Data Mining Applied to Laser Measurement

Author: Jonas Galdikas (H00200578)
Supervisor: Dr. Michael Lones

A thesis submitted in fulfilment of the requirements for the degree of MSc. Information Technology (Business) in the School of Mathematical and Computer Sciences

August 2015
Declaration of Authorship

I, Jonas GALDIKAS, declare that this thesis titled, 'Data Mining Applied to Laser Measurement' and the work presented in it is my own. I confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: 

Date: 

Abstract

Wavelength meters are used by scientists, universities and companies that require precise laser wavelength measurement for various reasons. High resolution measurements of laser wavelengths usually require expensive equipment. Thus, there is a need for low-cost, high-resolution spectrometers. Recent research shows that a pattern, produced when a laser is shone through a disordered media, creates a fingerprint speckle pattern that can be used to identify the wavelength.

This project, as part of the larger ongoing research, aims to apply and evaluate sophisticated image analysis and data mining techniques to develop a prototype software for speckle pattern based laser wavelength meter. Techniques of the image processing and feature extraction will be employed to translate the visual data, perceived through CCD camera, to a unique numerical representation which is then processed by predictive models to estimate the wavelength value.
I would like to thank Dr. Michael Adam Lones and Dr. Nikolaus Klaus Metzger for their time, guidance, suggestions and advices. Also, I would like to express my deepest gratitude to my family for their support and motivation. My grateful thanks goes to Vismantas D. for valuable discussions on complex matters. Last, but not least - a special thanks to Michael T. and Colin T. of ZypeTV for providing the work schedule flexibility I needed to complete this thesis.
# Contents

Declaration of Authorship ................................. i
Abstract ................................................................ ii
Acknowledgements ................................................. iii
Contents ................................................................ iv
List of Figures ....................................................... vi
List of Tables ......................................................... vii
Abbreviations ......................................................... viii

1 Introduction ....................................................... 1

2 Background and Literature Review ....................... 3
   2.1 Related Work .................................................. 3
   2.2 Image Analysis ............................................... 4
      2.2.1 Image Analysis Workflow ......................... 5
      2.2.2 Preprocessing ......................................... 6
      2.2.3 Feature Detection and Descriptor Extraction ...... 7
   2.3 Predictive modelling ....................................... 9
   2.4 Tools .......................................................... 10

3 Requirements Analysis ........................................ 12
   3.1 Project Aims .................................................. 12
   3.2 Project Objectives .......................................... 12
   3.3 Requirements ............................................... 12

4 Methodology and Research Design ....................... 14
   4.1 The Data ...................................................... 14
   4.2 Methodology ................................................ 16
      4.2.1 Principal Component Analysis (PCA) Based Approach ... 16
      4.2.2 Bag-of-Visual-Words (BoV) based approach ............. 18
      4.2.3 Evaluation ............................................. 20

5 Results and Analysis .......................................... 21
   5.1 PCA Experiment #1: The Baseline ..................... 21
5.1.1 Overview ......................................................... 21
5.1.2 Mean Squared Error (MSE) and Explained Variance Score (EVS) Analysis ......................................................... 22
5.1.3 Learning Curve Analysis ........................................ 23
5.1.4 Challenge Data Analysis ......................................... 23

5.2 PCA Experiment #2: Preprocessing .................................... 25
5.2.1 Overview ......................................................... 25
5.2.2 Contrast Stretching ................................................ 27
5.2.3 Histogram Equalization .......................................... 27
5.2.4 Adaptive Histogram Equalization .................................. 28
5.2.5 Local Histogram Equalization ..................................... 29
5.2.6 Gaussian Blur ...................................................... 30
5.2.7 Combined Preprocessing Techniques .............................. 31

5.3 PCA Experiment #3: Transformations .................................. 32
5.3.1 Overview ......................................................... 32
5.3.2 9 PC Model ....................................................... 33
5.3.3 19 PC Model ....................................................... 34

5.4 PCA Summary ......................................................... 34

5.5 BoV Experiment #1: Keypoints ......................................... 38
5.5.1 Overview ......................................................... 38
5.5.2 Results .......................................................... 38

5.6 BoV Experiment #2: Baseline .......................................... 41
5.6.1 Overview ......................................................... 41
5.6.2 Results .......................................................... 41

5.7 BoV Experiment #3: Preprocessing ...................................... 44
5.7.1 Overview ......................................................... 44
5.7.2 Results .......................................................... 44

5.8 BoV Experiment #4: Transformations .................................. 45
5.8.1 Overview ......................................................... 45
5.8.2 Binary Robust Invariant Scalable Keypoints (BRISK) Model ......................................................... 45
5.8.3 Maximally Stable Extremal Regions (MSER) Model ............. 46
5.8.4 Star (also known as CenSurE) (Star) Model ....................... 46
5.8.5 Speeded Up Robust Features (SURF) Model ....................... 46

5.9 BoV Summary ......................................................... 47

6 Professional, Legal, Ethical and Social Issues ........................ 48
6.1 Professional Issues ................................................... 48
6.2 Legal Issues ........................................................ 48
6.3 Ethical Issues ........................................................ 49
6.4 Social Issues ........................................................ 49

7 Further Work .......................................................... 50

8 Conclusions and Evaluation ............................................. 51

References ............................................................ 53
# List of Figures

2.1 Image Processing Workflow .................................................. 5  
4.1 Dataset item examples ....................................................... 15  
4.2 PCA experiment structure ................................................... 17  
4.3 PCA experiment structure ................................................... 19  
5.1 PCA baseline MSE and EVS .................................................. 22  
5.2 PCA experiment #1 learning curves ....................................... 24  
5.3 PCA experiment #1 sinusoids ................................................. 25  
5.4 PCA experiment #2 preprocessed images .................................. 26  
5.5 PCA experiment #2 contrast stretching ................................... 27  
5.6 PCA experiment #2 histogram equalization ............................... 28  
5.7 PCA experiment #2 adaptive histogram equalization .................... 29  
5.8 PCA experiment #2 local histogram equalization ....................... 30  
5.9 PCA experiment #2 Gaussian blur ......................................... 31  
5.10 PCA experiment #2: Adaptive histogram equalization with Gaussian Blur 32  
5.11 PCA experiment #3: Most improved 9 PC model ....................... 33  
5.12 PCA experiment #3: Most improved 19 PC model ..................... 34  
5.12 PCA 9 PC model all challenge data ....................................... 36  
5.13 PCA 19 PC model all challenge data ..................................... 37  
5.14 PCA final models estimations residuals .................................. 38  
5.14 Keypoint detectors .......................................................... 40  
5.15 MSE and EVS of the keypoint detectors .................................. 42  
5.15 BoV experiment 2 challenge data ......................................... 43  
5.16 BRISK experiment 4 challenge data ..................................... 45  
5.17 MSER experiment 4 challenge data ....................................... 46  
5.18 SURF experiment 4 challenge data ....................................... 47
List of Tables

2.1 Feature detection and description extraction algorithms summary . . . . . 9
5.1 PCA experiment #1 setup summary . . . . . . . . . . . . . . . . . . . . . 21
5.2 PCA experiment #2 setup summary . . . . . . . . . . . . . . . . . . . . . 26
5.3 PCA experiment #3 setup summary . . . . . . . . . . . . . . . . . . . . . 33
5.4 BoV experiment #2 setup summary . . . . . . . . . . . . . . . . . . . . . 41
5.5 BoV experiment #3 setup summary . . . . . . . . . . . . . . . . . . . . . 44
5.6 BoV experiment #3 results summary . . . . . . . . . . . . . . . . . . . . . 44
5.7 BoV experiment #4 setup summary . . . . . . . . . . . . . . . . . . . . . 45
Abbreviations

**AGAST**  Adaptive and Generic Accelerated Segment Test.

**BoV**  Bag-of-Visual-Words.

**BRIEF**  Binary Robust Independent Elementary Features.

**BRISK**  Binary Robust Invariant Scalable Keypoints.

**CCD**  Charge-Coupled Device.

**CW**  Continuous Wave.

**EVS**  Explained Variance Score.

**FAST**  Features from Accelerated Segment Test.

**FREAK**  Fast Retina Keypoint.

**FV**  Fisher Vectors.

**GFTT**  Good Features To Track.

**GMM**  Gaussian Mixture Model.

**KAZE**  KAZE Features.

**KNN**  K-Nearest Neighbours.

**MARS**  Multivariate Adaptive Regression Splines.

**MSE**  Mean Squared Error.

**MSER**  Maximally Stable Extremal Regions.
**OpenCV** Open Source Computer Vision Library.

**ORB** Oriented FAST and Rotated BRIEF.

**PCA** Principal Component Analysis.

**SIFT** Scale-Invariant Feature Transform.

**Star** Star (also known as CenSurE).

**SURF** Speeded Up Robust Features.

**SVM** Support Vector Machines.
Chapter 1

Introduction

A wavelength meter is a device used to precisely measure the wavelength of a laser beam [1] and providing far higher accuracy than a regular spectrometer or a grating monochromator. Its application varies from simple tasks like laser wavelength measuring to more complex, such as controlling a laser that is used to measure the distance to another planet [2].

The traditional high-accuracy devices use scanning Michelson or Fizeau interferometers to perform spectral to spacial mapping on photodetector arrays or Charge-Coupled Device (CCD) sensors. The scanning Michelson interferometer based wavemeters have an advantage of continuous calibration and can be used to measure over the range from 375 nm to 12 µm with an accuracy of up to 0.0001 nm [3]. However, the usage is limited to Continuous Wave (CW) lasers and it cannot be used with pulsed lasers, since it requires continuous observation of the fringes created by the interference. Fizeau interferometer based wavemeters on the other hand, are often used to measure the pulsed lasers, since they have no moving parts and capture the full frame. Though their functional range is restricted by the limitation of the photodetector arrays used in this type of devices, which makes it only able to work with wavelengths of lower than 1700 nm [3].

This project is a part of a larger research that focuses on creating a simple, compact, low-cost, high-resolution laser wavelength meter that uses a random diffuser and a CCD camera to capture the random speckle pattern. The main focus is on investigating
the application of advanced image analysis and data mining techniques on such speckle patterns to identify the laser wavelength.

The related work is overviewed and tools used in the research are introduced in Chapter 2, along with an introduction to image analysis techniques and predictive modelling approaches, followed by requirements analysis and research project details in Chapter 3. Chapter 4 introduces the data used in the research, the methodology used in the selected approaches, and the evaluation approach. Results of the experiments are provided and analyzed in Chapter 5, followed by brief consideration of the professional, legal, ethical and social issues that were involved in the project in the Chapter 6. The report is finalized with a suggestion of further work in Chapter 7 and the conclusions and project evaluation in Chapter 8.
Chapter 2

Background and Literature Review

In this chapter, related research is discussed and critically evaluated. The research that is the basis of this project is introduced as well, along with the software side approach used so far as a proof of concept. A brief introduction to general image analysis workflow follows with an overview of the image analysis and statistical analysis tools and techniques that have been considered for this project.

2.1 Related Work

It has been noticed that light propagation through a stationary random medium diffuser can produce a deterministic speckle pattern [4]. Such speckle pattern maintains its initial spatial and temporal coherence when the light source is a coherent beam, allowing reconstruction of the initial information based on the produced speckle pattern characteristics. This encouraged a number of research to create a low-cost high-resolution spectrometers, using a range of different approaches.

A multimode fiber was used as a dispersive element, where a speckle pattern is created by the interference among the guided modes [5]. The resolution of such spectrometer is dependent on the fiber length [6]. However, the downside of such method is the sensitiveness of the multimode fiber to the environment. A speckle pattern for the same wavelength changes due to the ambient temperature changes that influence the refractive
index of the glass fiber. Furthermore, the magnitude of pattern change is dependent
on the used fiber length which makes the spectrometer requiring re-calibration if the
temperature changes by $0.01 \, ^\circ C$ when using 100 m fiber [7]. Another research, that
this project is part of, used a thin diffuser as a scattering media [4].

Regardless of the scattering media used, be it a multimode fiber [6, 7, 5], a thin dif-
fuser [4], or any other material or technique, they all create random patterns of light
intensity and interference which is then digitized using CCD sensor arrays or cameras.
The main research so far was classifying the digitized image by using the transmission
matrices. One of the approaches to building transmission matrix was to use multivariate
PCA of the eigenvectors, achieving the accuracy of a picometer [4]. Another approach
was to build a $M \times N$ matrix, where $M$ is the number of speckles produced (thus, feature
vector size of the image is $M$), and $N$ is the number of training set items [5]. Such matrix
was built using training set of images taken in intervals of 0.5 picometer, although the
highest resolution reached was one picometer (using 100 meters of multimode fiber and
only for a wavelength of 1500 nm) [6].

2.2 Image Analysis

Common image based data mining applications require the image to be represented in
a descriptive form, which can be used for further computer processing. The description
consists of a set of selected features, defining the global characteristics of an image, and
the ones of the selected regions of interest, identified in the image. Image data contains
a large amount of information and it is of great importance to capture it while reducing
its loss as much as possible during the description building process.

In this section, the techniques used for building the image description are described
starting with the overall workflow of the image processing, further detailing on prepro-
cessing techniques, feature extraction and feature selection methods in the subsequent
sections. The phrases image analysis and image processing are used interchangeably and
in this context mean the image conversion into a feature vector that uniquely describes
it.
2.2.1 Image Analysis Workflow

The high-level overview of the image analysis workflow is displayed in Figure 2.1 [8]. It should be noted that not all the displayed steps are compulsory and their necessity is rather situation dependent. Thus, only the stages applicable to the project will be briefly explained.

The image analysis begins with the acquisition of the image from the source. Generally, it can vary from interpreting analog data and converting it into digital representation, to having an image already in a digital format. If any image adjustment is required to normalize, filter, enhance or restore the image data, it is performed in preprocessing stage. Preprocessing will be discussed in greater detail in section 2.2.2.

![Figure 2.1: Image analysis workflow. [8]](image)

After preprocessing, the next step is to convert the information from the image to a machine processable form. There are different approaches that can be applied to achieve this - using global features, image segmentation, sampled features and local features. Regardless the approach used, the main goal in this project’s context is to produce representation of an image that has maximum similarity within the allowed
wavelength error, while maintaining minimal similarity to detect when it changes more than allowed error. In this project, two main approaches were chosen.

The first one builds upon the approach taken by the original research, where PCA was used to compute the feature vector. Since positive results were observed, it is a good baseline for further improvement.

The second approach focuses on application of local feature oriented algorithms, since they better describe the contents of the image rather than the image as a whole [9]. Information extraction using local features is a two-step process, where the areas of interest are identified and then their descriptions extracted. Feature encoding techniques can be used to produce a global descriptor, reflecting the overall contents of the image. Methods used in both of these approaches are described in more details in section 2.2.3.

2.2.2 Preprocessing

While digital imaging sensors are constantly improving, resulting in increased resolution and image quality, there are still problems to tackle before the feature extraction step. Image preprocessing is an important stage in the fields involving any kind of image perception [8, 10]. The main goal is to equalize the input images that may be acquired from different sources, having various resolutions, luminosity, different levels of noise, etc., and enhance their distinctive details. Essentially, it improves the image analysis process allowing more accurate feature detection and descriptor extraction, although overprocessing can skew or even destroy the image data, resulting in false outcome. Preprocessing encompasses image manipulation techniques, such as, intensity transformations, spatial filters, Fourier transforms, to name a few [11].

Intensity transformations are some of the simplest image processing techniques. It comprises of three main classes of operators: point, local and global. When using point operators, the output value at a specific coordinate is dependent only on the input value at that same coordinate. Local operators take into account the surrounding area of the specific coordinate when calculating its value. The neighbouring coordinates used in processing can be selected in a number of different ways: 4-connected rectangular sampling, 8-connected rectangular sampling, weighted sampling, etc. The pixel value when the global operators are used depends on all the values in the input image [11].
Examples of such techniques are Gaussian smoothing [12], where the image is blurred, reducing noise and eliminating weak features; histogram equalization [13], where the image intensity histogram is reshaped by reassigning pixel intensity values based on particular mapping, enhancing the details that might not have been visible at all. The specific transformations will be described in more details further in the paper.

Aforementioned transformations manipulate the image in the spatial representation. Fourier transform is used to decompose the image from its spatial representation into its sine and cosine components, resulting in the image, represented in the frequency domain without losing information [14]. It can be used for noise removal and image reconstruction during preprocessing stage, and as a shape descriptor during feature extraction stage.

### 2.2.3 Feature Detection and Descriptor Extraction

In order to apply predictive modelling techniques in a context where the data is represented by images, the image data needs to be converted into numeric format. Measurements, taken from the image properties and regions around identified local features, can be converted into descriptors that identify distinguishing properties of the image. The application that involves image analysis can be only as accurate as its image analysis part is. Thus, it is important to identify good features. In general, good features should have the following properties [9]:

- **Repeatability** - given a number of images of the same wavelength, produced using the same spectrometer setup, the feature should be found in all of them.

- **Distinctiveness/informativeness** - the representation of the feature should capture sufficient level of variation in feature’s underlying intensity pattern.

- **Locality** - feature region should not be too big in order to minimise the possibility of occlusion, geometric and photometric deformation influence, and, more importantly in this project’s context, to reduce the required computational resources.

- **Quantity** - sufficient number of features should be detected, depending on the application requirements.
• Accuracy - detected features should be accurately localized in terms of image location (and image scale, which is not applicable in this context).

• Efficiency - the detection process should be fast, preferably allowing real-time processing.

For the purposes of this project, a global feature vector of an image is required. Multiple approaches exist for building a global feature vector. PCA has been used to capture the global properties of an image by projecting the mean-shifted image data into N-dimensional eigenspace. Another approach is to extract the local features and then use encoding techniques to form a global feature vector. Local features are points, edges or small image patches that are distinctive in their immediate region [9].

A number of algorithms have been developed to identify and describe such features. Scale-Invariant Feature Transform (SIFT) is considered to be one of the best quality features providing algorithm [15], and regardless of being relatively old (first introduced in 1999), it is often used as a standard when evaluating performance of the state-of-the-art algorithms. Table 2.1 summarizes some of the best known algorithms that aim to improve on SIFT performance. Open Source Computer Vision Library (OpenCV) has native implementations of the majority of these cutting-edge feature detectors and description extractors. Furthermore, OpenCV allows any additional algorithms being added by extending relevant classes.

The aforementioned algorithms detect and extract a large number of features from an image. This is not helpful unless the features can make sense in a global context. Feature encoding algorithms can be used to compute a global image descriptor from a large set of local feature vectors. This approach is called the BoV approach.

BoV uses the local features to compile a codebook which is used to compute the global feature vector [25]. The codebook is built by clustering a variety of features extracted from a training data set. The most basic approach is the histogram encoding, where K-means clustering is used to compute a chosen number N of cluster centres, also known as the visual words. The image local features are then quantized into N-dimensional histogram of visual words occurrences, where each local feature vector is assigned to the closest visual word. However, the distance of each feature vector from the assigned cluster centre is not accounted for, essentially losing some of the information. Fisher
## Table 2.1: Feature detection and description extraction algorithms summary

<table>
<thead>
<tr>
<th>Title</th>
<th>Detector</th>
<th>Descriptor</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale-Invariant Feature Transform (SIFT) [16]</td>
<td>Yes</td>
<td>Yes</td>
<td>Provides robust, accurate 128 dimension feature vector, but computationally expensive and sensitive to blur and lighting changes. Uses Difference of Gaussian and Scale Extrema Detection.</td>
</tr>
<tr>
<td>Speeded Up Robust Features (SURF) [17]</td>
<td>Yes</td>
<td>Yes</td>
<td>Based on SIFT algorithm. Employed box-filters to achieve fast approximation of the Hessian matrix and Gaussian derivatives. Produces 64 dimensional feature vector for each identified region of interest. Slightly less accurate than SIFT, but claimed to be 3-5 times faster and more resilient to noise. Algorithm is patented.</td>
</tr>
<tr>
<td>Features from Accelerated Segment Test (FAST) [18]</td>
<td>Yes</td>
<td>No</td>
<td>A corner detection algorithm, efficient enough for real-time video processing. Generates and compiles decision tree code in C.</td>
</tr>
<tr>
<td>Binary Robust Independent Elementary Features (BRIEF) [19]</td>
<td>No</td>
<td>Yes</td>
<td>Binary intensity comparison tests based descriptor. Claimed to provide better results in a fraction of the time required by SURF. Sensitive to rotation.</td>
</tr>
<tr>
<td>Oriented FAST and Rotated BRIEF (ORB) [20]</td>
<td>Yes</td>
<td>Yes</td>
<td>Based on Features from Accelerated Segment Test (FAST) detector and Binary Robust Independent Elementary Features (BRIEF) descriptor. Claimed to be more accurate than SURF and fast enough for real-time applications.</td>
</tr>
<tr>
<td>Adaptive and Generic Accelerated Segment Test (AGAST) [21]</td>
<td>Yes</td>
<td>No</td>
<td>FAST based corner detector with improved computational performance</td>
</tr>
<tr>
<td>Binary Robust Invariant Scalable Keypoints (BRISK) [22]</td>
<td>Yes</td>
<td>Yes</td>
<td>Uses AGAST to detect features, uses concentric circles around the features as a symmetric pattern to extract the description</td>
</tr>
<tr>
<td>Fast Retina Keypoint (FREAK) [23]</td>
<td>No</td>
<td>Yes</td>
<td>Human eye retina inspired algorithm. Provides fast computation with low memory memory load, making it attractive for use in embedded systems. Claimed to be more robust than SIFT, SURF or BRISK</td>
</tr>
</tbody>
</table>

Vectors (FV) [26] is another kind of BoV approach, where the Gaussian Mixture Model (GMM) is used to build a codebook. Its advantage is that it also stores the second-order information about the features, taking into account the distance to the cluster centre, which results in a high-dimensional feature vector, which might be disadvantageous when working with large data sets.

### 2.3 Predictive modelling

Predictive modelling can be summarized as a process of finding an underlying mathematical relationship between a set of independent variables, also called features or predictors, and a dependent variable, also called response variable or a result [27]. The process encompasses machine learning, pattern recognition, data mining, statistical mathematics, and other techniques to build a predictive model.
The result can be generalised as being a discrete or a continuous variable. When the response variable is discrete, classification models are used to separate the dataset into classes, defined by the response variable. When the result is continuous, regression models are used. In this project, the focus is on the latter category, since wave length is a continuous variable. Regression models can be split into three groups - regression-type models, nonlinear regression models, and regression trees and rule based models [27].

The first of the aforementioned groups includes ordinary linear regression, partial least squares, ridge regression, lasso, to name a few [27]. The models derive a formula, that generally includes a sum of weighted features, estimated intercept and an error value that the model cannot explain. Advantage of such models is their interpretability. However, in order to capture more complex relationship (e.g. quadratic, cubic, etc.) between the predictors and the response variable, additional techniques, such as model predictor augmenting, are required. The nonlinear models include neural networks, Multivariate Adaptive Regression Splines (MARS), Support Vector Machines (SVM), K-Nearest Neighbours (KNN) and others. These typically use their own generated intermediate features and non-linear functions. The last group is rather separated type of nonlinear regression models, where a result is achieved by following a path in the tree, based on defined rules and conditions.

When the relation between the independent variable and the dependent variable is nonlinear, the linear regression models might not capture the relation well enough, leading to large estimation errors. One way to counter this is to transform the data [28]. Transforming the independent variable using non-linear functions, such as, logarithm, cubic root or any other, can linearize the relation, allowing for better prediction of the unseen data. The data transformations can also be applied as a feature engineering technique to augment the feature vector with its own transformations, adding local complexity.

2.4 Tools

After initial prototyping stage, Python programming language was chosen over C++ for its ease of use, fast prototyping speed, clear syntax and sufficient execution speed. Moreover, its statistical and scientific packages make it a fair competitor against other more purpose-built languages, such as, Matlab and R.
OpenCV, as the name describes, is an open-source computer vision and machine learning software package, and is widely used by people ranging from individual enthusiasts to technology giants like Google, Microsoft, Toyota to name a few [29]. It is released under BSD licence [30], making it suitable for this project due to the fact that the image processing part can be based on it both in research and the production stages, free of charge [31]. Furthermore, it encompasses more than 2500 optimized algorithms including the classic ones mentioned in the previous sections, and state-of-the-art algorithms in the fields of computer vision and machine learning [29]. It is natively written in C++, which makes it suitable for real-time video processing, has interfaces with a number of popular programming languages (C, C++, Java, Matlab, Python), and supports Windows, Linux and Mac OS operating systems, meaning it can run on almost any device. In this project, the library will be used to reduce the overhead of implementing the required algorithms in image processing part of the software.

SciPy [32] is a collection of open-source software written in Python for mathematics, science, and engineering. It is released under BSD licence allowing its use in open source and commercial applications with minimal restrictions. A number of packages from this collection were used. NumPy is a fundamental package for any statistical or scientific data manipulation and is a requirement for other packages used in this project. It enables matrix manipulation in Python through multi-dimensional arrays, also provides linear algebra, Fourier Transform implementations amongst others. Matplotlib is a 2-D and 3-D plotting library. SciPy Library provides a collection of numerical algorithms.

Scikit-learn [33] is a collection of simple and easy to use tools for data mining and data analysis, based on the aforementioned NumPy, SciPy Library and Matplotlib libraries. It is released under the BSD licence, allowing its use with minimal restrictions.

Scikit-image [34] is a community-developed open source image processing library for Python. Although it has less computer vision algorithms than OpenCV and is not as efficient in non-linear operations, being written in Python and having a better documentation makes it more convenient for prototyping.
Chapter 3

Requirements Analysis

3.1 Project Aims

The project aim was to apply the data mining techniques on image data produced using a novel low-cost laser wavelength measurement technique, in order to improve its software-side approach.

3.2 Project Objectives

The objectives for fulfilling the project aims were as follows:

- Gain an understanding of the image global feature extraction techniques
- Gain an understanding of the predictive models
- Design and develop the experiments to test the selected approaches using the provided test data
- Analyze the results and suggest the further research

3.3 Requirements

The project requirements (research goals) were identified as follows:
• [mandatory] Regression model should be used to estimate the wavelength value, in order to output a continuous variable, allowing to notice the small change of the laser wavelength

• [optional] Predictive model accuracy of 1 picometer

• [mandatory] Predictive model should have enough generalization to estimate the wavelength values beyond the trained interval when the laser wavelength drift occurs

• [mandatory] Develop experiment scripts to support the hypotheses

• [optional] Camera input should be supported by the test scripts
Chapter 4

Methodology and Research Design

In this chapter, the methodology used throughout the project is introduced by overviewing the test data set, followed by explanation and reasoning behind the chosen approaches, and finalized by discussion on evaluation criteria. Two main approaches were chosen. Namely, the PCA based which builds upon the previous research results, and the BoV approach, which aims to attempt to apply the advanced computer vision algorithms.

4.1 The Data

The dataset consists of 7 sets of images with known wavelength (~50 images each) varying between 799.999 nm to 780.092 nm, and 6 sets of images with unknown wavelengths inbetween known values. All the sets are also marked in terms of laser voltage used when the images were taken. These vary from 0 to 6 volts inclusive, with 0.5 volt granularity. Thus, the regression outcome can be either the voltage or the actual wavelength for this particular dataset, although only the actual wavelength values were used.

The dataset also includes two sets of challenge images, where the laser beam is controlled by an oscillator, oscillating the laser beam wavelength within the interval covered by training dataset, at 1 Hz frequency. These two sets will be used to plot the oscillation
curve, which will be further used as an evaluation tool. Figure 4.1 displays example images of three different wavelengths.

![Example Images](image)

**Figure 4.1:** Dataset item examples. A visible difference can be noticed when comparing the images.

Upon closer visual inspection of the provided example images, two main texture changes of this particular dataset can be noticed that contribute to the uniqueness of the produced speckle pattern. The first one is the change of the interference ripples. They have a slight movement, which occurs due to dependency on the phases differences of the dispersed wave at the projection plane, which in this case is the camera sensor. Likewise, the sharpness of their minima and maxima boundaries vary, changing the patterns within the brighter areas. The brighter areas are the second noticed dynamic image pattern. The same already mentioned wave interference influences their shape, intensity and sharpness change, supplementing the distinctiveness of the speckle pattern. This suggests that the features will most likely be based on the extracted parameters of such areas that can be identified throughout the full range of the target wavelength interval spectra. It should be noted that there are other diffuser-specific parameters that impact the speckle pattern, but they will not be discussed here.

The dataset with known values and the challenge data set have a fundamental difference. The images with known values were produced during the same wavelength oscillation.
as the challenge one. However, the former images are not subject to laser wavelength drifting, since the images were produced at that exact value. When creating the challenge data, the laser drift was not countered. Furthermore, the laser wavelength drift is not uncommon when measuring at such small scale, and this should be taken into account during evaluation.

4.2 Methodology

Since the problem of the project implies a great variety of possible configurations, it was important to reduce that number as much as possible to be able to perform a more in-depth analysis. Hence, a decision was made to select an approach that was based on an initial research, and another one based on application of local feature vectors. Thus, the first approach used PCA to derive a global feature vector, and the second one used a BoV approach, where a number of local feature vectors are used to form a global one. Both approaches were divided into sets of experiments, described in further sections. After the initial prototyping with a range of regression model provided by the Scikit-learn package, it was decided to choose a simple linear regression model and focus more on the feature vector extraction part of the process. Finding the optimal parameters and taking into account the specifics of the more complex regression models would have taken far too much of the valuable project time. Furthermore, the in-depth statistical analysis to optimize the regression models trained using the produced feature vector extraction methods is large enough to be a separate project itself.

4.2.1 PCA Based Approach

The approach used PCA to extract a global descriptor from the image data. The general structure of the experiments is provided in Figure 4.2. Firstly, the training data is loaded. Each image is loaded in grayscale colour space, preprocessed using provided preprocessing function for the experiment and transformed into a vector. The training data is then normalized and the PCA model is trained.

The training data with known values and the validation data set are then projected using the trained aforementioned model, followed by the feature vector extension using the
feature augmentation function, provided as a parameter for the experiment. A regression model is then trained using the extended training data set and used in a set of tests.

The first test is the learning curve test, where the data is split 10 times into random subsets of training and validation data sets for cross-validation, the former consisting of 80% of the data and the latter one the remaining 20%. For each split, the training data is further split 20 times into subsets varying from 10% to 100% of its size to train the model and derive its learning curve. The default regression model’s scoring function is used for calculating the training and validation scores. The scores are then plotted on a graph and saved into experiment’s directory.

The following test trains the new model using the full training data set and obtains the model estimations for the same training data. The MSE and EVS are calculated using the predicted and actual values. The validation data set estimations are computed and the same metrics are calculated. The values are logged into the log file.

The last test performs an estimation on two challenge data sets (or their subsets in some tests for clarity) separately, plotting the results and saving the graph into experiment’s directory.

To summarize, the experiment script has the following independent variables: image preprocessing function, number of principal components that defines the feature vector dimension size, data augmentation/expansion function that extends the feature vector by appending its transformations, and the regression model to be used for estimations.

Five different image preprocessing functions were used from the Scikit-image package to enhance the image features. Number of PCA principal components define the feature vector dimension size. If a value is too small, the projected vector will not have enough information and too large will include additional noise, leading to predicting outlier
values. The data augmentation function helps to linearize the relation between the independent variables and the regression result by applying non-linear re-expression of the feature vector. Linear regression model was used for all the experiments. The more specific experiments parameters will be specified in the Chapter 5, where their results will be presented and analyzed.

The PCA training data set consists of the data set with known exact values (experiment training data set) and the images with unknown values inbetween the known ones. This was expected to allow for identification of the principal components of the full wavelength range of the training data set.

The experiments were setup to provide reduced number of possible variations of the variable values analysed to their successors. A baseline for further improvement was set in the first experiment by finding the optimal number of principal components to use for feature vector projection. The possible values of the number of principal components have then been fixed and the impact of preprocessing techniques was analyzed in the second experiment. Selected values of number of principal components and the preprocessing techniques based on the previous experiments have been used with a range of data transformation functions to attempt to capture any non-linearity in the third experiment. The final model setups then have been used on full challenge data set as the final evaluation.

4.2.2 BoV based approach

The set of BoV approaches was discovered after the extensive research into forming a global image feature vector based on local feature descriptors. The histogram bag-of-words, Fisher Vectors and VLAD - all of them could be applied to the project’s problem. The latter two techniques are relatively new and none of the used software packages provided their implementations. OpenCV provides a K-means clustering based histogram bag-of-words implementation, which was chosen to be used as an alternative approach to PCA based analysis, allowing to test it along with the advanced local feature detectors and descriptors.

There are a few disadvantages of the used BoV approach. When a local feature vector is assigned to the closest cluster, the distance between them is discarded, loosing valuable
information which might be of crucial importance when trying to notice every slightest change between the images. Another disadvantage lies within the actual implementation of the K-means clustering algorithm of OpenCV. OpenCV has two types of descriptors in terms of representation - binary and non-binary. The binary descriptors return a feature vector, where each dimension value represents a binary vector in decimal base, while the non-binary descriptors return the floating point values. The OpenCV K-means clustering algorithm uses the Euclidean distance which is absolutely fine for non-binary descriptor vectors. However, the binary descriptor clustering requires Hamming distance to be used to obtain valid cluster centres. Thus, the approach uses a SURF descriptor, which returns a floating point feature vector, in combination with different feature keypoint detectors.

The general structure of the experiment is provided in Figure 4.3. A few examples of each of the wavelength values were loaded and their extracted local descriptors were used to train a vocabulary for the BoV model. Training data and the validation data were then loaded and their local features encoded using the BoV model. Following, a regression model was trained and the same set of tests performed as in the PCA approach.

![Figure 4.3: PCA experiment structure](image)

The experiments used the same preprocessing and feature vector extension techniques as the PCA approach. A keypoint detector analysis was performed in the first experiment to select the ones with different characteristics. Optimal feature vector sizes were determined for each detector in the second experiment. The third experiment tested whether image preprocessing had any positive impact on estimation, and the fourth did the same using the data extension functions. The final model setups then have been used on full challenge data set as the final evaluation.
4.2.3 Evaluation

Evaluation is an important part of the project. There are multiple aspects to consider when evaluating regression performance. Do the feature vectors contain enough information to reflect the output variance within the required error boundaries and is the model able to capture it? How accurately does the model predict the outcome?

The regression model performance was measured in terms of EVS and MSE. EVS \[35\] represents how much variance of the data the model accounts for. The value is between 0 and 1, the former meaning that the regression predictions are meaningless and the latter meaning that all the predictions were correct with perfect accuracy. MSE \[33\] is the mean squared prediction error. Alternatively, the predictive models internally tend to use the R\(^2\) \[33\] score, which is the coefficient of determination. It represents how much of the variance of the dependent variable was accounted for by the predictive power of the independent variable.

In this project, the evaluation is quite complex. EVS and MSE are important metrics that allow to evaluate how well the model fits the data when the expected values are known. However, the aforementioned metrics cannot be applied to evaluate the estimations of the challenge data sets. Furthermore, the models were trained and validated using the data with known values from the same data set, which did not train the model to take into account any changes in environment that might occur, such as, laser wavelength drifting. Thus, the visual analysis of the challenge data estimations was part of the evaluation.
Chapter 5

Results and Analysis

This chapter overviews the details of the experiments and the results attained, along with their evaluation and analysis.

5.1 PCA Experiment #1: The Baseline

5.1.1 Overview

The goal of this experiment was twofold. A baseline was required in order to evaluate the improved models as well as finding an optimal number of principal components to be used in subsequent experiment setups. The experiment’s setup is summarized in Table 5.1. No image preprocessing nor feature vector expansion was performed to obtain the plain PCA performance results. The experiment was executed with 45 different principal component values, ranging from 5 to 49, both inclusive. A linear regression was chosen as the baseline regression model, being one of the most simple, but accurate models.

<table>
<thead>
<tr>
<th>Preprocessing function</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of principal components</td>
<td>range [5-49]</td>
</tr>
<tr>
<td>Feature vector expansion function</td>
<td>None</td>
</tr>
<tr>
<td>Regression models</td>
<td>Linear regression</td>
</tr>
</tbody>
</table>

Table 5.1: PCA experiment #1 setup summary
5.1.2 MSE and EVS Analysis

Figure 5.1 visualizes how the MSE and EVS metrics changed depending on number of principal components used to project the feature vector. The ideal setup would have the EVS value of the training and, more importantly, validation data sets equal to one, meaning that the regression model was able to fit a perfectly matching curve to the data. The MSE values in that case would be 0, meaning the model predicts a perfectly correct value without an error.

![Figure 5.1: Linear regression mean squared error and explained variance score dependence on number of principal components](image)

Based on the graph, it can be noticed that the EVS of the training set spikes to an almost perfect fit when the number of principal components value is 7. Onwards, the fit only gets better with reaching a perfect fit with 40-dimensional feature vector. The validation data set EVS has top values when the number of principal components value is between 7 and 12, and the peak is with 9. The further increasing feature vector dimensionality gradually reduces the EVS value. This means that the increase of the feature vector beyond 12 dimensions imposes the model to overfit the data, correctly estimating more training data set items, but failing to correctly predict the unseen examples.
The MSE values only further support the analysis demonstrating that while the error of estimation for the training data set decreases to a perfect fit, the validation data set prediction error gradually increases. Thus, the 9 principal components were chosen as the most optimal, providing the best fit for the unseen data. Another interesting point can be noticed in the graph where the number of principal components is 19. The metrics values jump out of the trend line, identifying a better performance than neighbouring values. This suggests that the combination of the projected 19 principal components might capture some additional useful information when compared to the 9-dimensional feature vector. Hence, this value was selected for further investigation as well.

5.1.3 Learning Curve Analysis

It is important to know whether the training data set size is sufficient for training a good regression model. Figure 5.2 shows a few examples of the regression model cross-validation and the learning curves. The scoring used by linear regression and hence, for the aforementioned scores calculation, is the $R^2$ score. Based on the graphs, the minimum training sample size should be at least 60 images. The cross-validation variance (green area) in (c-f) can be explained by the lack of training set variety as a result of a random shuffle split. This supports the claim that the increase in feature vector dimensionality increases the error of unseen data estimation.

5.1.4 Challenge Data Analysis

The challenge data estimation of the first 100 images (for graph clarity reasons) using the previously selected numbers of principal components is provided in Figure 5.3. The estimated values are slightly above the initial training set interval. This can be explained by the laser wavelength sensitivity to the environment. The data set in the second sinusoid appears to be more drifted upwards than the first one.

The predictions of both models formed a sinusoidal curve. The first difference between the models can be noticed by looking at the extremum areas. These areas, as expected, showed poor performance of the 9 PC model to detect the fine changes in the image data, predicting the same value. On the other hand, the 19 PC model predicted nice sinusoid curves in the lower extremum area, however, the top extremum area was problematic for
both models. Investigation of the predicted values showed a difference of around 0.004 nm for the equivalent points in the set. The amplitude of the training data set was 0.093 nm, while the both models predicted much smaller amplitudes, indicating that
the predicted values most likely have a large error.

![Graphs showing Sinusoid #1 and Sinusoid #2 data estimation with 9 and 19 principal components.](image)

Figure 5.3: PCA experiment #1 challenge data predictions

### 5.2 PCA Experiment #2: Preprocessing

#### 5.2.1 Overview

The experiment was designed to analyze how different image details enhancing techniques affect the regression accuracy. The setup summary is provided in Table 5.2. The difference from the previous experiment is that instead of a range of number of principal components, only the selected two values have been used, and a range of image preprocessing functions were added, provided by the Scikit-Image package [34].

The outcome images of the applied preprocessing functions are provided in Figure 5.4. Contrast stretching (b) is a simple image enhancement technique, where a linear scaling
Chapter 5. \textit{Results and Analysis}

Preprocessing function

- Contrast stretching
- Histogram equalization
- Adaptive histogram equalization
- Local histogram image equalization
- Gaussian blur with sigma in range $[1-10]$

Number of principal components

9 and 19

Feature vector expansion function

None

Regression models

Linear regression

Table 5.2: PCA experiment #2 setup summary

Function is applied to adjust the range of intensity values it contains to span a desired range of values, improving the image contrast. Histogram equalization (c) adjusts the image contrast so that all the pixel intensity values would be equally common, resulting in a flattened histogram. Adaptive histogram equalization (d) uses histograms computed over different regions of the image which enables enhancement of more fine-grain local features in darker areas of the image. The local histogram image equalization (e) uses histogram computed from the pixel neighbourhood to adjust the pixel intensity value. Gaussian blur (f) smoothes the image, reducing the noise. Its parameter $\textit{sigma}$ is the standard deviation for the kernel, controlling the amount of blur. Different levels of blur have been tested in this experiment and only the one with the best performance is discussed here.

![PCA experiment #2 preprocessed images](image)

(a) Original image  (b) Contrast stretching  (c) Histogram equalization

(d) Adaptive histogram eq.  (e) Local histogram eq.  (f) Gaussian blur, $\textit{sigma}=9$

\textbf{Figure 5.4: PCA experiment #2 preprocessed images}
5.2.2 Contrast Stretching

Contrast stretching enhanced the colours of the image. The sinusoid amplitudes of both models were still much smaller than they should have been. The shape of the lower extremum area of the 9 PC model was improved in sinusoid #2. The changes for the 19 PC model were noticeable by the improved extremum area shapes in both sinusoids. The estimation created a more smooth curve with fewer outlier values.

![Figure 5.5: PCA experiment #2 contrast stretching](image)

5.2.3 Histogram Equalization

After histogram equalization, the 9 PC model estimated values were slightly higher compared to the baseline. The top extremum area became more flat and some overestimated values appeared in the lower area. For the 19 PC model, the preprocessing method only introduced more noise. The effect on amplitudes was negative for both models.
5.2.4 Adaptive Histogram Equalization

The amplitudes for 9 PC model have increased however, the estimation towards the extremum values tendency was noticed, resulting in very wide flat sinusoid bottom. For the 19 PC model, the sinusoid amplitude has shrunk and outlier estimations appeared in the lower extremum area.
5.2.5 Local Histogram Equalization

The local histogram equalization extracted the texture features from the image. This resulted in estimations being grouped around two lines for both models with a lot of outlier estimations.
5.2.6 Gaussian Blur

Gaussian blur with sigma value equal to 9 provided the best looking sinusoids from all the tested variations. The sinusoid amplitudes increased for both models although, not enough to match the supposed amplitude value. The 9 PC model did not estimate the lower values very well, resulting in flat lower extremum areas of the sinusoid. On the other hand, the 19 PC model estimations formed a very reasonably looking sinusoid, although of lower amplitude than it should have been.
5.2.7 Combined Preprocessing Techniques

Based on the results of the previous experiment setups, it was decided to run another set of experiment setups, where the Gaussian smoothing with sigma value 9 would be applied together with each of the other four image enhancing functions. The results did not show any improvements worth discussion apart from a combination of Gaussian smoothing and histogram equalization (Figure 5.10) using 19 PC model. The combined preprocessing technique improved the curvature of the sinusoid at the extremum areas.
Chapter 5. Results and Analysis

5.3 PCA Experiment #3: Transformations

5.3.1 Overview

The experiment was designed to analyze how different feature vector transformation functions affect the regression accuracy. The setup summary is provided in Table 5.3. The adaptive histogram equalization with Gaussian blur and the Gaussian blur pre-processing functions were selected based on the previous experiment results. Different types of transformation functions were chosen for feature vector augmentation to attempt to capture any non-linearity in the data. The transformation function was applied to each dimension in the feature vector and the transformed vector appended to the original, doubling the feature vector dimensionality. 36 different experiment setups were executed, but only the setups with the most improvement will be discussed.
Chapter 5. Results and Analysis

33

Preprocessing function

- None
- Adaptive histogram equalization with Gaussian blur
- Gaussian blur

Number of principal components

9 and 19

Feature vector expansion function

- $\sqrt{x}$
- $\sin(x)$
- $\sin(x^2)$
- $\cos(x)$
- $\ln(x)$
- $x^2$

Regression models

Linear regression

Table 5.3: PCA experiment #3 setup summary

5.3.2 9 PC Model

The experiment setup with Gaussian blur preprocessing and $x^2$ data transformation function resulted in the closest amplitude range to the real value (0.090 nm). The training data set was fit perfectly by the model and the MSE of the validation data set predictions was $1.24378109447e^{-07}$, which is marginally larger than the baseline model allowing for more generalization when estimating the unseen data. Estimations in top extremum areas form curves similar to real sinusoid shape and the lower extremum area has some outlier values. The predictions inbetween the mid-area look Nevertheless, based on the known facts about the challenge data, the model provides the closest estimates to the expected values, along with very little estimation error of the validation data.

![Sinusoid #1](image1.png)

![Sinusoid #2](image2.png)

Figure 5.11: PCA experiment #3: Most improved 9 PC model
### 5.3.3 19 PC Model

The 19 PC model had the closest to the expected sinusoid estimation results using Gaussian blur preprocessing and $x^2$ data transformation function. The amplitude was larger than the training data one (0.102 nm) and, interestingly, the second sinusoid drift was negative, never seen in other models results. Again, it is impossible to verify whether this is correct due to unmonitored laser drift when creating the test data.

The top extremum and mid-areas appear to follow the sinus curve quite uniformly in both graphs, but the lower part of the sinusoids have outlier values, which appears to be a problematic area for all the tested models. The training data was fit perfectly and the validation data set MSE was $9.45273631796 \times 10^{-8}$, which is smaller than the 9 PC model.

![Sinusoid #1](image1.png) ![Sinusoid #2](image2.png)

*Figure 5.12: PCA experiment #3: Most improved 19 PC model*

### 5.4 PCA Summary

Two different baseline models were selected for further analysis after analyzing the implications of projecting the image in terms of a range of principal components. Applying various image preprocessing and data transformation techniques, the final two models were presented. Both models were able to fit the training data perfectly and the validation data set mean squared errors were very little. The image smoothing using Gaussian blur, even though eliminating the fine distinctive features initially thought to be of most importance, such as, the interference ripples, demonstrated an improved ability to handle the unseen data. From a range of different data transformation functions, the $x^2$
transformation improved the prediction granularity of both models when estimating the challenge data. Figures 5.12 and 5.13 display the estimations over all the challenge data samples. The estimation values do not form smooth sinusoid curves at the lower end of the range, which suggests that the model is not general enough to fully handle the change of the conditions, if any occurred during the training and the challenge data creation. A more in-depth feature vector statistical analysis could be applied to investigate the differences between the data sets, but this is out of the scope of this project.

![Image of Sinusoid #1](image_url)
Figure 5.12: **PCA 9 PC model all challenge data estimations**
Figure 5.13: PCA 19 PC model all challenge data estimations
The validation data set estimation residuals graphs in Figure 5.14 show that both models achieved accuracy of 0.001 nm when the estimated values were rounded to three decimal places to match the granularity of the known values.

![Residuals graphs for 9 PC model and 19 PC model](image)

**Figure 5.14:** PCA final models validation data set estimation residuals

## 5.5 BoV Experiment #1: Keypoints

### 5.5.1 Overview

The goal of this experiment was to investigate the features detected by the different feature detection algorithms on the same image and to choose the most interesting ones for further experiments.

### 5.5.2 Results

Majority of the algorithms detected the features along the interference ripples. Adaptive and Generic Accelerated Segment Test (AGAST) (a) and FAST (c), both being based on accelerated segment testing, detected nearly identical keypoints. The Good Features To Track (GFTT) (d) algorithm and the BRISK (b) detected very similar keypoints as the former two. All four tend to detect many overlapping keypoints, especially in the brightest areas. KAZE Features (KAZE) (e) detector mainly covered the brightest areas with less overlapping keypoints. The MSER (f), Oriented FAST and Rotated BRIEF (ORB) (g) and Star (i) detected the least number of keypoints, since they search more stable looking features. Nevertheless, their identified keypoints patterns are completely
different. **MSER** detected keypoints are very sparse, spread out throughout the whole image. **ORB** ones are condensed around the most texture-rich areas of the image while the **Star** keypoints are in the brighter areas of the image center. **SIFT** (h) keypoints follow the interference ripples in the bright areas of the image. **SURF** (j) detector appears to have covered the image the most uniformly.

**BRISK** was selected from the first four since it ignores the area around the edges, where there is very minimal change. **MSER** and **Star** were selected for their keypoints spread and higher feature quality criteria. And the last detector was the **SURF** algorithm for its uniform keypoint identification.

- (a) Agast
- (b) BRISK
- (c) FAST
- (d) GFTT
Figure 5.14: Keypoints detectors
5.6 BoV Experiment #2: Baseline

5.6.1 Overview

SURF descriptions of the keypoints detected by the previously selected four feature detection algorithms were used to train the BoV model and apply linear regression to estimate the wavelength values.

<table>
<thead>
<tr>
<th>Preprocessing function</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature vector size</td>
<td>5 to 50 with increment interval of 5, and 100</td>
</tr>
<tr>
<td>Feature vector expansion function</td>
<td>None</td>
</tr>
<tr>
<td>Regression models</td>
<td>Linear regression</td>
</tr>
</tbody>
</table>

Table 5.4: BoV experiment #2 setup summary

5.6.2 Results

The Figure 5.15 shows that the EVS values were slightly smaller than those of the PCA model, meaning that more values have been estimated with an error. The mean squared error values were larger as well. The MSER and SURF had very similar performance, both performing the best with a 45-dimensional vector. BRISK had the lowest MSE and the highest EVS values with a 50-dimensional vector, while the Star keypoint detector was the most accurate using 30-dimensional vector.

The challenge data estimation was very chaotic for all the models, ultimately meaning that the models were not able to handle the unseen data very well (Figure 5.15).
Figure 5.15: MSE and EVS of the keypoint detectors
Figure 5.15: BoV experiment 2 challenge data
5.7 BoV Experiment #3: Preprocessing

5.7.1 Overview

Optimal feature vector sizes were fixed for each appropriate descriptor and the previously described preprocessing functions were applied to test whether they improved the models accuracy. The sigma interval was chosen smaller for this experiment, since larger smoothing values would make the image features too vague.

<table>
<thead>
<tr>
<th>Preprocessing function</th>
<th>• Contrast stretching</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Histogram equalization</td>
</tr>
<tr>
<td></td>
<td>• Adaptive histogram equalization</td>
</tr>
<tr>
<td></td>
<td>• Local histogram image equalization</td>
</tr>
<tr>
<td></td>
<td>• Gaussian blur with sigma in range [1-3]</td>
</tr>
<tr>
<td>Detectors</td>
<td>• BRISK (50-D FV)</td>
</tr>
<tr>
<td></td>
<td>• MSER (45-D FV)</td>
</tr>
<tr>
<td></td>
<td>• Star (30-D FV)</td>
</tr>
<tr>
<td></td>
<td>• SURF (50-D FV)</td>
</tr>
<tr>
<td>Feature vector expansion function</td>
<td>None</td>
</tr>
<tr>
<td>Regression models</td>
<td>Linear regression</td>
</tr>
</tbody>
</table>

Table 5.5: BoV experiment #3 setup summary

5.7.2 Results

In general, all the models had the best MSE and EVS results with Gaussian blur (sigma value 2) preprocessing, while the other detail enhancement preprocessing functions reduced the accuracy due to emphasized weak features. More robust features were detected by the algorithms after the weak ones were lost through smoothing the image. Despite the improved evaluation metrics results, there were no improvements in challenge data estimations worth discussing. The Table 5.6 summarizes the metrics values calculated for the validation data set using the baseline experiment setups and the ones with Gaussian blur preprocessing.

<table>
<thead>
<tr>
<th>Detector</th>
<th>MSE</th>
<th>EVS</th>
<th>MSE Gaus.</th>
<th>EVS Gaus.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRISK</td>
<td>1.33974177707e-05</td>
<td>0.986226977104</td>
<td>8.8802574784e-06</td>
<td>0.99107477775</td>
</tr>
<tr>
<td>MSER</td>
<td>7.03594874861e-06</td>
<td>0.99240989926</td>
<td>2.4479122653e-06</td>
<td>0.99750778777</td>
</tr>
<tr>
<td>Star</td>
<td>3.3736843688e-05</td>
<td>0.9659853916</td>
<td>2.20092803328e-05</td>
<td>0.97650034173</td>
</tr>
<tr>
<td>SURF</td>
<td>1.52409102691e-05</td>
<td>0.984006540601</td>
<td>7.15337431606e-06</td>
<td>0.992430720332</td>
</tr>
</tbody>
</table>

Table 5.6: BoV experiment #3 results summary
5.8 **BoV Experiment #4: Transformations**

5.8.1 **Overview**

Table 5.7 summarizes the experiment setup. The \( \ln(x) \) function was omitted due to unresolved incompatibilities with the produced feature vectors.

<table>
<thead>
<tr>
<th>Preprocessing function</th>
<th>Gaussian blur with sigma = 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detectors</td>
<td>• BRISK (50-D FV)</td>
</tr>
<tr>
<td></td>
<td>• MSER (45-D FV)</td>
</tr>
<tr>
<td></td>
<td>• Star (30-D FV)</td>
</tr>
<tr>
<td></td>
<td>• SURF (50-D FV)</td>
</tr>
<tr>
<td>Feature vector expansion function</td>
<td>• ( \sqrt{x} )</td>
</tr>
<tr>
<td></td>
<td>• ( \sin(x) )</td>
</tr>
<tr>
<td></td>
<td>• ( \sin(x^2) )</td>
</tr>
<tr>
<td></td>
<td>• ( \cos(x) )</td>
</tr>
<tr>
<td></td>
<td>• ( x^2 )</td>
</tr>
<tr>
<td>Regression models</td>
<td>Linear regression</td>
</tr>
</tbody>
</table>

Table 5.7: BoV experiment #4 setup summary

5.8.2 **BRISK Model**

The model achieved the second best result of all four models in terms of validation data set **MSE**. The result of 8.239035306e-06 was achieved by augmenting the feature vector with the \( x^2 \) data transformation function. However, the challenge data estimations were very unstable, with already observed tendency of problematic estimations in the lower extremum area of the sinusoids.

![Sinusoid #1](image1.png) (e) Sinusoid #1

![Sinusoid #2](image2.png) (f) Sinusoid #2

Figure 5.16: BRISK experiment 4 challenge data
5.8.3 MSER Model

By far the best MSE value of all the four has been achieved by the model (1.77101851694e-06), using the $x^2$ data transformation function. Inspecting the challenge data showed the model tendency to estimate the values in the lower end of the range, not following the sinusoid curve.

![Figure 5.17: MSER experiment 4 challenge data](image)

5.8.4 Star Model

The model showed no improvements with any of the data transformation functions in terms of MSE and EVS, nor in terms of the challenge data estimation.

5.8.5 SURF Model

Unlike the first two, the SURF model performed best when the sin($x^2$) transformation function was applied. The MSE achieved was 9.91479183785e-06. The challenge data estimations were chaotic and did not follow the sinusoid shape.
5.9 BoV Summary

10 feature keypoint detectors have been evaluated. Four of them with different characteristics were selected for the analysis throughout the further experiments. After finding the most optimal feature vector size for each of them, a baseline was set using the SURF feature description algorithm and the histogram BoV approach. Further analysis of preprocessing and feature expansion techniques showed the models being less accurate than the PCA ones and rather unable to handle the unseen data. The challenge estimations did not resemble the expected sinusoid shape and amplitude, and were rather chaotic. As a result of poor models performance, the estimation using all the challenge data set items was deemed to be of any use and, thus, was not carried out.
Chapter 6

Professional, Legal, Ethical and Social Issues

6.1 Professional Issues

The code written as part of the project follows the British Computing Society codes of conduct. Any third party libraries, software or any other products will be used only if permitted by their licence. Any citations or external information are referenced as appropriate.

6.2 Legal Issues

As mentioned earlier, all third party libraries licences will be respected. The developed experiments scripts, as a part of this project, are released under the MIT licence, included below and with the accompanying source code.

The MIT License (MIT)

Copyright (c) 2015 Jonas Galdikas

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the ”Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge,
publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons
to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or
substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-
INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLD-
ERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT
OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.

6.3 Ethical Issues

The project does not come across any ethical issues.

6.4 Social Issues

The project does not come across any social issues.
Chapter 7

Further Work

The project attempted to apply two different approaches to solving the problem. The further research possibilities are very broad and could be split into stages. The first stage could focus mainly on applying feature engineering techniques to manually derive a stable global feature template using a variety of measurements performed over multiple sessions using the same hardware setup to take into account any non-obvious change in data creation environment. Further stage could then apply an in-depth statistical analysis based on the engineered feature vectors and research the most appropriate estimation models for different applications of the spectrometer. If the device is meant to hold the laser wavelength at a fixed value, a more fine-grain change detection in a small range is required, while the wavelength value measurement can have a larger error.

In terms of the approaches presented in the report, the experiments could be repeated on training and validation data that would encompass data from multiple recording sessions, allowing to capture more variety. Fisher Vectors and VLAD feature encoding techniques could be applied to test whether more advanced BoV techniques result in a more general model, accurately estimating the unseen data.
Chapter 8

Conclusions and Evaluation

Two different data analysis approaches were applied in order to estimate the laser wavelength value based on image data. The first one used the PCA to project a global image descriptor, while the second one used a histogram BoV approach to encode the local feature descriptors, extracted using advanced computer vision algorithms, into a global feature vector, describing the contents of an image.

The analysis was complicated due to some uncertainties regarding the differences between the truth data (training and validation data sets with known values) and the challenge that appeared when performing the experiments. Having in mind that the same hardware setup was used to produce both sets of data, an assumption was made that any variation that might have occurred, be it a light leak or a dust particle on the camera sensor, the model should be prone to such variations. Thus, additional techniques have been explored to developing a more generalized model, able to handle some level of data variation.

The experimental analysis showed that under the same conditions, it is possible to achieve high accuracy of laser wavelength estimation using PCA, image preprocessing and feature engineering techniques. The BoV approach used did not prove to be superior to the aforementioned model, which suggests that the features, extracted from the images of the test data sets by the local feature analysis algorithms, did not have enough distinctivity to take advantage of the used BoV approach.

The personal outcomes of this project include developed understanding of the various image analysis algorithms, trending computer vision techniques, statistics and machine
learning. The research skills have been improved. The extended background information research on global feature extraction techniques and experiment prototyping stages left very little time for implementing more thorough analysis tools for better evaluation.

The experiments developed to test the hypotheses have plenty of space for improvements. A more in-depth statistical analysis of the produced feature vector data could possibly improve the quality of the predictive models. Also, there are other image analysis techniques that could be added.
References


References


