gain coefficients are calculated as follows...

$$g_R(\chi_{\mathrm{GeO_2}}, \nu) = \frac{n_{\mathrm{clad}}^2}{n_{\mathrm{core}}^2}\left[g_R\left(\mathrm{SiO_2}, \nu\right) + C(\nu)\chi_{\mathrm{GeO_2}} g_R^p\left(\mathrm{SiO_2}, \nu\right)\frac{\lambda_s^3}{\lambda_{s\mathrm{Peak}}^3}\right], \tag{B.7}$$

where the pump (p) and Stokes (s) waves maintain the same state of polarization as they travel along the fiber. The GeO_2 mole concentration in percent ($\chi_{\mathrm{GeO_2}}$) is considered to be uniform throughout the fiber core.

The peak gain at a Stokes wavelength of 1.0μm in pure silica is $g_R\left(\mathrm{SiO_2}, 1.0\mu\mathrm{m}\right) =$

$1.046 \cdot 10^{-13}$ m / Watt. In equation (B.7)

$$g_R\left(SiO_2, \nu\right) = \frac{g_R\left(SiO_2, 1.0\mu m\right)\lambda_{1.0\mu m}}{\lambda_s}$$

and

$$g_R^p\left(SiO_2, \nu\right) = \frac{g_R\left(SiO_2, 1.0\mu m\right)\lambda_{1.0\mu m}}{\lambda_p}.$$

The linear regression factor in equation (B.7) is $C(\nu) \approx 0.079$ for a Raman shift of 440 cm^{-1} = 13.2 THz. The parameter $\lambda_{s\text{Peak}}$ is the wavelength of the Raman shift at the peak gain starting from the pump wavelength. Thus,

$$\lambda_{s\text{Peak}} = \frac{c}{\frac{c}{\lambda_p} - 13.2\ \text{THz}}.$$

The effects of the GeO_2 concentration are felt the most in the chromatic dispersion in the fiber. Usually the Sellmeier equation is used to determine frequency dependence of the index of refraction, assuming that reliable coefficients can be found for the given fiber. The Sellmeier equation that accounts for doping concentrations usually manifests itself the form of

$$n^2(\lambda, \chi_{GeO_2}) = 1 + \sum_{k=1}^{3} \frac{B_k(\chi_{GeO_2})\lambda^2}{\lambda^2 - C_k(\chi_{GeO_2})},$$

where λ is the vacuum wavelength measured in micrometers [103]. The coefficients $B_k(\chi_{GeO_2})$ and $C_k(\chi_{GeO_2})$ are found through fitting experimental data to the Sellmeier equation. It has been discovered that these coefficients depend linearly on the GeO_2 doping concentration [85]. For pure fused silica, a reasonable choice for the Sellmeier equation is

$$n^2(\lambda) = 1 + \frac{0.696166300\lambda^2}{\lambda^2 - 4.67914826 \cdot 10^{-3}} + \frac{0.407942600\lambda^2}{\lambda^2 - 1.35120631 \cdot 10^{-2}} + \frac{0.897479400\lambda^2}{\lambda^2 - 97.93440025},$$

according to the CVI Melles Griot technical report [90].

Using the data collected in Table 2.2 of Y. Kang's Master's Thesis [85], the linear best-fit regressions in Table B.1 can be calculated. These best-fit lines are drawn in

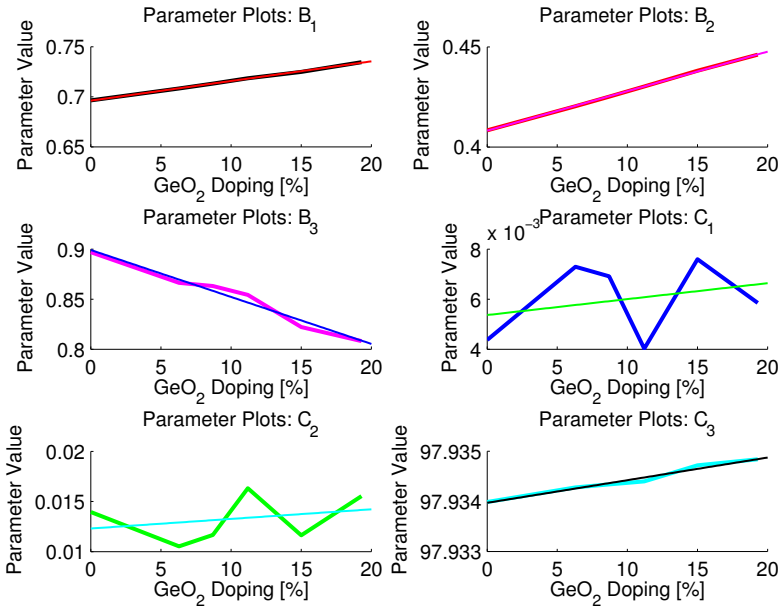


Figure B.1: These are the plots of the parameter values of the Sellmeier Equation that were measured and tabulated in Table 2.2 of Y. Kang's Master's Thesis [85]. The best-fit lines to these data points are also plotted. The formulas for these lines are delineated in Table B.1.

.

Fig. B.1. Thus, the chromatic dispersion of a GeO_2 doped fiber may be described by

$$n^2(\lambda, \chi_{GeO_2}) = 1 + \frac{(0.6963 + 0.19623 \cdot \chi_{GeO_2})\,\lambda^2}{\lambda^2 - (0.005365 + 0.006367 \cdot \chi_{GeO_2})} + \frac{(0.4081 + 0.19737 \cdot \chi_{GeO_2})\,\lambda^2}{\lambda^2 - (0.012303 + 0.009579 \cdot \chi_{GeO_2})} + \frac{(0.8991 - 0.4689 \cdot \chi_{GeO_2})\,\lambda^2}{\lambda^2 - (97.9340 + 0.0045 \cdot \chi_{GeO_2})}, \tag{B.8}$$

where the vacuum wavelength (λ) is measured in microns.

B.2.2 Stimulated Raman and Brillouin Scattering

As the intensity of light in the fiber increases, nonlinear effects can stimulate Raman scattering. This stimulated Raman scattering (SRS) is a by-product of third-order nonlinear susceptibility such that local vibrational effects in the medium down-shift the frequency of the photons that are emitted as the upper level electrons decay to a lower level that is not quite as energetically low as the lower level of the spontaneous emission from which lasing occurs [88, 86, 113, 108]. The frequency shift in the light caused by SRS creates a Stokes line usually around the peak of the Raman gain, which is about 440 cm^{-1} = 13.2 THz for optical fibers. Stokes lines, and even anti-Stokes lines, can grow from noise given enough pump power and interaction length; however, these lines grow much faster when seeded with power or when inside a cavity, properly tuned to a wavelength in the gain profile of the SRS. The SRS coupling between the pump and signal wavelengths is as equally strong for light propagating in the same or in the opposite direction of the incident light [108].

Table B.1: The best-fit linear regression lines that describe the Sellmeier equation parameters' dependencies on the GeO_2 doping concentration calculated from the data in Table 2.2 of Y. Kang's Master's Thesis [85]. These best-fit lines are drawn in Fig. B.1.

$B_1 = 0.6963 + 0.19623 \cdot \chi_{GeO_2}$
$B_2 = 0.4081 + 0.19737 \cdot \chi_{GeO_2}$
$B_3 = 0.8991 - 0.4689 \cdot \chi_{GeO_2}$
$C_1 = 0.005365 + 0.006367 \cdot \chi_{GeO_2}$
$C_2 = 0.012303 + 0.009579 \cdot \chi_{GeO_2}$
$C_3 = 97.9340 + 0.0045 \cdot \chi_{GeO_2}$

Another by-product of third-order nonlinear susceptibility is stimulated Brillouin scattering (SBS), in which the intense light of the laser induces an acoustical wave in the medium and then interacts with that acoustical wave, creating a light wave that is slightly down-shifted in frequency and propagating in the opposite direction to the incident light [86, 108]. SBS down-shifts the frequency $\mathcal{O}(10^{12})$ Hz. Similar to the SRS, SBS can grow from noise given enough pump power and interaction length. However, when the spectral width of the pump is much greater than the full width at half maximum (FWHM) of the SBS gain spectrum ($\Delta\nu_{\text{pump}} \gg \Delta\nu_{\text{SBS}}$), the peak value of the Brillouin gain is reduced by a factor $1 + \Delta\nu_{\text{pump}}/\Delta\nu_{\text{SBS}}$. For example, a SBS pump wavelength of 1178 nm has a FWHM of the SBS gain profile of $\Delta\nu_{\text{SBS}} \approx 30$ MHz. Thus, the SBS only has a significant effect when its pump has a narrow bandwidth [108].

The Raman gain coefficient, or coupling factor, is both wavelength and GeO_2 doping dependent, and is usually $\mathcal{O}(10^{-13})$ m / Watt. The formula for determining the Raman gain coefficient is found in equation (B.7). The Brillouin gain coefficient, or coupling factor, is almost two orders in magnitude greater than the Raman coupling factor, usually $\mathcal{O}(10^{-11})$ m / Watt. The Brillouin coupling factor may also be dependent on its pump wavelength and on the fiber GeO_2 doping concentration, but, for the purposes of this laser/amplifier system, is kept constant at $5 \cdot 10^{-11}$ m / Watt because only the wavelength of 1178 nm will experience any significant SBS gain [113, 108].

Both the Raman and Brillouin gain also dependent on the transverse modal overlap between the pump and signal frequencies. This effective area of interaction can occur in both the fiber core and cladding, and is calculated as follows...

$$\mathrm{A}^{\text{eff}}_{\text{pump-signal}} = 2\pi \frac{\int_0^\infty F^2_{\text{pump}}(r) r \; dr \cdot F^2_{\text{signal}}(r) r \; dr}{\int_0^\infty F^2_{\text{pump}}(r) F^2_{\text{signal}}(r) r \; dr}, \tag{B.9}$$

where $F_{\text{pump}}(r)$ and $F_{\text{signal}}(r)$ are the transverse modal distributions in equation (B.2) [85]. A large effective area will weaken the overall coupling strength between the pump and the resultant signal.

All of the frequencies of light present in the fiber undergo loss and scattering. Rayleigh back-scattering (RBS) and attenuation of these intensities are accounted for in the general linear loss parameter α, which is measured in m^{-1}. This linear loss is both frequency and GeO_2 doping dependent [108].

Therefore, the steady state governing equations in the frequency domain, which accounts for how power is transferred between the various wavelengths in the fiber as the light propagates in the fiber, are written out as follows...

$$
\begin{aligned}
\pm\frac{dP^{\pm}_{1069}}{dz} &= \Big[-\frac{\nu_{1069}}{\nu_{1121}\mathrm{A}^{\mathrm{eff}}_{1069-1121}}\Big(\gamma^{+}_{1121}P^{\pm}_{1121}+\gamma^{-}_{1121}P^{\mp}_{1121}\Big) - \\
&\qquad \alpha_{1069}\Big]P^{\pm}_{1069} \\
\pm\frac{dP^{\pm}_{1121}}{dz} &= \Bigg[\frac{\gamma^{+}_{1121}P^{\pm}_{1069}+\gamma^{-}_{1121}P^{\mp}_{1069}}{\mathrm{A}^{\mathrm{eff}}_{1069-1121}} - \\
&\qquad \frac{\nu_{1121}}{\nu_{1178}\mathrm{A}^{\mathrm{eff}}_{1121-1178}}\Big(\gamma^{+}_{1178}P^{\pm}_{1178}+\gamma^{-}_{1178}P^{\mp}_{1178}\Big) - \\
&\qquad \frac{\nu_{1121}}{\nu_{1178.06}\mathrm{A}^{\mathrm{eff}}_{1121-1178.06}}\Big(\gamma^{+}_{1178.06}P^{\pm}_{1178.06} + \\
&\qquad \gamma^{-}_{1178.06}P^{\mp}_{1178.06}\Big) - \alpha_{1121}\Bigg]P^{\pm}_{1121} \\
\pm\frac{dP^{\pm}_{1178}}{dz} &= \Bigg[\frac{\gamma^{+}_{1178}P^{\pm}_{1121}+\gamma^{-}_{1178}P^{\mp}_{1121}}{\mathrm{A}^{\mathrm{eff}}_{1121-1178}} - \\
&\qquad \frac{\nu_{1178}\gamma_{SBS}}{\nu_{1178.06}\mathrm{A}^{\mathrm{eff}}_{1178-1178.06}}P^{\mp}_{1178.06} - \alpha_{1178}\Bigg]P^{\pm}_{1178} \\
\pm\frac{dP^{\pm}_{1178.06}}{dz} &= \Bigg[\frac{\gamma^{+}_{1178.06}P^{\pm}_{1121}+\gamma^{-}_{1178.06}P^{\mp}_{1121}}{\mathrm{A}^{\mathrm{eff}}_{1121-1178.06}} + \frac{\gamma_{SBS}}{\mathrm{A}^{\mathrm{eff}}_{1178-1178.06}}P^{\mp}_{1178} - \\
&\qquad \alpha_{1178.06}\Bigg]P^{\pm}_{1178.06}\ .
\end{aligned}
\tag{B.10}
$$

B.2.3 Four-Wave Mixing

Understanding the spectral broadening of the power profile of the light requires a study of the parametric FWM of the light as it propagates down the fiber. FWM in-

cludes self-phase modulation, cross-phase modulation, and both non-degenerate and degenerate four-wave mixing within a bandwidth of frequencies around a central frequency ($\bar{\nu}$).

In order to completely model parametric FWM in a fiber, one would have to account for the possibility of having multiple modes and multiple polarizations of the light inside of the fiber. However, for simplicity, this model assumes that a single-mode, polarization maintaining (PM) fiber is used to construct the laser/amplifier system. Plus, it is assumed that the light that undergoes FWM inside of the fiber is quasi-monochromatic, which means that the spectral width $\Delta\nu$ is concentrated around a central frequency $\bar{\nu}$ such that $\frac{\Delta\nu}{\bar{\nu}} \ll 1$ [128, 104, 82, 93].

Assuming that the nonlinear response is instantaneous, the nonlinear induced polarization is given by

$$\mathbf{P}_{\mathrm{NL}} = \epsilon_0 \chi^{(3)} \mathcal{E}\mathcal{E}\mathcal{E},$$

where $\chi^{(3)}$ is the third-order susceptibility tensor [127, 87, 86, 126, 98]. The third-order susceptibility tensor is a fourth-rank tensor, which accounts for different polarizations and is frequency dependent. Converting to the spectral domain by means of a Fourier transform, making an ansatz for the electric field that separates variables such that

$$\mathcal{E} = F(r,\theta)\mathcal{A}(z,\nu)e^{i\bar{\beta}z}, \tag{B.11}$$

plugging into the governing equation (B.6), and ignoring the other nonlinear effects like SRS and SBS, one can derive the governing amplitude equation for FWM experienced by the j^{th} spectral component of the frequency band centered about $\bar{\nu}$, which yields...

$$\frac{\partial \mathcal{A}_j}{\partial z} = i\gamma^{\mathrm{NL}}_{j\ell mn} \cdot \sum_{\ell=1}^{M}\sum_{m=1}^{M}\sum_{n=1}^{M}\Big[(\mathcal{A}_\ell)^* \mathcal{A}_m \mathcal{A}_n \cdot \delta_{\nu_j+\nu_\ell,\nu_m+\nu_n} e^{-i\Delta\beta_{j\ell mn}z}\Big], \tag{B.12}$$

where $j = 1, 2, \ldots, M$, $(\cdot)^*$ represents the complex conjugate operator, and $\mathcal{A}_j$ represents the j^{th} spectral component of the amplitude propagating along the

fiber [108, 104, 82, 93]. The amplitude is normalized such that $\mathcal{A}_j = \sqrt{P_j}e^{i\phi_j}$, where P_j is the j^{th} spectral component of the power and ϕ_j is the j^{th} spectral component of the phase of the amplitude [126, 104, 82, 93].

The Kronecker delta function in equation (B.12) is defined by

$$\delta_{j,k} = \begin{cases} 0, & \text{if } k \neq j \\ 1, & \text{if } k = j \end{cases}.$$

This Kronecker delta function enforces energy conservation:

$$\nu_j + \nu_\ell = \nu_m + \nu_n. \tag{B.13}$$

Material dispersion accounts for the conservation of momentum:

$$\Delta\beta_{j\ell mn} = \beta_m + \beta_n - \beta_j - \beta_\ell \tag{B.14}$$

Recall that the j^{th} spectral component of the propagation constant (β_j) is found by solving the eigen-equation (B.3) [106, 104, 82, 93].

The FWM gain, sometimes called the nonlinear coefficient for FWM, of the j^{th} spectral component is given by

$$\gamma^{\text{NL}}_{j\ell mn} = \frac{n_j^{\text{NL}}\omega_j}{c\text{A}^{\text{eff}}_{j\ell mn}} \tag{B.15}$$

The nonlinear index of refraction n^{NL} (commonly denoted n_2) is defined so as to make the index of refraction $(\tilde{n})$ an intensity (and frequency) dependent parameter: $\tilde{n} = n + n^{\text{NL}}|E|^2$. The nonlinear index of refraction can then be shown to be $n^{\text{NL}} = \frac{3}{8n}Re(\chi^{(3)})$ [100, 126, 104, 82, 93].

The strength of all FWM processes should be directly dependent upon the third-order susceptibility tensor $\chi^{(3)}$; however, the values of the components of the third-order susceptibility tensor can be difficult to ascertain both theoretically and experimentally [127, 87, 86, 126, 98]. This is a limiting factor to the accuracy and validity of numerically modeling FWM [114, 125, 104, 82, 93]. FWM is also pronounced when

it is phased-matched, and this is more likely to occur near the zero dispersion wavelength of the fiber, where the material dispersion and chromatic dispersion cancel each other other [115, 96, 97].

The effective area for FWM in equation (B.15) came from an inconsistency while deriving equation (B.12) from the governing equation (B.6). This inconsistency is a direct consequence of assuming that the solution to the electric field could be separated as was done in the ansatz (B.11). However, in order to circumvent this inconsistency, the transverse mode profiles were integrated over and moved to the same side of the equation, producing the effective area for FWM. The j^{th} frequency experiences FWM in its own bandwidth in an effective area that is given by

$$\mathrm{A}_{j\ell mn}^{\text{eff}} = \frac{\sqrt{\prod_{q=j,\ell,m,n} \int_0^{2\pi} \int_0^{\infty} |F_q|^2 r dr d\theta}}{\int_0^{2\pi} \int_0^{\infty} F_j^* F_\ell^* F_m F_n r dr d\theta} \text{ [114, 104].} \tag{B.16}$$

It is also important to understand that the derivation of equation (B.12) relied heavily upon the assumption that the frequencies involved are centered around some central frequency [108]. Because of the nonlinear nature of the governing equations and the shear number of assumptions involved, it is not clear how separated the frequencies can be from one another before equation (B.12) is no longer valid. This issue probably is not a concern for the FWM that occurs within one bandwidth centered around one frequency, but may be relevant when considering FWM between stokes lines created from stimulated Raman scattering (SRS).

B.2.4 Fiber Bragg Gratings

Fiber Bragg gratings (FBGs) are usually created by inducing ultraviolet light across the fiber core, over some length, in order to produce a periodic structure of alternating regions of high and low refractive indices. FBGs can be tuned to have specific maximum reflectivities and reflective bandwidths, among other properties. The spectral dependence of FBG can be obtained through coupled-mode theory, which is discussed in greater detail in T. Erdogan's paper [95].

For this model it will be assumed that the FBGs are unchirped, that the incident light is only single-mode, and that the FBG manufacturer provides only the central wavelength of the FBG $(\bar{\lambda})$, along with its maximum power reflectivity $R_{\max}$ and a FBG bandwidth at a specified reflectivity (e.g. $\Delta\lambda_{\mathrm{FBG}}^{R=95\%}$). The power (R) and amplitude (ρ) reflectivities as a function of wavelength are given in equations (B.22) and (B.23) respectively, where $R = |\rho|^2$ [95]. However, in order to use these formulas other nontrivial parameters must be determined from what is already known about the FBG, including the length of the FBG (L), the perturbation to the effective index of refraction $\overline{\delta n}_{\mathrm{eff}}$, the number of periods of refractive index change (N), the period length (Λ), the AC coupling coefficient (κ), and the DC self-coupling coefficient $(\hat{\sigma})$.

Note that the maximal reflectivity occurs when $\lambda = \bar{\lambda}$. In this situation $\hat{\sigma}(\bar{\lambda}) = 0$; thus, $R_{\max} = \tanh^2\left[L\kappa(\bar{\lambda})\right]$. The grating strength, or AC part of the induced index change, is given by $v\overline{\delta n}_{\mathrm{eff}}$, where v is the fringe visibility (also called the "interferometric visibility"), which is ideally equal to 1 if the interfering optical waves are monochromatic and of equal intensity, and $\overline{\delta n}_{\mathrm{eff}}$ is a measure of the perturbation to the effective index of refraction. Since the FBG are assumed to be unchirped, then the grating is uniform along z, and so $\overline{\delta n}_{\mathrm{eff}}$ is a constant. Therefore,

$$v\overline{\delta n}_{\mathrm{eff}}(L) = \frac{\bar{\lambda}\tanh^{-1}\left(\sqrt{R_{\max}}\right)}{\pi L} \text{ [95].} \tag{B.17}$$

The DC self-coupling coefficient $(\hat{\sigma})$ is related to the DC coupling coefficient (σ) in a unchirped FBG as follows...

$$\hat{\sigma}(L,\lambda) = \delta(L,\lambda) + \sigma(L,\lambda),$$

where δ is the detuning of the FBG, which for all gratings is independent of z. The detuning is defined to be

$$\delta(L,\lambda) = 2\pi n_{\mathrm{eff}}\left(\frac{1}{\lambda} - \frac{1}{\lambda_{\mathrm{D}}(L)}\right),$$

where the effective index of refraction $n_{\mathrm{eff}} = \beta(\bar{\lambda})/k_0$ and λ_{D} is the design wavelength for Bragg scattering. The propagation constant at the center wavelength is $\beta(\bar{\lambda})$

and the wavenumber in a vacuum at the center wavelength is $k_0 = 2\pi/\bar{\lambda}$. The design wavelength for Bragg scattering is a resonant wavelength for reflection for the modes in the FBG. For single-mode FBGs, this design wavelength is given by $\lambda_D(L) = 2n_{\text{eff}}L/N$. The DC coupling coefficient for a single-mode FBG is given by

$$\sigma(L,\lambda) = \frac{2\pi\overline{\delta n}_{\text{eff}}(L)}{\lambda} \text{ [95].}$$

Using these definitions and the fact that $\hat{\sigma}(\bar{\lambda}) = 0$, the number of FBG periods can be determined by the following relation...

$$N(L) = \frac{2L}{\bar{\lambda}}\left[n_{\text{eff}} + \overline{\delta n}_{\text{eff}}(L)\right]. \tag{B.18}$$

The FBG nominal period is calculated by

$$\Lambda(L) = \frac{L}{N(L)}. \tag{B.19}$$

Therefore, the DC self-coupling coefficient is given by

$$\hat{\sigma}(L,\lambda) = 2\pi n_{\text{eff}}\left(\frac{1}{\lambda} - \frac{1}{2n_{\text{eff}}\Lambda(L)}\right) + \frac{2\pi\overline{\delta n}_{\text{eff}}(L)}{\lambda}. \tag{B.20}$$

Finally, for a single-mode FBG, the AC coupling coefficient is given by

$$\kappa(L,\lambda) = \frac{\pi v\overline{\delta n}_{\text{eff}}(L)}{\lambda} \text{ [95].} \tag{B.21}$$

The reflectivity of a single-mode, unchirped FBG is given by

$$R(\lambda) = \frac{\sinh^2\left(L\cdot\xi(L,\lambda)\right)}{\cosh^2\left(L\cdot\xi(L,\lambda)\right) - \left(\frac{\hat{\sigma}(L,\lambda)}{\kappa(L,\lambda)}\right)^2} \tag{B.22}$$

and

$$\rho(\lambda) = \frac{-\kappa(L,\lambda)\sinh\left(L\cdot\xi(L,\lambda)\right)}{\hat{\sigma}(L,\lambda)\sinh\left(L\cdot\xi(L,\lambda)\right) + i\xi(L,\lambda)\cosh\left(L\cdot\xi(L,\lambda)\right)}, \tag{B.23}$$

where R is the reflectivity of power, ρ is the reflectivity of the amplitude, L is the length of the FBG, κ is the AC coupling coefficient, $\hat{\sigma}$ is the DC self-coupling coefficient, $\xi(L,\lambda) = \sqrt{\kappa^2(L,\lambda) - \hat{\sigma}^2(L,\lambda)}$, and $R = |\rho|^2$.

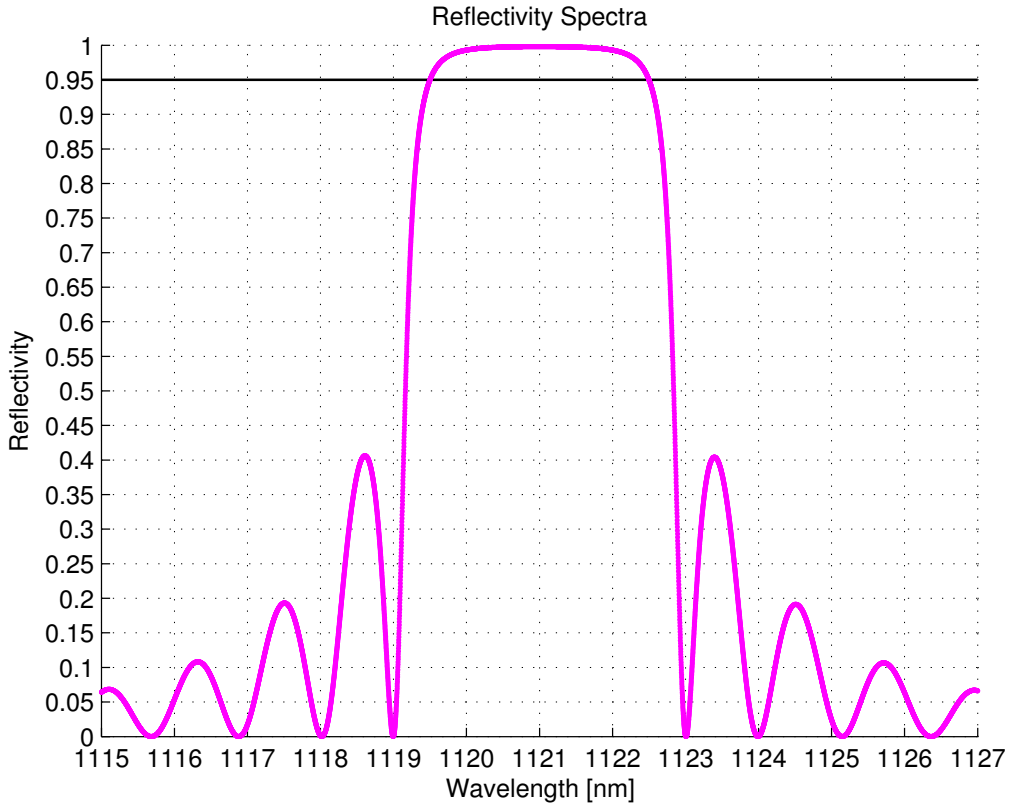


Figure B.2: This is a plot of equation (B.22), with $\lambda_0 = 1121$ nm, $v = 1$, $\Delta\lambda_{\text{FBG}}^{R=95\%} = 3$ nm, and $R_{\max} = 0.998$. Using equation (B.24), the length of the FBG was determined to be 338.723 μm.

At this point, the reflectivity spectra of the FBG in equation (B.22) could be calculated if the fringe visibility v and the FBG length L were known. Unfortunately, the fringe visibility is not likely to be given by the FBG manufacturer, and any calculation of this parameter would require addition assumptions and would therefore likely be inaccurate at best. Fortunately, the final reflectivity spectra is not sensitive to the value of the fringe visibility. Therefore, assuming that the fringe visibility is equal to its ideal value of 1 is an acceptable guess that will not introduce much error

into the final spectra result.

The length of the FBG can be determined by ensuring that the bandwidth of the reflectivity spectra at its specified reflectivity (e.g. $R = 95\%$) matches what the FBG manufacturer claims in the given parameter $\Delta\lambda_{\text{FBG}}^{R=95\%}$. Note that under the assumption of having an unchirped grating, the reflectivity will be symmetric about the center wavelength $\bar{\lambda}$. Thus, $\lambda_{R=95\%} = \bar{\lambda} - \Delta\lambda_{\text{FBG}}^{R=95\%}/2$ is one of the two wavelengths that corresponds to the $R = 95\%$ reflectivity. The FBG length can then be found by solving equation (B.24) for L. The length of the FBG can be determined by solving the following equation for L:

$$0.95 = \frac{\sinh^2\left(L\sqrt{\kappa^2(L,\lambda_{R=95\%}) - \hat{\sigma}^2(L,\lambda_{R=95\%})}\right)}{\cosh^2\left(L\sqrt{\kappa^2(L,\lambda_{R=95\%}) - \hat{\sigma}^2(L,\lambda_{R=95\%})}\right) - \left(\frac{\hat{\sigma}(L,\lambda_{R=95\%})}{\kappa(L,\lambda_{R=95\%})}\right)^2}. \tag{B.24}$$

B.3 Algorithm Implementation & Development

It has been discovered that great care is needed in the way that the laser/amplifier system is translated into a numerical model with only a finite set of discrete points so as to preserve the realistic processes that are occurring in the fiber. The numerical model calculates the amplitude and power of the light inside the core as it propagates down the fiber.

One of the most persistent problems that occurs when implementing a numerical model is the fact that numerical models must be discrete and finite, yet must model continuous physical processes. In order to implement a numerical model of the system one must be careful to ensure that the numerical routine accurately reflects the physics that are occurring in the fiber.

B.3.1 Numerical Aperture

One important issue that immediately arises when starting to implement a computational model of the laser/fiber system concerns the validity of the parameters that

are given by the fiber manufacturer. Specifically, some manufacturers may provide a numerical aperture (NA) for the fiber core, while others may provide the indices of refraction of the fiber core and cladding at some specified frequency. Either way, it is likely that these given values will come with error measurements because they are based on the manufacturing averages, and not on the specific fiber that was purchased. These three fiber parameters are related by the equation

$$\text{NA} = \sqrt{n_{\text{core}}^2 - n_{\text{cladding}}^2} \text{ [108].} \tag{B.25}$$

Note that it has been assumed that the numerical aperture is constant for all frequencies of light the fiber may experience.

Obviously, the best situation would be if either the manufacturer provided a Sellmeier equation for the fiber, or if one could measure the refractive indices of the given fiber and calculate the parameters of the Sellmeier equation based on the collected empirical data. However, it is likely that one will have to use published Sellmeier equation parameters that do not exactly match the fiber that was purchased for the laser/amplifier system. In this case, there can be important consequences to which manufacturer provided values are trusted over other values because the effective areas for the SRS and SBS, and consequently the entire laser/amplifier output, depend greatly on the fiber core numerical aperture that gets used in the model.

If just the numerical aperture is given by the fiber manufacturer, then one is left to find a trustworthy Sellmeier equation to find the refractive index in the fiber core, and then calculate the refractive index in the fiber cladding through equation (B.25). However, if the manufacturer also provides the index of refraction of the fiber core at a specified wavelength, then one may need to adjust the Sellmeier equation to match the given data. This can be done by adding an artificial offset (translation) to the Sellmeier equation. Or, if there is GeO_2 doping in the fiber, then one can use a nonlinear solver to calculate the GeO_2 doping concentration (χ_{GeO_2}) that adjusts the Sellmeier equation so that it matches the given data. In the later scenario, it is important to

remember both the true GeO_2 doping concentration and the calculated concentration because, for example, the Raman gain coefficients need to be determined using the true concentration, while the calculation of the propagation constants in the fiber rely on the indices of refraction, which are determined by the Sellmeier equation and therefore use the calculated GeO_2 doping concentration.

Another possibility is when manufacturer provides only the indices of refraction for the fiber core and cladding, but not the fiber core numerical aperture. In this case, one can adjust a published Sellmeier equation, as has just been described, and can calculate the numerical aperture according to equation (B.25).

Unfortunately, it may occur that the manufacturer may provide all three fiber parameters, and yet they are not consistent with equation (B.25). In this case, one is left to use his/her best judgement, or to perform experiments, in order to determine which pieces of data are the most trustworthy.

B.3.2 Storage

Often it is the case that computational models can become very large in terms of the number of discretized points that are being used for calculations. Having more discretized points usually corresponds to having a more accurate and robust numerical model, but comes at the price of occupying large chunks of memory. In the end, one usually only cares to write out or see plots of the relevant results, and does not need to view all of the data at every discretized point used in the calculations.

In order to address this issue, the laser/amplifier model has a couple of parameters to reduce the amount of data that gets stored into the output files at the end of the program. Specifically, the two parameters are integers, *z_nth_portion* and *nu_nth_portion*, that control the proportion of the data used in the calculations that gets stored in the output files. The larger in magnitude the integer, the less data that gets stored; for example, an integer of 10, should mean that about 1/10 of the discretized points used for the calculations will be stored in the output files.

Along the fiber length, or equivalently in the z-direction, it would be prudent to ensure that the first and last discretized points (the ends of the fiber) get stored in the output files. In order to accomplish this, one has to make sure that the *z_nth_portion* is a integer factor of the number of discretized z-values. And, in the spectral domain, or equivalently in the *nu*-direction, it would be prudent to ensure that the central frequency gets stored, but not necessarily the endpoints.

B.3.3 Spectral Domain

The spectral domain contains a set of evenly spaced frequencies around a central frequency. The fact that the spectral domain is evenly spaced has some non-intuitive consequences on the amount of spectral broadening that is calculated under certain circumstances.

One important consideration when implementing a numerical model can be seen in equation (B.12), where M is the number of discrete frequency points in the model. An extra condition is needed to make the numerical model match reality; namely, that as $M \to \infty$, $\left| \sum_{j=1}^{M}, \mathcal{A}_{\bar{\lambda}_j} \right| < \infty$; this means that even if there are an infinite number of frequencies present in the system, then the total amount of amplitude needs to remain finite. This can be resolved by ensuring that the discrete amplitudes constitute a convergent series when summed over all M.

It should be noted that in the same equation (B.12), when the discretized frequency points are evenly spaced in the spectral domain, $\delta_{\nu_j+\nu_\ell,\nu_m+\nu_n}$ is equivalently represented by $\delta_{j+\ell,m+n}$.

Even though the nonlinear index of refraction $\left(n_j^{\mathrm{NL}} = n^{\mathrm{NL}}(\nu_j)\right)$ is frequency dependent, in this numerical model it will be treated as a constant because of the small frequency range being considered, and because of the lack of accurate data for this parameter.

If the magnitude of the initial ("initial" referring to the values at the beginning of the fiber: $z = 0$), amplitude $\mathcal{A}_0$ centered at a certain frequency $(\bar{\nu})$ is spread out

over a Gaussian profile in the spectral domain with a standard deviation of σ_ν, then the total magnitude of the initial amplitude is given by

$$|\mathcal{A}_0| = \int_{-\infty}^{+\infty} |\mathcal{A}(\nu)| \, d\nu.$$

However, when this Gaussian profile is represented by a discrete set of points, then the total magnitude of the initial amplitude is given by

$$|\mathcal{A}_0| = \sum_{j=1}^{M} |\mathcal{A}(\nu_j)|,$$

where M is the number of discrete points. Given that this a Gaussian profile, the amount of magnitude of the amplitude represented by the j^{th} spectral component can be expressed as

$$|\mathcal{A}(\nu_j)| = \text{NC} \exp\left(-\frac{(\nu_j - \bar{\nu})^2}{2\sigma_\nu^2}\right),$$

where NC is the normalization constant. Assuming that one wants to have the discrete points evenly spaced such that the spectral spacing $\Delta\nu = \frac{\sigma_\nu}{p}$, σ_ν is one standard deviation of the Gaussian profile (a measure of spectral bandwidth), and p is a positive integer representing the number of points per standard deviation, then the normalization constant can be found to be

$$\text{NC} = \frac{|\mathcal{A}_0|}{2\sum_{k=1}^{\frac{M-1}{2}} \exp\left(-\frac{(M-2k+1)^2}{8p^2}\right) + 1}.$$

It has been assumed that M is an odd number. This means that when $\bar{\nu} = 0$, the magnitude of the amplitude in the j^{th} spectral component should be given by

$$|\mathcal{A}(\nu_j)| = \frac{|\mathcal{A}_0| \exp\left(-\frac{(M-2j+1)^2}{8p^2}\right)}{2\sum_{k=1}^{\frac{M-1}{2}} \left[\exp\left(-\frac{(M-2k+1)^2}{8p^2}\right)\right] + 1}, \tag{B.26}$$

where $j = 1, 2, \ldots, M$. Since the $P(\nu_j) = |\mathcal{A}(\nu_j)|^2$, the corresponding power at each frequency point is given by

$$P(\nu_j) = \frac{P_0 \exp\left(-\frac{(M-2j+1)^2}{4p^2}\right)}{\left(2\sum_{k=1}^{\frac{M-1}{2}} \left[\exp\left(-\frac{(M-2k+1)^2}{8p^2}\right)\right] + 1\right)^2}, \tag{B.27}$$

where $j = 1, 2, \ldots, M$.

It is important to note that $\sum_{j=1}^{M} |\mathcal{A}(\nu_j)| = |\mathcal{A}_0|$ by definition. But, according this definition, $\sum_{j=1}^{M} P(\nu_j) \neq P_0$. In fact, $\sum_{j=1}^{M} P(\nu_j) = \frac{P_0}{\mathrm{PC}}$, where the power constant (PC) is defined by

$$\mathrm{PC} = \frac{\left(2\sum_{k=1}^{\frac{M-1}{2}} \exp\left(-\frac{k^2}{2p^2}\right) + 1\right)^2}{2\sum_{k=1}^{\frac{M-1}{2}} \exp\left(-\frac{k^2}{p^2}\right) + 1}.$$

Also, note that by implementing these discretization methods and definitions, neither the amplitude nor the power is dependent on the initial spectral bandwidth of the profile of the amplitude. Under the discretization methods that have been defined, the sbf along the fiber can be calculated from

$$\mathrm{sbf}(z) = \sqrt{\frac{\sum_{j=1}^{M}\left[P(\nu_j, z)\left(j - \frac{M+1}{2}\right)^2\right]\sum_{j=1}^{M} P(\nu_j, z=0)}{\sum_{j=1}^{M}\left[P(\nu_j, z=0)\left(j - \frac{M+1}{2}\right)^2\right]\sum_{j=1}^{M} P(\nu_j, z)}} \text{ [104, 82, 93].}$$

The spectral domain is divided up into M evenly spaced discrete frequencies such that each point is spaced $\Delta\nu = \frac{\sigma_\nu}{p}$ apart. Specifically, the discrete frequencies points are centered about a certain central frequency $\bar{\nu}$ such that

$$\nu_j = \bar{\nu} + \frac{\sigma_\nu}{p}\left(j - \frac{M+1}{2}\right).$$

The spectral width (sw) of the power profile can be defined as

$$\mathrm{sw} = 2\sqrt{\frac{\sum_{j=1}^{M}\left[P(\nu_j)(\nu_j - \bar{\nu})^2\right]}{\sum_{j=1}^{M} P(\nu_j)}}. \tag{B.28}$$

However, by applying the discretization method previously described, the spectral width can be simplified to

$$\mathrm{sw} = \frac{2\sigma_\nu}{p}\sqrt{\frac{\sum_{j=1}^{M} P(\nu_j)\left(j - \frac{M+1}{2}\right)^2}{\sum_{j=1}^{M} P(\nu_j)}}.$$

By comparing the spectral width at the beginning of the fiber ($z = 0$) to the spectral width farther down the fiber, one can define the spectral broadening factor (sbf) as

$$\mathrm{sbf}(z) = \frac{\mathrm{sw}(z)}{\mathrm{sw}(z=0)}. \tag{B.29}$$

It is important to recognize here that the spectral broadening factor is independent of the spectral bandwidth of the initial amplitude profile, whether that bandwidth be measured by the full width at half maximum (FWHM) or the standard deviation (σ_ν) of the profile. This means that one ought not to expect to see the spectral broadening factor to change when the FWHM of the initial amplitude profile is varied. Assuming a Gaussian distribution, one standard deviation is related to the full width at half maximum (FWHM) of the distribution profile by

$$\sigma = \frac{\text{FWHM}}{2\sqrt{2\ln(2)}}. \tag{B.30}$$

For each discrete spectral point in the frequency domain a random phase is given to the amplitude at the beginning of the fiber. The phase for the j^{th} amplitude component (ϕ_j) is chosen from a uniform distribution between $[0, 2\pi)$. Since these phases are chosen randomly, the dynamics of the system of coupled ODEs may evolve slightly differently each time the model is run. In order to better capture the true physics of the laser/amplifier system, it is prudent to run the model multiple times, randomly choosing the initial amplitude phases (ϕ_j) for each run, and then average the spectral broadening results of all the runs [104, 82].

B.3.4 Memory

Another issue with the computational code has to do with storing values verses calculating values on the fly. This issue arises when there is an enormous number of quantities that need to be calculated just once, but are used multiple times for other calculations. Clearly, it would be more efficient to just calculate the quantities once and store their values for future reference. However, if the number of quantities is too large to keep in memory, then one may have to recalculate the quantities as needed throughout the program.

This exact scenario occurs in this laser/amplifier model. One of the biggest culprits for over-utilizing memory is the effective area calculation for the FWM. The

program is using nearly all the available memory in the computer trying to store the effective area for each permutation of the M discretized frequencies mixing with 3 other frequencies, each with M possibilities, a total of M^4 permutations.

Because the integral in the denominator of the effective area equation (B.16) cannot be analytically solved, a numerical calculation is necessary in order to find the effective area. Since the numerical calculation can be time consuming, it is more efficient to store all the necessary effective areas needed for further calculations. Also, note that the conjugate operations in the integral in the denominator of the effective area formula are not necessary because the transverse modal distribution $F_j(r)$ is real for all frequencies indexed by $j = 1 \ldots M$.

The effective area changes based upon the four spectral points being accessed in the calculation, indexed by $j, \ell, m,$ and n. There are M spectral points in total. This means that in order to store all the effective areas one would need 4 dimensional tensor with M^4 points. Clearly, storing all the effective areas in this manner becomes intractable as M grows larger.

Fortunately there is one major advantage afforded by the effective area formula that is found in the symmetry among the four frequencies being accessed. This symmetry makes it possible to store only one of each permutation of the indices $j, \ell, m,$ and n because rearranging the order of the indices does not affect the resulting value of the effective area.

This problem of determining the number of unique combinations can be generally stated as having M distinct numbers that can be placed into D available positions. For the effective area problem, the dimension is $D = 4$ since there are four indices to permute. However, if one starts with $D = 1$, then there are clearly only M distinct combinations of 1 number given M distinct numbers.

When $D = 2$, one can fix the first number in the sequence and then note that the second number can be chosen as before in the $D = 1$ case, hence M distinct combinations. Then, if $M > 1$, fixing another number to be the first number in

the sequence, the second number in the sequence can be found by again considering $D = 1$, accept only having $M - 1$ available numbers, since the first number has already been exhausted. If $M > 2$, then this process can be repeated, until all the numbers have been exhausted. This method can produce a formula for the number of stored memory points (NSMP) as a function of the number of distinct numbers M and available positions for those numbers D:

$$\mathrm{NSMP}(M, D) = \begin{cases} \sum_{s=1}^{M} 1 = M, & D = 1 \\ \sum_{s=1}^{M} s = \frac{M(M+1)}{2}, & D = 2 \end{cases}. \tag{B.31}$$

$$\mathrm{NSMP}(M, 3) = \sum_{t=0}^{M-1} \sum_{s=1}^{M-t} s = \frac{M(M+1)(M+2)}{6} \tag{B.32}$$

Table B.2 lists the number of stored memory points for having up to five distinct spectral points (M) and up to five dimensions (D), and Table B.3 expresses the formulas used to compute those numbers. The general formula for calculating the NSMP is

$$\mathrm{NSMP}(M, D) = \prod_{s=1}^{D} \left[\frac{M + s - 1}{s} \right] = \binom{M + D - 1}{D} = \frac{(M + D - 1)!}{D!(M - 1)!}. \tag{B.33}$$

Table B.2: The number of stored memory points (NSMP) needed in order to store every distinct combination of M numbers in D available positions.

NSMP	$M = 1$	$M = 2$	$M = 3$	$M = 4$	$M = 5$
$D = 1$	1	2	3	4	5
$D = 2$	1	3	6	10	15
$D = 3$	1	4	10	20	35
$D = 4$	1	5	15	35	70
$D = 5$	1	6	21	56	126

The corresponding fractional reduction in memory storage is given by

$$\mathrm{FRMS}(M, D) = 1 - \frac{(M + D - 1)!}{M^D D!(M - 1)!}. \tag{B.34}$$

The maximum fractional reduction in storage points needed given the number of available positions D is found by taking the limit as the number of distinct points $M \to \infty$:

$$\mathrm{FRMS}_{\max}(D) = \lim_{M\to\infty} \mathrm{FRMS}(M, D) = \frac{D! - 1}{D!}. \tag{B.35}$$

For $D = 3$, a similar recurrence trick can be used as was done for the $D = 2$ case. First, fix the first number in the sequence, and the next two numbers in the sequence can be chosen just as was done for the $D = 2$ with M numbers case, yielding $M(M + 1)/2$ distinct combinations. Then fix another number to be the first number in the sequence, and chose the next two numbers in the sequence as would be done for a $D = 2$ with $M - 1$ numbers case. Continue until all M numbers are exhausted. This can be summarized by the equation (B.32). Table B.2 delineates some of the number of stored memory points for various numbers of spectral points (M) and dimensions (D), and Table B.3 expresses the formulas used to compute those numbers.

Table B.3: The formulas for calculating the number of stored memory points (NSMP) needed in order to store every distinct combination of M numbers in D available positions.

NSMP	Summation Formula	Formula
$D = 1$	$\sum_{s=1}^{M} 1$	M
$D = 2$	$\sum_{s=1}^{M} s$	$\frac{M(M+1)}{2}$
$D = 3$	$\sum_{t=0}^{M-1} \sum_{s=1}^{M-t} s$	$\frac{M(M+1)(M+2)}{6}$
$D = 4$	$\sum_{u=0}^{M-1} \sum_{t=0}^{M-1-u} \sum_{s=1}^{M-t-u} s$	$\frac{M(M+1)(M+2)(M+3)}{24}$
$D = 5$	$\sum_{v=0}^{M-1} \sum_{u=0}^{M-1-v} \sum_{t=0}^{M-1-u-v} \sum_{s=1}^{M-t-u-v} s$	$\frac{M(M+1)(M+2)(M+3)(M+4)}{120}$

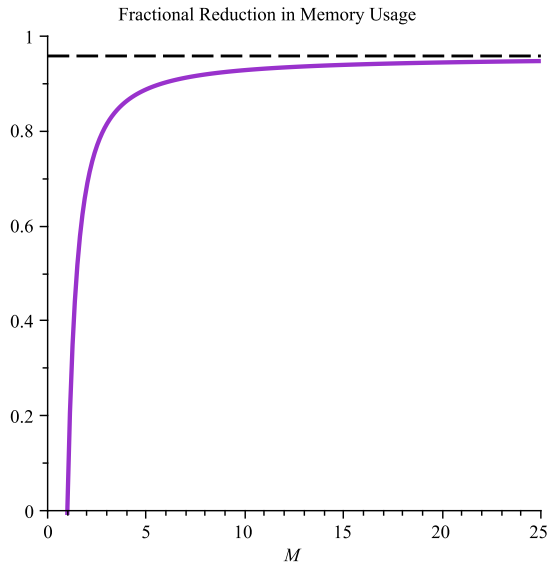


Figure B.3: This is a plot of equation (B.34), the fractional reduction in memory storage, as a function of the number of distinct points (M) and with the dimension fixed at $D = 4$. The theoretical maximum amount of fractional reduction for $D = 4$ is also plotted, which is $\mathrm{FRMS}_{\max}(4) = 23/24 \approx 0.9583$.

From Table B.3 it is fairly straight-forward to deduce that the general formula for the NSMP needed in order to store every distinct combination of M numbers in D available positions is given by equation (B.33). Originally, M^D memory points were needed to store the all the permutations of M numbers in D positions. Therefore, the fractional reduction in memory storage (FRMS) needed is given by equation (B.34). Finally, taking the limit as the number of distinct points $M \to \infty$ yields the maximum fractional reduction in storage points needed given the number of available positions D, which is expressed in equation (B.35).

Figure B.3 illustrates how quickly the fractional reduction in memory storage nears its maximal value as the number of distinct points M is increased, and with the number of positions fixed at $D = 4$.

B.3.5 Nonlinear Solver

Solving nonlinear equations for their roots can be fairly common in numerical models. In this laser/amplifier system there are three nonlinear equations that get solved: 1) the eigen-equation (B.3), which is solved in order to find the propagation constant β, 2) the FBG equation (B.24), which is solved in order to find the FBG length L, and 3) the Sellmeier equation (B.8), which is solved in order to find the GeO_2 doping concentration χ_{GeO_2} given a known refractive index of the fiber core at a specified wavelength.

This laser/amplifier model implements the Bisection method in order to solve these nonlinear equations. In order to solve these equations using the Bisection method one must first correctly guess an interval in which the root exists. The user must input the endpoints of this interval, and if the endpoints do not have functional values with opposite signs then the bisection method will stop, terminating the program, and output a warning indicating the problem.

The user also has control of other parameters that control the bisection method. The *fatol* parameter is an absolute error tolerance for the function value of the root. If an approximate root x satisfies $|f(x)| \leq$ fatol, then x will be accepted as the root and the iterations of the Bisection method will be terminated. The *xatol* parameter is an absolute error tolerance for the root, which is used to stop the solver if the root x is not changing very much from one iteration to the next. The *max_step* parameter is an upper limit on the number of iterations the Bisection method is allowed to take.

There are various reasons for which the Bisection method may terminate, and there is an output flag, called *iflag*, which provides some information as to why the solver terminated its calculation. Here are the possible *iflag* values:

- 0 - means $|f(x)| \leq$ *fatol*, which is good convergence,
- 1 - means $|x_{\text{left endpoint}} - x_{\text{right endpoint}}| <$ *xatol*, which is partial convergence, and

one ought to check the $f(x)$ output value,

- 2 - means the number of iterations is greater than the *max_step* parameter allows, which likely is not convergence, and

- 3 - means that the sign of the function values at the endpoints match when they ought to be opposite

For only the eigen-equation (B.3), which is solved in order to find the propagation constant β, the left and right endpoints of the interval around the root are chosen in a special way. Since it is known that $k_0 n_{\text{clad}} < \beta < k_0 n_{\text{core}}$, the interval $[k_0 n_{\text{clad}}, k_0 n_{\text{core}}]$ is divided into sixtyfour equal parts, and the user choses the starting and ending points of the interval using the parameters *sixtyfourths1* and *sixtyfourths2*, which determine how far, in terms of fractions of sixtyfour, into the known bounds on β the interval used in the Bisection method will start and end respectively. For example, if *sixtyfourths1* = 0 and *sixtyfourths2* = 64, then the interval used in the Bisection method would be $[k_0 n_{\text{clad}}, k_0 n_{\text{core}}]$.

B.3.6 Numerical Differentiation

Since this laser/amplifier system has a system of ODEs pertaining to the power propagation in the fiber centered around the wavelengths 1069 nm, 1121 nm, 1178 nm, and 1178.06 nm (SBS), and another system of ODEs that describes the amplitude propagation in the fiber for the bandwidths around 1121 nm and 1178 nm, a numerical scheme is needed in order to accurately and efficiently estimate the derivatives. Because of the nonlinearities in this model, one has to be careful to select a robust numerical scheme that maintains fidelity under the nonlinearities. Yet, it is also desirable to have a numerical scheme that is computationally fast and able solve the system in a reasonable amount of time.

To this end, the fourth-order, explicit Runge-Kutta (FORK) method was chosen for the necessary numerical differentiation. The FORK method works as is outlined in Table B.4.

Physically, the laser/amplifier system ought to be governed by a single set of coupled amplitude equations composed of the governing equations (B.10) and (B.12). However, for computational efficiency and because the governing FWM equations (B.12) do not actually change the total amount of power in the fiber, this model lends itself well to a split step method. This split step method works by propagating the total power in each bandwidth about 1069 nm, 1121 nm, 1178 nm, and 1178.06 nm (SBS) in the fiber for one step (Δz) in the fiber as if there were only one wavelength in each bandwidth with all of the power of the bandwidth accumulated into that bandwidth. At the same time, the FWM propagates at the half steps $\Delta z/2$, determining how the power is spreading as it propagates in the fiber. The FWM references the power at each step in the fiber to update how the power may have

Table B.4: The fourth-order, explicit Runge-Kutta (FORK) finite difference scheme that is used in this model to solve the steady-state ODEs, equations (B.10) and equations (B.12). The ODEs are assumed to be written in the following form: $\frac{d\mathbf{V}}{dz} = \mathrm{RHS}(z, \mathbf{V})$, where $\mathbf{V}$ can be a vector of powers or amplitudes at several different frequencies.

	Forward Propagating	Backward Propagating
Fourth	$S_1 = \mathrm{RHS}(z_n, \mathbf{V}_n^+)$	$S_1 = \mathrm{RHS}(z_n, \mathbf{V}_n^-)$
Order	$S_2 = \mathrm{RHS}\left(z_n + \frac{\Delta z}{2}, \mathbf{V}_n^+ + \frac{\Delta z}{2} S_1\right)$	$S_2 = \mathrm{RHS}\left(z_n - \frac{\Delta z}{2}, \mathbf{V}_n^- - \frac{\Delta z}{2} S_1\right)$
Runge-	$S_3 = \mathrm{RHS}\left(z_n + \frac{\Delta z}{2}, \mathbf{V}_n^+ + \frac{\Delta z}{2} S_2\right)$	$S_3 = \mathrm{RHS}\left(z_n - \frac{\Delta z}{2}, \mathbf{V}_n^- - \frac{\Delta z}{2} S_2\right)$
Kutta	$S_4 = \mathrm{RHS}\left(z_n + \Delta z, \mathbf{V}_n^+ + \Delta z S_3\right)$	$S_4 = \mathrm{RHS}\left(z_n - \Delta z, \mathbf{V}_n^- - \Delta z S_3\right)$
Method	$\mathbf{V}_{n+1}^+ = \mathbf{V}_n^+ + \frac{\Delta z}{6}\left(S_1 + 2S_2 + 2S_3 + S_4\right)$	$\mathbf{V}_{n-1}^- = \mathbf{V}_n^- - \frac{\Delta z}{6}\left(S_1 + 2S_2 + 2S_3 + S_4\right)$

transferred between the bandwidths. This split step method will remain sufficiently accurate as long as the step sizes in the z-direction Δz are not too large, though there is no specific measure of what "not too large" means. This split step method is somewhat similar to what D. Soh did in his simulation method [104].

B.3.7 Convergence

This fiber laser/amplifier model solves the governing equations with the implicit boundary conditions that come from the FBGs through a shooting method. This means that the computational model will solve the governing equations by propagating the light down the fiber (from $z = 0$ to $z = L$) and then back up the fiber (from $z = L$ to $z = 0$) in multiple rounds. If the system is inherently stable, based upon the fiber parameters that are given, then the system should start to converge to a relatively stable solution.

In order to determine when a relatively stable solution has been reached within the system, the convergence criterium must be met. After many rounds of propagating the light through the fiber, it is expected that the absolute difference in the power levels between one round and its previous round ought to be essentially null. With this in mind, the maximum absolute difference among the forward and backward propagating light from the current round and its previous round at 1069 nm, 1121 nm, 1178 nm, and 1178.06 nm (SBS) are stored in two values labelled as the actual forward and backward tolerances respectively. As long as both of these tolerances remain below a specified *converge_tol* value, set by the user, for a minimum number of rounds, specified by the *num_converge* parameter, then the program will converge.

The program may also terminate if the power at the SBS wavelength of 1178.06 nm exceeds a maximum tolerance 100 Watts. Or, the model will end if the program surpasses the maximum number of rounds, which is set in the *max_round* parameter. If the SBS is under control, the program will not even start looking to see if the convergence criterium is met until at least *num_converge* rounds have been calculated

in order to ensure that the program is truly approaching some sort of steady state.

Also, the program will print to the screen warnings about the power spreading outside of the spectral domain in the first and final rounds if the power at the ends of the spectral domain exceed a hundredth of the maximum amount of power that the initial 1121 nm and 1178 nm spectral profiles have at their peaks, which are at the center wavelengths of the bandwidths. These warnings indicate that more spectral points are needed in order to capture the spectral broadening. If these warnings appear in the final round, then the model output results should not be trusted because the power that should have continued to spread has probably reflected off of the spectral domain boundary and reverberated back into the spectral domain, changing the dynamics of the model in way that is not physically realistic.

The system may never reach a truly stable solution because of the FWM of the light. The FWM depends on phase coherence, and since the initial phases of the amplitudes in the 1121 nm and 1178 nm bandwidths are randomly chosen, and because the FBGs will change the phases of the 1121 nm light in every round, the spectral broadening will always be slightly different each round [104, 82]. These slight differences will also mildly effect the amount of power that escapes around the FBGs, and therefore the amount of power available in the 1121 nm cavity to convert to 1178 nm light through Raman coupling. Therefore, it would not be prudent to set the *converge_tol* value too low such that the changes caused by the FWM would prevent the convergence criterium from ever being satisfied.

B.4 Results & Discussion

Currently, the fiber laser/amplifier model is able to run and achieve the convergence criterium for a fairly broad range of parameters in a very reasonable amount of time. The spacing for the discretized z-values can be as large as 10 cm, maybe even a bit larger, and still maintain fidelity with the true physical dynamics occurring in the fiber. The frequency spacing for the model is chosen according to the FBG bandwidth,

and more or less, to the width of the side-band peaks of reflectivity (see Fig. B.2).

It is worth noting that thermo-broadening of the energy levels of the medium's molecules ought to impose a lower limit to the frequency spacing in the model. This is because the broad energy levels would absorb and emit photons of similar frequencies as if they had the same frequency. This issue has not been investigated so far, and it is not clear whether this effect will ever be important enough to include in the numerical model. The free spectral range (FSR) of the cavity will also impose a lower limit to the frequency spacing.

Technically, in order to best model a true laser/amplifier system, each run of the model ought to be repeated multiple times, possibly on the order of thousands of times, and the results ought to be averaged. This is because each time the model is run, the initial phases of the amplitudes are randomly chosen, and as has been discussed earlier, this leads to slightly different results [82]. To this end, there is a *repetition* parameter that can be set in the model, which will cause the program to rerun the model the specified number of times. The code is even parallelized over theses repetitions, and can run each repetition on its own thread, up to the number of threads set by the *nthreads* parameter. However, it turns out that so far that most repeated runs are similar enough to one another to warrant confidence that only one or a few few runs is enough to capture the pertinent dynamics of the system.

Related to the repetition issue, is whether or not each initial amplitude phase ought to be chosen completely independent of the other initial amplitude phases. There are some reasons to expect that one phase may be related to the phases of the neighboring spectral components [104, 82]. First, one might argue that the molecules of the medium may have some coupling, possibly due to thermo- or lattice- vibrations, that could cause a certain degree of synchronization of the stimulated emission. Also, FWM mixing events may also cause neighboring phases to align as the light propagates in the fiber. The degree of correlation between the initial amplitude phases is mostly an unexplored issue; though because of phase detuning in the fiber, and due to the

phases changes induced by the FBGs, this may not be an important issue [104, 82, 93, 98].

According to the testing on this laser/amplifier model that has been completed so far, the most influential parameters on the results are the 1069 nm pump power (*P_1069_0*), the 1178 nm seed power (*P_1178_0*), the cavity/fiber length (L), and the numerical aperture in the fiber core (*NA_core*). These preliminary results seem to indicate that the laser/amplifier design ought to be successful in producing the desired output of 1178 nm power that is still narrow in bandwidth. The main limiting factors that will prevent the system from producing higher output powers are the amount of 1121 nm power that the FBGs can reasonably contain in the cavity, and the growth of the SBS power at 1178.06 nm. The power that can escape around the FBGs is of some concern because of the power that may leak back into the optical elements of the system and potential damage the equipment.

It has already been said that the model uses a split step method to propagate the power and amplitudes in the fiber. This method was chosen for its simplicity and apparent computational efficiency. However, as the model has progressed, it seems that a full and more realistic model composed of one large system of ODEs, using only amplitude equations, may be nearly as efficient in terms of computational time as the split step method. Also, a full system of ODEs would not be making any additional approximations to the physical system like the split step method does [124].

In the future, when the laser/amplifier system gets assembled in the laboratory, it would be very advantageous to measure the actual fibers of the system in order to obtain accurate parameter values. This is especially true for the indices of refraction in the fiber core and cladding, along with the corresponding numerical aperture. The GeO_2 doping concentration is definitely a parameter that can vary from fiber to fiber, and the Sellmeier equation is only an approximation which ought to be best-fit to the actual fibers being used in the system. Also, it would be better is measure the shape and size of the transverse modal distributions at the pertinent wavelengths of

the model because there are many non-physical assumptions needed to derive the LP_{01} Bessel function modal distribution (B.2). Plus, parameters like the Raman and Brillouin gain coefficients and the nonlinear index of refraction are very fiber dependent, even among supposedly identical fibers made by the same manufacturer.

The Bragg gratings that have been purchased for the laser/amplifier system are chirped, but because there is no documentation as to how or to what degree the gratings are chirped, the model assumes that the gratings are not chirped. The FBG spectral profiles should also be measured in the real fibers instead of approximated using the formulas outlined in this chapter.

Another way in which the current laser/amplifier model can be improved is by including more nonlinear effects in the model. This includes a more accurate model of the scattering and loss that occur in the system, and the leakage and scattering to and from the cladding region [117]. A third Stokes line above the 1178 nm wavelength and an anti-Stokes line below the 1069 nm wavelength could be included in the model. Though it is not expected that these extra Stokes lines will see any significant growth, they may still account for some of the loss in the system. Similarly, it may be prudent to add SBS effects to the 1121 nm wavelength, even though this wavelength is quite broad. Since the 1121 nm light is contained in a cavity, the power levels can be significant enough to see some small amounts of SBS. Plus, one could add FWM coupling between the fast and slow axes of the fiber, especially for the 1121 nm light, which may have power levels high enough to transfer small amounts of energy between the different polarizations [118, 82]. A less likely, yet still interesting, possibility would be the appearance of a temperature induced modal instability in the 1121 nm cavity.

Another topic that merits further investigation, is the growth of the Raman gain at a frequency shift of 490 cm^{-1}. At higher powers, the Raman gain peak, which is normally around 440 cm^{-1}, saturates and can be eclipsed by a growing peak at 490 cm^{-1} [88]. If the Raman gain at 490 cm^{-1} becomes significantly higher than the 440 cm^{-1} gain, than it may be beneficial to change the laser/amplifier system to

take advantage of the stronger Raman coupling that is available, which can lead to a higher 1178 nm output power [88]. It would also be wise to account for the Raman gain intensity dependence [114].

One last physical phenomenon that may play an important role in this laser/amplifier system that is not already included in the model is temperature. Temperature changes the physical properties of the fiber itself, which can have significant effects of the light propagating inside. Temperature and/or stress forces in the fiber can change the refractive indices of the fiber, and consequently change the polarization of the light, and they will most definitely cause some phase detuning as the light propagates down the fiber. Also, thermal effects can significantly change the properties and performance of the FBGs. Understanding how temperature changes alter the laser/amplifier system can be a fruitful topic of research.

This model of the Raman fiber laser/amplifier design is accomplished in the spectral domain, but it would also be possible to model the system in the time-domain. However, in the time-domain, handling FBG reflections and amplitude phase information becomes a much more difficult task, where even more assumptions have to be made. If the proposed laser/amplifier system has a true CW steady-state, then a time-domain model may actually be feasible if one initializes the model judiciously; but, if this system is inherently unstable either for some or all parameter choices, then the frequency-domain model will have a better chance of detecting this, and finding any stable regions.