NTNU

Innovation and Creativity

An Application of Image Processing Techniques for Enhancement and Segmentation of Bruises in Hyperspectral Images

Henrik Mogens Gundersen
Bjørn Fossan Rasmussen

Master of Science in Computer Science

Submission date:	June 2007
Supervisor:	Torbjørn Skramstad, IDI
Co-supervisor:	Lise Lyngsnes Randeberg, IET

Norwegian University of Science and Technology
Department of Computer and Information Science

Problem Description

This master's thesis is interdisciplinary in nature and seeks to use known image processing techniques on a new problem within the biomedical optics field of research. The thesis seeks to determine if the image algorithms difference, ratio, and principal component analysis (PCA) can be applied to enhance bruise visibility in hyperspectral images for visual inspection and image segmentation. Known spectral characteristics form the experimentation basis in addition to identification through visual inspection. Furthermore, this thesis seeks to segment out the bruises using the results from the enhancement algorithms using K-means clustering and the watershed transform. To this end, a series of experiments were conducted to answer the following two questions:

- Can the difference, ratio, and principal component analysis algorithms be used to enhance bruises on human skin for visual analysis?
- Can the difference, ratio, and principal component analysis algorithms provide improvements of bruise segmentation compared to segmentations on the original images?

Assignment given: 22. January 2007
Supervisor: Torbjørn Skramstad, IDI

Abstract

Hyperspectral images contain vast amounts of data which can provide crucial information to applications within a variety of scientific fields. Increasingly powerful computer hardware has made it possible to efficiently treat and process hyperspectral images. This thesis is interdisciplinary and focuses on applying known image processing algorithms to a new problem domain, involving bruises on human skin in hyperspectral images. Currently, no research regarding image detection of bruises on human skin have been uncovered. However, several articles have been written on hyperspectral bruise detection on fruits and vegetables. Ratio, difference and principal component analysis (PCA) were commonly applied enhancement algorithms within this field. The three algorithms, in addition to K-means clustering and the watershed segmentation algorithm, have been implemented and tested through a batch application developed in C# and MATLAB.

The thesis seeks to determine if the enhancement algorithms can be applied to improve bruise visibility in hyperspectral images for visual inspection. In addition, it also seeks to answer if the enhancements provide a better segmentation basis. Known spectral characteristics form the experimentation basis in addition to identification through visual inspection. To this end, a series of experiments were conducted.

The tested algorithms provided a better description of the bruises, the extent of the bruising, and the severity of damage. However, the algorithms tested are not considered robust for consistency of results. It is therefore recommended that the image acquisition setup is standardised for all future hyperspectral images. A larger, more varied data set would increase the statistical power of the results, and improve test conclusion validity.

Results indicate that the ratio, difference, and principal component analysis (PCA) algorithms can enhance bruise visibility for visual analysis. However, images that contained weakly visible bruises did not show significant improvements in bruise visibility. Non-visible bruises were not made visible using the enhancement algorithms.

Results from the enhancement algorithms were segmented and compared to segmentations of the original reflectance images. The enhancement algorithms provided results that gave more accurate bruise regions using K-means clustering and the watershed segmentation. Both segmentation algorithms gave the overall best results using principal components as input. Watershed provided less accurate segmentations of the input from the difference and ratio algorithms.

Preface

This master's thesis has been written by Henrik Mogens Gundersen and Bjørn Fossan Rasmussen as part of the graduate course *TDT4900 Master's Thesis* at the *Norwegian University of Science and Technology (NTNU), Department of Computer and Information Science (IDI)*, during the spring of 2007.

We would like to thank our supervisors Torbjørn Skramstad (Professor) and Lise Lyngsnes Randeberg (Ph.D, Department of Electronics and Telecommunications) for their support, expert knowledge, and constructive criticism throughout the process. Without their help, this thesis could not have been written. A special thank is given to Lise. Thanks for letting us play with expensive equipment and for providing much needed image data. Also, thanks for giving us an unhealthy interest for human bruises that will last for life.

Few fellow students were bruised during this thesis. We would like to thank those bruised students that allowed us to record the results of their mishaps with hyperspectral imaging technology.

Finally, we would like to thank Bjørn Kåre Alsberg (Professor, Department of Chemistry) for contributing with ideas and expert advice.

Trondheim, June 2007

Henrik Mogens Gundersen　　　Bjørn Fossan Rasmussen

Contents

List of Tables

List of Figures

Listings

Chapter 1

Introduction

This chapter presents the motivation for conduction this thesis and a summary of previous work carried out during the autumn of 2006. The problem description with specified goals is then presented, followed by the scope of the project. The chapter concludes with outlines for the report.

1.1 Motivation

Hyperspectral images contain vast amounts of data which can provide important information to applications within a variety of scientific fields. The data size is a result of high resolution both spatially and spectrally. Development of image scanner technology is often faster than computer technology, introducing problems related to computation time and computer memory [35]. However, increasingly powerful computer hardware has made it possible to efficiently treat more of this information with respect to both time and storage. This increased usability has led to a number of new applications. These range from mine detection [5] and earth observation [17] to forensic science [16]. This thesis focuses on image processing related to forensic science, more specifically, bruises on human skin. To start things off, a fictional scenario is presented:

A murder has been committed. Police arrive at the scene and finds a naked lifeless individual on the ground in a dark alley. Forensic experts are called to the crime scene to document and collect evidence. They bring with them a newly developed piece of equipment which looks like a regular camera. Images of the body along with the surrounding areas are captured. The data is then processed on a laptop at the scene. The images reveal several important facts to the investigators. First, bruises on the body are determined to be one hour old. Secondly, deeper bruises not visible to the human eye are detected. Comparing the shape of bruises and the depth of bleeding with a database, possible weapons are suggested to the investigators. Thirdly, a small amount of biological material is detected on the wall next to the victim in one of the overview images. A DNA sample can then be taken from the spot which otherwise might have been missed.

This fictional scenario is based on ideas and research concerning hyperspectral image technology related to forensic science. Although the proposed use is fictional, some of the features have already been researched extensively. Firstly, in Randeberg et al. [26, 27] spectroscopic information is used to determine the age of bruises. Hyperspectral images can combine this feature with spatial information to detect and describe the shape of bruises. Secondly, objects that caused bruises might be identified based on another important discovery: A white spot was observed in the central zone of bruises. The shape of this white spot seemed to resemble the shape of the object hitting the skin. Currently, no research regarding image detection of bruises on human skin have been uncovered. With this in mind different fields of research were explored in previous work as described in Section 1.2.

Finally, the identification of biological material elsewhere in the crime scene is an example for further work. To solve this, working with uneven lighting conditions in non-laboratory environments are among the challenges that must be dealt with.

The fictional scenario presented above gives an example of envisaged use, but much research remains before such a robust and accurate device can be used in the field. The motivation behind this thesis is to contribute towards the ultimate goal of a hyperspectral imaging application for law-enforcement and forensic science. Two areas that will take research one step closer to this application have been covered in this thesis. These involve enhancement of bruises on human skin for visual analysis and segmentation of bruises. The end result can be used as a stepping stone towards a future where criminals have yet another obstacle to dodge.

1.2 Summary of Previous Work

This thesis follows from a research project conducted by the authors during the autumn of 2006. Since the autumn project is unpublished, key points related to this thesis is briefly summarised here.

Part of the work was to investigate and identify methods that had previously been used on bruises on human skin. The prestudy phase uncovered that no such work had been done, so the search was expanded to include bruise related research from other fields. Several papers have been published on bruise detection on fruits and vegetables with the aim to improve food quality and safety [2, 39, 15, 14, 20, 19]. It was decided to identify some commonly used methods applied in this regard and single these out for use in this thesis. Ratio, difference and principal component analysis (PCA) were chosen based on their inclusion in several articles that showed promising results. The parameters for the selected methods were mostly chosen based on visual inspection or previous experience. Clustering was suggested as a possible algorithm for segmentation of bruises in future work.

1.3 Problem Description

This master's thesis is interdisciplinary in nature and seeks to use known image processing techniques on a new problem within the biomedical optics field of research. The intent of this thesis is therefore not to present new image processing algorithms. The thesis seeks to determine if the image algorithms difference, ratio, and principal component analysis (PCA) can be applied to enhance bruise visibility in hyperspectral images for visual inspection and image segmentation. Known spectral characteristics form the experimentation basis in addition to identification through visual inspection.

Furthermore, this thesis seeks to segment out the bruises using the results from the enhancement algorithms using K-means clustering and the watershed transform. To this end, a series of experiments were conducted to answer the following two questions:

- Can the difference, ratio, and principal component analysis algorithms be used to enhance bruises on human skin for visual analysis?
- Can the difference, ratio, and principal component analysis algorithms provide improvements of bruise segmentation compared to segmentations on the original images?

To answer the two questions, a software program was developed. This program was designed to run the presented algorithms in batches. The implemented algorithms were experimented with using this program. The enhancement algorithms were chosen based on previous work, as was K-means clustering. The watershed segmentation algorithm was included as it is considered to produce more stable segmentation results and continuous regions, compared to segmentations based on detection of discontinuities, thresholding, and region processing [7].

1.4 Scope

The work performed in this thesis is interdisciplinary. Fields covered include biology, biomedical optics, and image processing. The focus is on the application of known image processing procedures on a new problem related to bruises on human skin. The intent is to contribute to research in biomedical optics, and not image processing. No new image processing approaches have been developed. Instead, known algorithms successfully applied within related areas of research have been transferred to bruise detection on human skin. Because of the interdisciplinary nature of this research, it was necessary for the authors to review basic background knowledge of the biomedical optics field. However, no spectral analysis was performed. Compound identification etc. is not considered within the thesis problem area.

The experimentation done in this thesis was aimed at testing a wide range of parameters on the selected algorithms. The goal was not to present a depth study of the methods, but to examine possibilities for bruise enhancement using the proposed algorithms.

Image restoration is not part of the scope. Images that contained visible flaws have been commented, but no further steps have been taken. Images that contained flaws which might influence results have been evaluated before they were either accepted or rejected as part of the data set. Because of factors beyond the control of this project, parts of the available data set were rejected. Details on these factors can be found in Section 3.1.3.

1.5 Report Outline

The outline for this report is as follows:

- **Chapter 1** gives an introduction to the thesis, a summary of previous work, and presents the problem description
- **Chapter 2** presents background and theory covering an overview of hyperspectral imaging, biomedical optics background material including a brief introduction to bruises, and finally algorithms for hyperspectral image processing
- **Chapter 3** gives a description of cameras used, the available data set, and tools for data viewing and programming. The chapter also presents other programming oriented aspects, such as adjustment of algorithms for implementation and an overview of the algorithm batch software that was developed
- **Chapter 4** presents the tests that were performed and the approach that was followed during testing
- **Chapter 5** presents the results from the tests that were conducted during experimentation
- **Chapter 6** gives a discussion on the results and validity of the results presented in the previous chapter. Each test is discussed individually, followed by a discussion on the overall results, including results not directly related to the problem description
- **Chapter 7** presents a conclusion based on the discussion, and thoughts on further work and research
- **Appendices** include tables, figures, source code etc. that provides additional information to topics covered in this report

Chapter 2

Theory

This chapter presents the theoretical background used in the implementation and experiments described in this report. A brief explanation of hyperspectral data and hyperspectral image processing compared to grey scale images is also covered. Further, a look into biomedical optics presents some issues regarding preservation of image cube dimensionality as well as crosstalk and scattering related to human biology. Basic knowledge of bruises has been included to give the reader some understanding of what transpires during the creation of a bruise, and the materials that influence its colouring. Finally the algorithms used for experimentation are presented.

2.1 Hyperspectral Data and Image Processing

Hyperspectral imaging (HSI) is the acquisition of spectroscopy data for each position in a spatially contiguous image [31, 29]. A typical hyperspectral remote sensing system measures hundreds of wavelengths. It is, however, not the number of measured wavelengths that define a sensor system as hyperspectral, but rather the contiguous narrow sampling of data. Sensors measuring wide and contiguous, or non-contiguous wavelengths are not considered hyperspectral. A normal digital RGB (Red, Green, and Blue) camera uses only three specific wavelength ranges from the red, green, and blue parts of the visible spectrum and is an example of a non-hyperspectral sensor system. This difference is illustrated in Figure 2.1, where the white lines signify a spectral band in the visible spectrum.

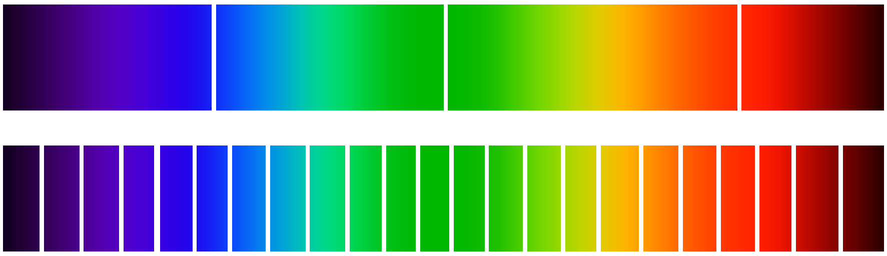


Figure 2.1: *The top spectrum shows an example of bands (white lines) that could be captured by a RGB camera, while the lower spectrum gives the same for a possible hyperspectral camera. The infrared part has been excluded for brevity. Image modified from original by Deborah S. Krolls [11]*

Hyperspectral data can be viewed as a *data cube* or as an *image cube*. Figure 2.2 shows an example of an image cube and the spectral information, often referred to as a spectral vector, which can be found at each spatial position. A band, or layer, in the image cube can be described as a monochrome (grey scale) image. The white lines seen in Figure 2.1 each represent such a monochrome image. Each band spans a two-dimensional grid where each cell is known as a pixel. Let $I(x, y)$ be one of these images, where I is a monochromatic image and x and y are the spatial pixel positions in the image. A hyperspectral image cube can be written as $H(I, \lambda)$, where H is the image cube and λ gives the spectral dimension. Alternatively it can be written as $H(x, y, \lambda)$. By keeping λ constant while iterating through x_n and y_m as shown in Expression 2.1 you address spatial information. The spectral information stored at each pixel position can be read by keeping x and y constant while alternating λ as shown in Expression 2.2. Spectral vectors of n bands are n-dimensional.

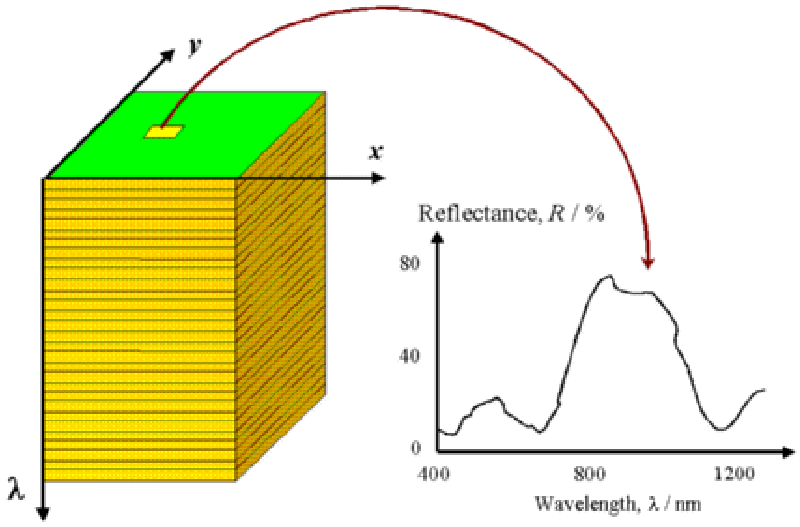


Figure 2.2: *The image cube is shown on the left, where x and y gives the spatial information and λ gives the spectral information. The graph on the right gives an example of the spectral information that can be found in a single pixel. Image from Center for Remote Imaging, Sensing & Processing [13]*

$$H(x_n, y_m, \lambda) \qquad n = 0, 1, ..., Width - 1 \qquad m = 0, 1, ..., Height - 1 \tag{2.1}$$

$$H(x, y, \lambda_k) \qquad k = 0, 1, ..., Depth - 1 \tag{2.2}$$

There are some important differences between hyperspectral image processing and what might be referred to as 'traditional' image processing. In 'traditional' image processing it is quite common to work on single monochrome images, and this also works in image cubes by selecting individual bands for processing. However, if this is done, important spectral information might be lost. This is because pixel values in adjacent layers in the image cube might correlate based on that they represent a sample of one or more specific materials that exist in a pixel position. This correlation can be used as a basis for identification of materials in hyperspectral images.

2.2 Biomedical Background

This section gives a brief introduction to some of the biomedical background material that is needed to understand the problem domain. First, an explanation for why it is necessary to use all the information provided by a hyperspectral imaging system is given. Secondly, a basic understanding of how a bruise is created and how the body reacts to the injury is briefly covered. Since the reader might be unfamiliar with this field, an overview of some of the medical expressions and substances mentioned in this report can be found in Table 2.1. After the bruise introduction, some of the challenges related to human skin and the use of light absorption for identification of different compounds found there is briefly described. Solving these challenges has not been the focus of this project, but the reader should be aware of their existence.

2.2.1 Preservation of Image Cube Dimensionality

The research involving hyperspectral imaging in medical applications is at an early phase. Examples of the work that has been done are the publications of Randeberg et al. [26, 27] relating to age determination of bruises. Compared to the extensive data collections available in i.e. geology through the AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) project [23], medical standards are virtually nonexistent. Presently it is not known what kind of information is needed, and as such all bands of the hyperspectral images might contain information that is necessary to for example divide skin into bruised and non-bruised regions. As research progresses and application specific wavelengths for required biological materials become better known, a reduction of the required number of captured bands might be possible. This would allow cheaper cameras and images with fewer bands. Images with fewer bands they would require less memory and storage space. This again would improve processing time.

2.2.2 Bruises

Bruises are typically caused by an impact that rupture blood vessels and cause haemorrhages underneath the skin without creating a wound [26]. After the impact, bleeding can continue for some time, and the bruise usually develops within 48 hours [12]. Blood outside the cardiovascular system is recognised as a foreign object by the immune system and this causes it to respond with an inflammatory reaction. Red blood cells and free haemoglobin molecules are engulfed and the heme oxygenase system starts to break down the haemoglobin and produce bilirubin and haemosiderin. Presence of bilirubin will give the bruise a yellow appearance [12]. The colouration of a bruise also depends on other factors, in addition to metheamoglobin, bilirubin and other chromophores. These factors include haemorrhage size, the age of a bruise, light

Table 2.1: *Important medical expressions and substances used in this report are listed in alphabetical order. Characteristic absorptions peaks are given for materials of relevance.*

Expression	**Description**	**Absorption peak(s)(nm)**
Bilirubin	A breakdown product of haemoglobin which gives bruises a yellow colour. The absorption peak for bilirubin is between 460 nm and 480 nm depending on the material it is located in. In liquid it can be found at 460 nm. It is translated 20 nm in dermis, to 480 nm [27]	460(480)
Chromophore	Part of a molecule that is responsible for colour	-
Deoxyheamoglobin	Heamoglobin without oxygen (Hb)	555, 760
Heme oxygenase	An enzyme that catalyzes the degradation of heme	-
Haemosiderin	An abnormal pigment found in the human body	-
Melanin	The pigment mainly responsible for coloration of human skin	No peaks
Metheamoglobin	Heamoglobin were the normal Fe^{2+} is in the Fe^{3+} state. This renders it incapable of carrying oxygen. It has a chocolate brown colour.	508, 630
Oxyheamoglobin	Heamoglobin with oxygen ($Hb0_2$)	542, 576

scattering in tissue, and skin properties like the amount of melanin. The depth of the bleeding is also a factor and according to Bohnert et al. [3] there is a distinct difference between the colouration of shallow and deep injuries. Deep injuries were found to be bluish, while shallow bruises appeared bright red. However, bruises that originate in the deeper subcutaneous fatty tissue can remain invisible to human observers or appear only as swellings. Oxyheamoglobin, deoxyheamoglobin, and melanin are the most important chromophores in human skin [26]. The different chromophores are identified through characteristic peaks in the reflectance spectra, which originate from light absorption in skin.

2.2.3 Melanin-Haemoglobin Crosstalk and Scattering

Crosstalk can be described as a mixing of separate signals which makes them difficult to identify individually. In human skin, this can occur between melanin and deoxyheamoglobin with respect to appearance of pigmentation. It has been pointed out that the perceived skin darkening does not depend only on the concentration of melanin, but that it is strongly affected by the concentration of deoxyheamoglobin [34]. This crosstalk effect is expected to make it difficult to visually separate the two contributing parts. The main reason for this is that the absorption spectrum of deoxyheamoglobin in the 630 - 700 nm range is very similar to the absorption spectrum of epidermal melanin, so whatever light is transmitted has the colour of pigment. Blood pooling (stasis) alone can also be confused with pigmentation.

Scattering in human skin is mainly attributed to collagen fibres [34]. Scattering typically forces photons to deviate from an otherwise straight trajectory. Path length dispersion might influence results since a change in path length changes the molecularly-specific absorption photons are subjected to. This makes it more difficult to know where photons observed by a camera originated from within the skin. In other words, identification of materials might be influenced by scattering.

2.3 Algorithms

This section presents the theoretical background for the image processing algorithms used in the implementation and experiments described in this report. Changes and modifications needed to make these algorithms work is covered in Section 3.3.

There are two purposes for performing feature enhancement on the hyperspectral images in this thesis. The first is to enhance transitions present in the image for easier visual inspection. The second is to provide a good basis for segmentation by using the enhanced image. The resulting segmentation can then be used to limit the data to a region of interest (ROI), for example a bruise. In addition to spatially locating desirable image features, the amount of spectral analysis and calculation is then reduced. The spectral analysis (5) has not been a part of this project, but is included to show what the reduced images can be used for. Figure 2.3 shows the data flow as described.

Segmentation is defined as a process that given an image, possibly a multi- or hyperspectral image, consideres every image point and assigns it a label that indi-

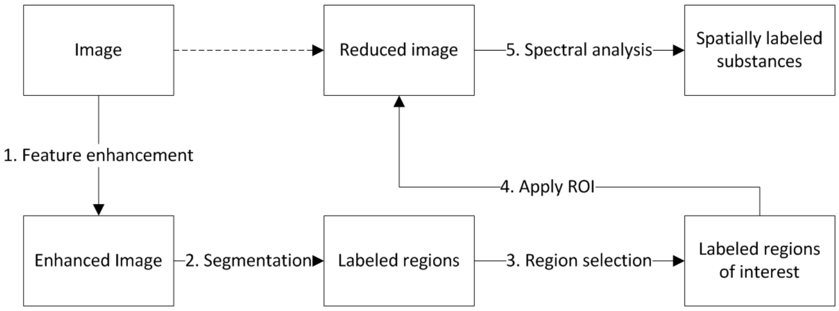


Figure 2.3: *The image is enhanced (1) for segmentation. Resulting regions from segmentation (2) are analysed and regions of interest (ROI) are selected (3). The ROI are applied to the original image (4). The dotted line indicates the connection with the original data when using the region of interest on it. Analysis is performed within the reduced region (5), resulting in a spatial substance mapping.*

cates its relation to a group that share some common property. The result of the segmentation can be called a label image or label map. The segmentation algorithms are presented in Sections 2.3.3 and 2.3.4.

Watersheds and clustering used as segmentation algorithms differ somewhat, as watershed segmentation primarily is used on grey scale images, and clustering on multi layered (banded) images. In a sense, watershed is a two dimensional, or spatial, segmentation algorithm, while clustering is a multi dimensional segmentation algorithm.

2.3.1 Ratio and Difference

The dual-band algorithms are considered simple and fairly straightforward, so the explanation will be kept short. The ratio or difference of an image is typically used to emphasise certain aspects in two images taken from the same area. Difference can be used to highlight regions of change, while the ratio between two bands can be used to enhance subtle differences in spectral characteristics [7, 29]. The formula for difference can be written as:

$$G_{diff}(i,j) = G_1(i,j) - G_2(i,j) \tag{2.3}$$

where G_{diff} is the resulting image, G_1 and G_2 are the two input images, while i and j are spatial pixel position within each image. Care should be taken to retain the dynamic range in the result image, for example by avoiding negative pixel values as a result of the difference operation. Details on this subject can be found in Section 3.3.1.

Rationing is the process of dividing pixel values in one image by the corresponding pixel values in a second image. This can be expressed as:

$$G_{ratio} = \frac{G_1(i,j)}{G_2(i,j)} \tag{2.4}$$

where G_{ratio} is the result image. Rationing is a non-linear transformation and images that have been transformed cannot be reconstructed back to the original [29].

2.3.2 Principal Component Analysis

A multivariate image is a three way array in which the first two dimensions represent a spatial plane, and the third dimension is a variable span, like for example temperature, mass, or some other physical or latent property. A hyperspectral image is a multivariate image in which the third dimension represents different wavelengths.

An important tool for statistical study of multivariate images is the principal component analysis (PCA) algorithm. PCA is a statistical method often used to preserve essential information with reduced dimensionality (compression), or to emphasise features not discernable in the original data. These properties are beneficial when working with hyperspectral images, because of the many bands in the image cube. Adjacent bands in the image cube are generally correlated. This correlation implies redundancy, since some of the information is repeated [17]. A linear operation, like PCA, can be applied to give an alternative description of the data, with the correlation described differently and with less redundancy. This is the principle of PCA based compression.

The multidimensional nature of hyperspectral images can be represented by a vector space with as many dimensions as there are spectral bands in each pixel position [29]. One set of images in the available data set has 160 bands where each band has a resolution of 500 x 600. This gives a total of 300 000 vectors, each with 160 elements, corresponding to 160 dimensions. These vectors are used to compute the covariance matrix and mean vector. A two dimensional example is presented next, as the principle applies to any dimensionality and illustration becomes difficult at higher dimensions.

The first step in PCA is to find the orthogonal vectors that give the best description of the data set. These vectors are called eigenvectors, and there is one per dimension. Each eigenvector has a corresponding eigenvalue that represents how well it fits a data set. A visual explanation of the concepts can be seen in Figures 2.4 and 2.5. Figure 2.4 shows two eigenvectors for a two dimensional data set.

The first (red colour, the longest) eigenvector points in the direction of the greatest variance in the data, the second (green colour, the shortest) points in the direction of the second greatest variance, and so on. The step after finding the eigenvectors is to reduce the number of dimensions. It is usually sufficient to represent a data set with 1-10 % of the eigenvectors, since most of the variance, or information, is contained in the first couple of principal components. Figure 2.5 shows how this is done, by projecting the input vectors onto an eigenvector. All the data is now along one axis, which means that the dimensionality has been reduced from two to one. The alignment of data along the eigenvectors is the mechanism that de-correlates the

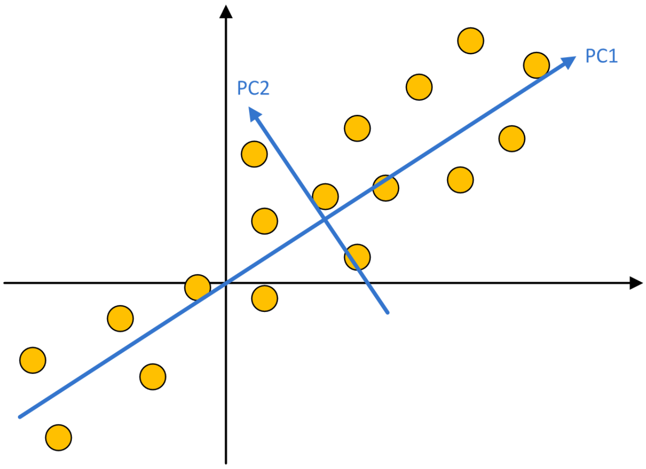


Figure 2.4: *Principal Component 1 (marked PC1, the longest) represents the eigenvector pointing in the direction of greatest variance. Principal Component 2 (marked PC2, the shorter) points in the direction of second greatest variance. The data is plotted as circles.*

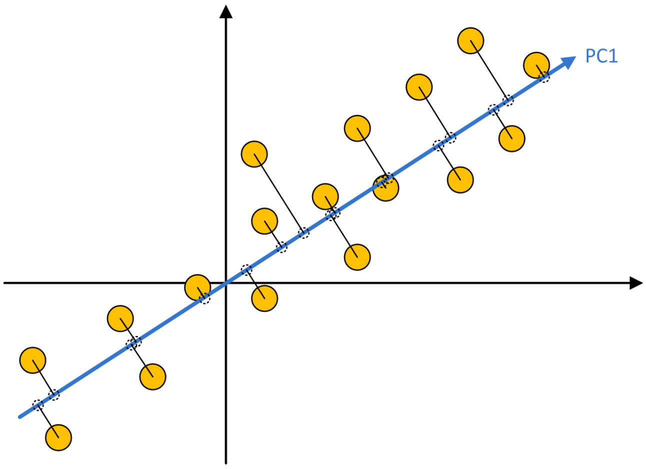


Figure 2.5: *The input vector data is projected orthogonally onto the eigenvector PC1.*

data set [7]. A consequence of this is that an RGB picture which is composed of three principal components will always contain more information than any combination of three spectral bands [17].

Listing 2.1 briefly presents the steps in the PCA algorithm [6]. The steps are not a recipe for implementation, but a high level presentation of the algorithm for easier understanding.

Listing 2.1: PCA Steps

```
1. Reorganize the data. The pixels in the N x M x P image must be
   organised in a single row of vectors. Each pixel is termed an
   OBSERVATION and each pixel vector consists of VARIABLES. The
   result is a MNxP matrix.

2. Subtract the mean from the data. The mean is the average across
    each dimension. This produces a dataset whose values are
   centred on origo (zero mean).

3. Calculate the covariance matrix for the data.

4. Calculate the unit eigenvectors and eigenvalues of the
   covariance matrix.

5. Sort the eigenvectors based from high to low eigenvalue. The
   eigenvector with the highest eigenvalue is the first principal
   component, the second highest is the second principal component
   , and so on. The matrix E of eigenvectors is formed.

6. The input data F is transposed with the matrix E of
   eigenvectors, resulting in the principal component image P.
```

A different approach to calculate the principal components is the Non-linear Iterative Partial Least Squares (NIPALS) algorithm [38]. The NIPALS algorithm does not calculate all the principal components at once. Instead it calculates the first score (t_1) and the first loading (p_1^T) from the image data X. These two components represent the first principal component. To calculate the following components, the residual data E is the reduction of $t_1 p_1^T$ from X. The residual is used to calculate t_2 and p_2^T. The process is repeated for each component.

Principal component analysis seeks to find the eigenvectors that lie along the maximum variance in the data set X. Partial least squares (PLS) regression and partial least squares-discriminant analysis (PLS-DA) seeks to find and use a model of the maximum covariance between the data set X and the model Y. A method developed and used for statistical purposes, PLS regression combines and generalises features of PCA and multiple linear regression. The PLS components can like PCA be found using NIPALS [1, 38, 6].

2.3.3 K-means Clustering

Unsupervised classification, or learning, is a term for grouping objects with similar properties together, without any foreknowledge of those properties. As stated in literature [32, 29, 4, 35], most often clustering algorithms are used for unsupervised classification.

Clustering can be used for classification on multivariate images. The image clustering result is an assignment of each spatial position to a spectral class based on the values of the different points in the image bands. The principle is illustrated in Figure 2.6. The results of clustering can be used to determine the location and number of classes present. A supervised connection can later be applied to the results with available spectral reference data. Clustering algorithms are typically computationally expensive, but are essential to the analysis of multi- and hyperspectral data [35, 32].

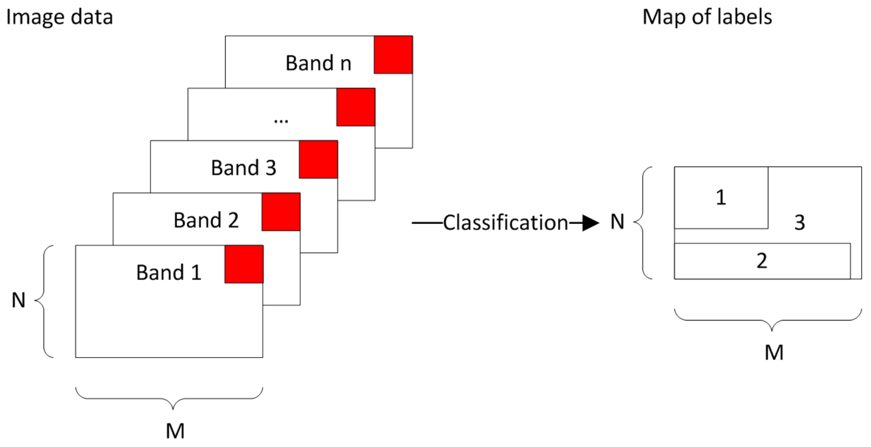


Figure 2.6: *The figure illustrates the principle of classification. Each pixel vector is called an object. The values of the vector correspond to a physical property such as wavelength. The resulting label map is shown with three labels as an example of a possible result.*

Multivariate image bands can be many different kinds of physical and latent properties. In literature, temperature, mass, wavelength, and principal components are mentioned as examples [35, 32]. An important difference in clustering on multivariate images compared to non-image data, is the extra spatial information available. Spatial information can improve results, but in most cases it is ignored [35].

The K-means clustering algorithm is a widely used deterministic clustering method. It clusters the given data in K clusters based on the characteristics of each observation. Each object (pixel) can only belong to a single cluster, and such clusterings are called hard. The number of clusters can be difficult to determine automatically, and can only be found manually with a trial and error approach [35].

The algorithm starts out by assigning K random centroids and assign each observation to the closest centroid. The distance is most commonly measured by the

Euclidean distance function $\epsilon\delta$ [10]. Even with random starting centroids, the method is deterministic, and will result in the same clustering if run until convergence. After the initial run, new centroids are calculated by finding the mean of all objects in the cluster. The algorithm is iterated with new mean centroids until the total intra-cluster distance sum is minimized. The iterations are commonly limited by convergence (when objects no longer switch clusters), or by setting a limited number of iterations. Equation 2.5 shows the typical minimization equation:

$$S = \sum_{i=1}^{k} \sum_{j=1}^{NM} \epsilon\delta(x_j - \mu_i) \tag{2.5}$$

where S is the resulting distance sum, k is the number of clusters, NM the number of objects (pixels), x_j the j-th object, $\epsilon\delta$ the distance function, and μ_i the i-th cluster centroid.

The Euclidean distance function can be replaced with the simpler square error function for a simplification of the equation [35]. By not performing the square root operation of the Euclidean distance, computation time can be saved. This can be done, since if $\sqrt{a} > \sqrt{b}$ then $a > b$.

Listing 2.2 shows a step by step description of the K-means clustering algorithm for a multivariate image.

Listing 2.2: K-means Clustering Steps

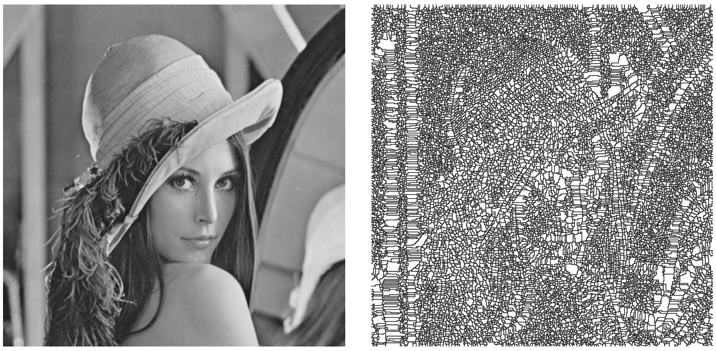
```
1. Place K points into the coordinate space represented by every
    point/object in the image that are being put in clusters. These
     K points are the starting centroids.

2. Assign each image point to the closest centroid.

3. Recalculate the positions of the inital centroid by meaning
    over the points assigned to the clusters.

4. Repeat step 2 and 3 until no points change group, or the number
     of repeats equal I, maximum iterations.
```

There are some theoretical concerns with clustering in literature [35, 32]. There are concerns with using clustering on high resolution images with many multispectral bands as the complexity and run time increases greatly. If run time can be ignored, there are still problems regarding noise. K-means clustering is sensitive to noise and many imaging devices produce outliers and noise due to limited sensor sensitivity, statistical variations, or signal interferences such as magnetic fields from electrical equipment. A commonly used solution is signal smoothing [7] which can eliminate some noise problems, but smoothing will reduce accuracy as pixels are combined with neighbouring values throughout the image.

2.3.4 Watershed Segmentation

In this section the watershed transform is presented. The watershed transform is a widely used and popular method for image segmentation in the field of mathematical morphology [30]. Watersheds often produce more stable results than other segmentation approaches based on detection of discontinuities, thresholding, and region processing [7].

The concept of the watershed transformation can be explained by looking at a real word analogy. Imagine an area of slopes, mountains, and valleys where it is raining evenly over the whole area. A rain drop hitting any point will travel along the greatest gradient towards the nearest local minima, or in some points two or more local minima. The latter would form, at the split point, a topological ridge, or *divide lines*. The objective of the watershed transform is to find these watershed lines. The principle is to allow flooding at a uniform rate through each local minimum from below. When two catchment basins are about to merge, a dam is constructed to prevent the merging. In the end, the flooding will cover the entire landscape, and only the dam tops will be visible above the water. The dam boundaries represent the watershed lines, and the different lakes are the segmented areas.



(a) Grey Scale Input Image

(b) Direct Watershed of Input Image

Figure 2.7: *An example of naive input to the watershed transform. The left image shows a common test image and the right image the results of watershed segmentation.*

Typically, a grey scale image directly input to the watershed transform algorithm would result in a oversegmentation, as there are large amounts of local minima in standard images. The results from the watershed transformation on an unprocessed grey scale image can be seen in Figure 2.7. To counter the oversegmentation, different pre-processing strategies are commonly used. Regions characterised by small grey value variations typically have small gradients, and therefore gradient images are often used as input. Another common approach [30, 20] is to threshold the greyscale input or gradient image, resulting in a black and white mask. That mask is then morphologically opened (an erosion[1]followed by a dilation[2]) with a structural element. A distance map is then calculated for the resulting image. The Euclidean distance from each white (or black) point to the nearest black (or white) point is measured and stored in an image map. The watershed transform is then performed on the image map. This method is considered supervised because of the image thresholding, while the gradient approach is mostly unsupervised.

The watershed segmentation algorithm is defined in Gonzalez et al. [7] by set operations. Briefly summarised, the flooding level will begin below the lowest value and in the discrete case iterate upwards until it is beyond the highest grey value. At any step, the number of points in the catchment basin below the current flooding level is needed. As the levels rise, the number of points in a region will remain the same or increase. When the flooding is complete, each group of points in each flooding level is examined and is classified as one of three types by how it is connected (neighbourhood connection). Class (i) is valid for the group of pixels forming the local minima, as no other points are connected to this group at its value or below. Class (ii) is the group of points that connects only to points in a single catchment basin (or label) at its value or below. Finally class (iii) points are the points that connect both to points in its own group at its value or below and points of another catchment basin (another labelling). Class (iii) is fulfilled for any point that lies along the boundary of a region, the aforementioned dam analogy.

[1] Erosion is one of two basic operators in mathematical morphology. The effect of the operator on a binary image is to erode away the boundaries of regions, where a masking element determines the degree of erosion. [7]

[2] Dilation is one of two basic operator in mathematical morphology. The effect of the operator on a binary image is to gradually add to the boundaries of regions in the image, where a masking element determines how much the boundaries are enlarged. [7]

Chapter 3

Implementation

This chapter presents the implementation details. This includes available data, gathering of new data, usage of existing software, implementation details for algorithms, and application design.

3.1 Materials

This section describes the capture devices and data set available during this thesis, limitations of the data acquisition, and radiance-to-reflectance conversion of the captured data.

3.1.1 Capture Devices

Four different hyperspectral cameras were used to capture the images available to this thesis. All were developed by Norsk Elektro Optikk AS (NEO, Lørenskog, Norway). Details on the four specific cameras can be found in Table 3.1. Figure 3.1 shows the VNIR-1600 hyperspectral camera mounted on a translation device that was used to capture some of the images.

The cameras capture raw radiance images which are then converted to the ENVI binary raster format. The format is a flat binary uncompressed format with a separate header file. The data can be stored, often called interleaved, in three different schemes. These schemes are described in Section C.1.

A variety of different light sources were used during separate recording sessions with different results. Details for the light sources can be found in Table 3.2. LS1 gave the best results, since it did not introduce any visible striping to the captured images. It would have been ideal if LS1 could have been used for all images, since this would eliminate a variable. This could not be done, because of available hardware when the images were captured.

3.1.2 Data Set

The data set was used in previous work mentioned in Section 1.2, but it was extended for this thesis. Details for the complete data set are provided in Section A.1. Three examples from the data set are presented here. Details on the three image groups can be found in Table 3.3. The details include an identifier (id) for the image group, the sex (male (M) or female (F)), and age (in years) of the subject. Comments in

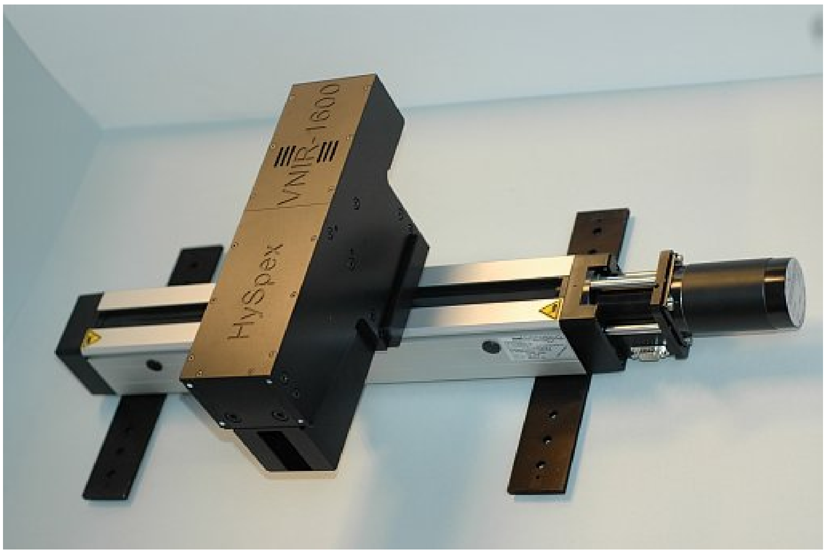


Figure 3.1: *NEO VNIR-1600 Hyperspectral camera mounted on a translation device. Image from NEO [24].*

Table 3.1: *System specifications for NEO hyperspectral cameras. For more information see the NEO webpage [24]. Two similar VNIR-1600 cameras where used, differing slightly in spectral range. This gives different band correspondance to certain wavelengths.*

Module	**VNIR-1600**	**VNIR-640**	**SWIR-320i**
ID	H1a and H1b	H2	H3
Detector	Si CCD 1600*1200	Si CCD 640*480	InGaAs 320*256
Spectral range	0.4-1.0 μm	0.4-1 μm	0.9-1.7 μm
Spatial pixels	1600	640	320
Spectral sampling	3.7 nm	5nm/10nm	5 nm
Spectral bands	160	128/64	170
Frame rate to HD	120 fps	500/850 fps	350 fps

Table 3.2: *Specifications to the different light sources used to capture images.*

ID	Details	Comments
LS1	DC Halogen 300W diffused	No striping
LS2	AC Halogen 2 x 500W	Heavy striping effect
LS3	DC SCHOTT KL 1500-T	Strongest setting used. Striping effect. Lightsource converted from AC to DC.
LS4	AC SCHOTT-FOSTEL LCC	Striping effect

the table give information on what the images contain, where and how old potential injuries are, and which cameras and light sources were used.

Table 3.3: *Details on the three sample images and the image collections they where taken from. Excerpt from Table A.1*

ID	Sex	Age	Comments
I1	M	32	A collection of 17 images taken at times ranging from 66 to 421 hours after injury. Images show two visible bruises on the volar side of the right elbow caused by paintball bullets. Camera:H1a Light:LS1
I2	M	32	A collection of 17 images taken at times ranging from 66 to 421 hours after injury. Images show two visible bruises on the volar side of the right elbow caused by paintball bullets. Camera:H3 Light:LS1
I4	Pig	-	A collection of 16 images taken at times ranging from before injury to about 1 hour after. Images show two bruises caused by paintball bullets. Camera:H2 Light:LS2

Since much of the available data set contained time series observing development of injuries, it was possible to reduce the number of images used for experimentation. To get a more efficient experimentation, it was decided to select images that could represent the data set and retain as much variation as possible. To uniquely identify these images from their collections, each image was assigned a unique identifier (UID). The UIDs, file names, and a brief summary for the example images can be found in Table 3.4, while Section A.2 provides the same for all the selected images. Figure 3.2 shows the three example images. By comparing the list of images available to the project with the list of selected images for experimentation it should be apparent that none of the images from I5 through I14 have been included in the last list. The reasons for this are presented in Section 3.1.3.

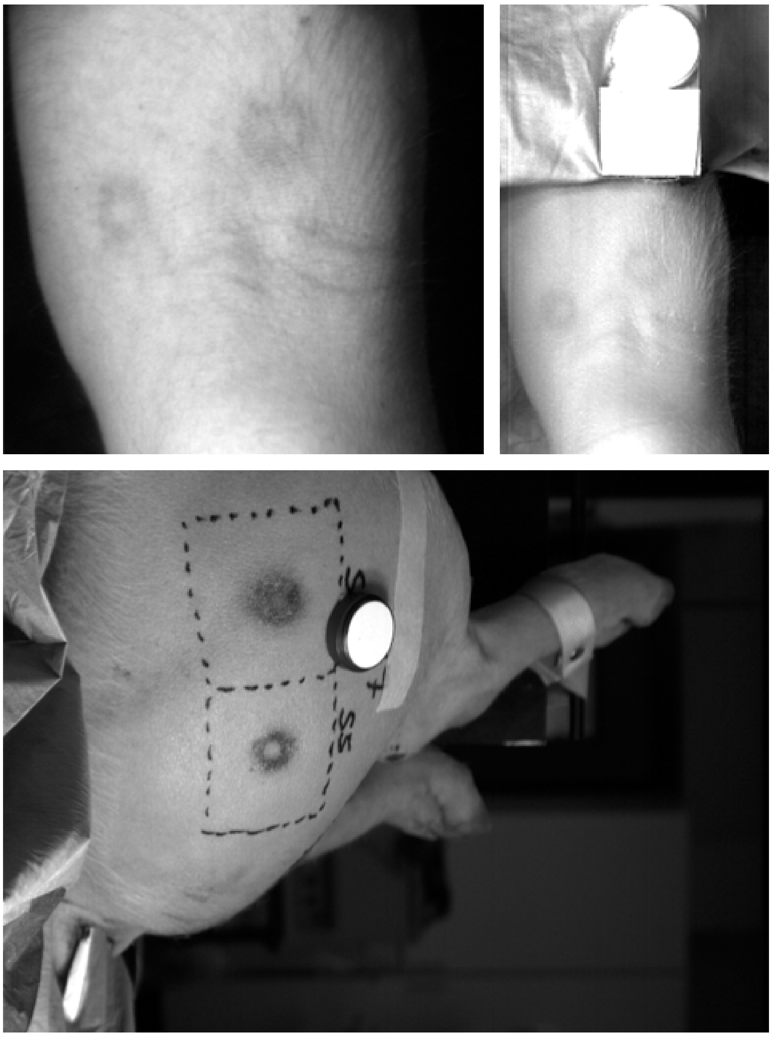


Figure 3.2: *Three sample images from the data set. The upper left is a cropped image (image collection I1, unique id I1-1) at wavelength 573.64nm. The upper right is an uncropped image (image collection I2, unique id I2-1) at wavelength 1362.67nm. This image shows the spectralon tile that was cropped from the upper left image. The lower image is from image collection I4 (unique id I4-3). This image band, from wavelength 854.68nm, shows two bruises in two control zones, spectralon sample tile, and surrounding tissue and setup.*

Table 3.4: *The file name and a brief summary for the sample images. A unique id (UID) is assigned to each image that belongs to an image collection as can be seen in Table A.1. For example, if two images come from image set I1, the first will be labeled I1-1 and the second I1-2. Excerpt from Table A.2*

ID	UID	File name	Comments
I1	I1-1	72h	Visible bruises
I4	I4-3	g7s5s6post1_2006_06_02_ 14_46_40_VNIR_640_SN4_c.img	Visible bruises
I2	I2-1	72h_swir	Visible bruises

3.1.3 Limitations During Data Acquisition

New images were captured for this project but none were used. The images used are all from previous work. There are several reasons why no new images were included and these involve both technical and non-technical issues. If these issues had occurred individually some of the images could have been used, but the combination of all led to their exclusion from experimentation. This section presents the limitations faced when new images were captured.

Availability of the hyperspectral camera system meant that new images could not be captured before late March. The images that were captured during the first session used light source LS3 and LS4 which had inadequate strength and introduced striping effects. Some images captured at a later date used LS1 which provided better lightning. Details on the light sources used for each image can be found in Section A.1.

The high resolution of the new images also became a problem because of limitations in computer hardware and software. When an area of interest was captured, for example the lower part of an arm, the image size was in excess of one gigabyte. Before the images could be used, they had to be cropped considerably. Since the images were captured using a lens with 20 cm focus, the area of skin visible in the cropped image was considered too small to be of interest (approximately 4 cm * 4 cm). A lens using 100 cm focus distance, as was used for images in I1, would show a larger area of skin for the same image size.

An initial visual analysis of the new images uncovered noise and scan line shifts that could not be explained at the time. Two possible answers were later discovered, one related to the equipment and the other to the environment the camera was situated in. The translation unit (horizontal camera scan movement) showed signs of being inaccurate. A firmware fix to solve the problem was delivered by NEO in May. The other possible source was discovered during a session using no light. Noise could clearly be seen in the captured image. Magnetic fields above normal have been measured in the laboratory used for image capture. It is believed that the camera equipment is affected by this interference. This problem was also discovered in May and it was decided that it was too late to include new images at that time.

Another issue observed was difficulties for the subject to remain still throughout the capture time. Movement during acquisition can give apparent scan line problems. It is believed that this could be a big problem if children are involved. Faster image acquisition could reduce this problem.

3.1.4 Radiance-to-Reflectance Conversion

Radiance and spectral radiance are the radiometric measures of light passing through, reflected, or emitted from a surface. Reflectance is the fraction of energy reflected from a material surface, relative to incident energy and its wavelength. Relative to incoming energy, the reflectance value is therefore independent of lighting intensity.

Standard digital, multispectral and hyperspectral sensors are all radiance sensors, covering different ranges of wavelengths. The image radiance function $F(x, y, \lambda)$ can be characterized as the product of the two components illumination $I(x, y, \lambda)$ and reflectance $R(x, y, \lambda)$ [7]. The model is shown in Equations 3.1 and 3.2 [7, 17].

$$Radiance = Reflectance \cdot Illumination \tag{3.1}$$

$$F(x, y, \lambda) = R(x, y, \lambda) * I(x, y, \lambda) \tag{3.2}$$

Reflectance data is available for many substances, for example water, aluminium, chlorophyll, and hemoglobin among many others. For comparison of the image against reflectance data, and to lessen the influence of the light sources, the radiance images are converted to estimated reflectance images. This estimation is based on a Spectralon (Ocean Optics, Duiven, The Netherlands, LabSphere) standard tile present in the image. The illumination is estimated based on this standard, shown in Equation 3.3:

$$I_\epsilon = \frac{F_s}{R_s} \tag{3.3}$$

where F_s is the Spectralon radiance mean, R_s the Spectralon reflectance, and I_ϵ the estimated illumination.

A group of pixels containing Spectralon are selected and a mean Spectralon radiance is found. This is divided with the known reflectance for Spectralon over the same wavelength range. The radiance image is converted to a reflectance image based on this illumination. This is shown in Equation 3.4:

$$R_i = \frac{F_i}{I_\epsilon} \tag{3.4}$$

where R_i is the reflectance image, F_i the radiance image, and I_ϵ the estimated illumination.

Converting radiance to reflectance images requires accurate spectralon radiance data. All the images used during experimentation were reflectance images, except those from data set I2. This is because the images had saturated and corrupt data in the Spectralon tile. After the reflectance conversion, the image corruption was multiplied. The images from I2 that were used during experimentation are therefore radiance images.

3.2 Tools

This section presents the tools and hardware used. This includes software for data viewing and presentation, computer hardware available, and the evaluation and final selection of a framework for algorithm development.

3.2.1 Data Viewer

The data in hyperspectral images can not be visualised directly in colours like standard RGB-images, because of the multitude of wavelengths or layers in the image cube. The data present in hyperspectral image cubes are typically beyond the RGB 8-bit value range, often 16 or 32 bits, and also beyond the display capabilities of standard computer displays. Hyperspectral images are therefore most often viewed as grey scale images or pseudo colour images with the value range compressed to an 8-bit range. Pseudo colour images are simply three bands from the image cube set respectively as the red, green, and blue component.

Data were visually inspected during the experimentation using ENVI 4.2 (ITT Visual Information Solutions, formerly Research Systems Inc.)[9] and MATLAB (The Mathworks Inc.)[18]. ENVI provides tools for visualisation, analysis, and presentation of all types of digital images. MATLAB has in addition to its computational environment several image processing and visualisation tools.

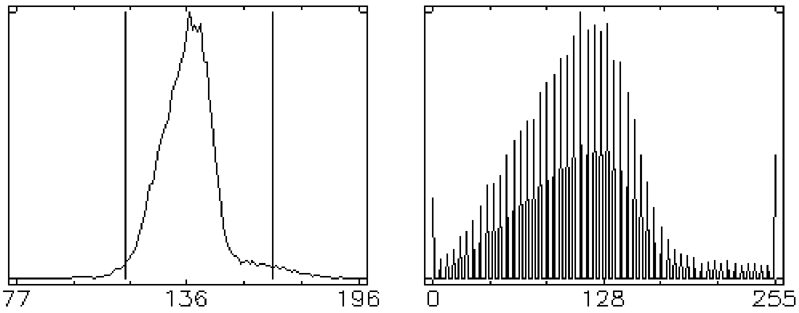


Figure 3.3: *Left: Histogram of input image. Right: Histogram of 2% linear stretched input image.*

ENVI performs a linear stretch of the input image before displaying it, excluding 2% of the pixel extremities. That means, given an image with 100 pixels, the one pixel with lowest and the one pixel with highest intensity is excluded. The new lowest intensity will be interpreted as zero intensity (black) and the new highest as maximal intensity (white). An example of this stretch is shown in Figure 3.3. The left graph shows the histogram of the input grey scale image with two vertical lines indicating the 1% extermity cut on each side. The histogram to the right shows the area between these lines mapped to the range of integers from 0 to 255.

3.2.2 Computer Hardware

For the development and execution of the algorithm code needed for the experimentation, two assigned student computers were used. These two computers differ only in operating system. Both systems are 32-bit Windows systems, with a default usage of 2Gb physical and virtual memory ($2 \cdot 1024^3$ bytes) [21, 22]. Beyond this, applications must utilise the hard drive for swapping. Aside from the operating system, only the processor and the physical memory had a direct impact on limitations set for the experimentation. Because of this, the input data was limited to 100 MB images. A conversion to double precision was needed for several algorithms, expanding these file size by a factor of four. The choice of MATLAB as a framework, see Section 3.2.3, confirmed this limitation, as there is no default hard drive swapping implemented. The relevant details for the two computers are listed in Table 3.5.

Table 3.5: *The specifications of the computers that were used in development and execution of the implemented algorithms.*

	Computer 1	**Computer 2**
Operating System	Microsoft Windows XP SP2 32-bit	Microsoft Windows Vista 32-bit
Physical Memeory	2048MB (2x1024MB)	2048MB (2x1024MB)
Processor	Intel Pentium 4 Prescott (90nm) HT 3.0GHz	Intel Pentium 4 Prescott (90nm) HT 3.0GHz
Primary Storage	Samsung HD080HJ 80Gb 7200RPM hard drive	Samsung HD080HJ 80Gb 7200RPM hard drive

3.2.3 Frameworks

There are two basic approaches for experimental implementation of algorithms. The first is to implement low and high level functionality, reducing the amount of overhead and unnecessary code fragments. Due to the amount of programming, this approach is time consuming and may present other problems such as debugging and difficult general verification of code. The second approach is to use existing, functional, and verified frameworks (often termed libraries) and software for underlying functionality, like vector and matrix math. Basing development on pre-built components will save time and amount of coding needed, but may prove difficult to use because the framework might be complex and difficult to integrate and the documentation can be of bad quality or hard to understand.

For this thesis, the second approach was chosen based on a need for experimentation and less implementation. We chose C# as the programming language, discarding C++ after a short test implementation period. MATLAB was chosen as algorithm framework. C++ compared to C# offers more low-level control, but is also harder to use for simple applications. The usage of MATLAB and C# is considered to have resulted in a more efficient development process.

The most important criteria set for the selection of the development method are shown in Table 3.7. The criteria have been weighted based on their importance. These criteria are then fulfilled (Y), partially fulfilled (P), or not fulfilled (N). For a simple and transparent evaluation, each category has been assigned a number. These values can be found in Table 3.6.

Table 3.6: *The values assigned to different levels of importance and different levels of coverage.*

Category	**Value**
High	3
Medium	2
Low	1
Fulfilled (Y)	1
Partially fulfilled (P)	0.5
Not fulfilled (N)	0

The level of importance is multiplied with the level of cover for each criterion and summed. This method is not purely objective as the fulfilment of each criterion is subjectively evaluated based on each software's documentation and user evaluations. Table 3.8 compares the frameworks and the final ranks.

This evaluation is not an exhaustive study of available libraries or software. Rather it is a selection based on experience with and knowledge of frameworks and programming languages. The following sections present the evaluated libraries with their strengths and weaknesses.

Open Computer Vision Library

Open Computer Vision Library, shortened OpenCV, is a free open source library of programming functions mainly aimed at real time computer vision. Supported by Intel[8], OpenCV can load at runtime the Intel Integrated Performance Primitives (Intel IPP), which optimise OpenCV for running on Intel's processors and chipsets. OpenCV is mainly written in C++, and is interfaced through C++ also. Libraries for integrating OpenCV with C# exist, but are mostly unfinished at the time this thesis was written. The two known wrapper libraries are SharperCV (Rhodes University, South Africa) and OpenCVDotNet (Interdisciplinary Center Herzliya,Israel).

OpenCV can among other things be used for object identification. One example can be found in Ozan and Gümüstekin [25] where OpenCV was used to detect a chess board and to identify inner corners on the board.

OpenCV was evaluated since it contains Principal Component Analysis (PCA). However, the framework does not support the ENVI standard file format without integration of GDAL (Geospatial Data Abstraction Library). Also the functions required to perform PCA currently only supports 8-bit images according to the online

Table 3.7: *Selection criteria for framework software.*

Criterion	**Description**	**Importance**
C1	Framework has functional large scale vector and matrix math.	High (3)
C2	Framework is generally well documented and/or supported.	High (3)
C3	Framework has external dynamic interface for integration through programming language of choice (i.e. Windows DLL).	Medium (2)
C4	Framework has previously implemeted and documented version(s) of planned algorithms.	Medium (2)
C5	Framework components are individually documented and correct.	Medium (2)
C6	Framework has large scientific user base.	Low (1)
C7	Framework has previously implemeted and documented version(s) of data format load and store.	Low (1)
C8	Framework has previously implemeted and documented version(s) of data viewer.	Low (1)
C9	Framework has documented high processing performance.	Low (1)

documentation [33]. The available hyperspectral data set contains only unsigned 16-bit images. These limitations could be solved, for instance by writing a load and write module for the ENVI format and by modifying internal methods. It is believed that the time cost of implementing this would exceed the time available.

The documentation for OpenCV is good for introduction to the framework, and all functions are individually documented. For information on the internals of each function, the source code is referenced. The documentation is however considered of lesser quality than for example the documentation of MATLAB which also shows examples and more detailed information for each function. No technical support is available, but there is a mailing list and an active forum for asking additional questions. A scientific and a commercial user base for OpenCV exists, but primarily it consists of developers who work on personal projects or small projects and need a real-time image processing framework.

Insight Segmentation and Registration Toolkit

The Insight Segmentation and Registration Toolkit, abbreviated ITK, is an open source software system incorporating many mid and high level segmentation and registration algorithms. The focus of ITK is on medical applications, although the toolkit is capable of processing other data types. The ENVI standard file format is not supported, but there is support for implementing and integrating loaders for other data storage formats. As ITK is open source it benefits from a large self-sustaining community, a recognisable feature in good open source software. Simple and quick reporting of issues and a dedicated user base with expert knowledge is a valuable resource for the toolkit.

The core developer team of ITK is a scientific team with background in computer vision and medical imaging. The developers offer function documentation and technical support via direct and indirect communication, but as the project is still in development the communication can be slow and the documentation under continuous change.

ITK is entirely written in C++ and follow a strict templated style of implementation, resulting in cross-platform compatibility and easier integration as the developer can address all functionality in the same fashion. There are no C# wrappers available. Visualisation of the data or general graphical user interfacing must be handled through other frameworks, as ITK only contains processing algorithms, general structures, and IO.

All data structures in ITK are considered N-dimensional matrices. To some degree vector and matrix math is implemented, but not completely. Principal component analysis and partial least square regression are not implemented. The framework enables implementation of your own algorithms and filters, and the documentation and introduction contains examples of this.

It is suggested in the documentation and user mailing list that ITK is slower than OpenCV, MATLAB, and IDL, partly due to the N-dimensional nature of ITK implementations. Beyond code alterations for issues discovered and discussed by the users, ITK is not performance optimised. ITK algorithms are mainly coded for multiple processor systems, and in such an environment it may outperform single processor algorithms, such as a majority of MATLAB algorithms.

ITK was considered for this thesis based on a light introduction through previous courses as a medical computer vision toolkit, the unified constructs and templates which eases the development usage over time, and the proven handling of the amounts of data needed for hyperspectral image analysis. Also, one of the needed algorithms is already implemented. ITK was not chosen, as it lacked full support for vector and matrix math in addition to possibilities for viewing, loading, and storing of the data.

MATLAB

MATLAB (The Mathworks Inc. International) is an interactive computation environment and a high-level programming language. The MATLAB programming language is constructed specifically for matrix and vector operations, and core functionality is performance and memory optimised. It runs on Windows and Linux with both 32-bit and 64-bit architectures. MATLAB is not written for a single scientific field. It finds uses in fields such as algorithm prototyping, signal and image processing, communications, statistics etc.

MATLAB is supported and updated regularly. Latest available version at the time of writing was MATLAB 7.4 (R2007a). In addition to extensive literature and in-product documentation, a large user community exists. The community, called MATLAB Central, is an open exchange of files and information related to MATLAB. Developers at Mathworks are encouraged to reply directly to user questions at MATLAB Central. The users typically range from undergraduates to research doctorates, and the community contribution is high relative to the number of users and comparable software.

The documentation of MATLAB as a whole and its individual components are considered of very high quality. There are tutorials for beginners, intermediate users, and advanced users available directly in the application and also online. Each function has documented input arguments, output arguments, and function behaviour. There are also one or more examples of usage for every function.

MATLAB contains many directly usable algorithms for this thesis, such as PCA, watershed segmentation and other simple image processing operations. Other functionality such as PLS, Clustering, and data read and write can be found or adapted from source code available at the user community's file exchange. MATLAB's core functionality is extended through licensed toolboxes that add relevant functionality, for example the Image Processing Toolbox. Natively, MATLAB can show colour and grey scale images quickly, and MATLAB has as mentioned in Section 3.2.1 been used for inspection and presentation of data.

One toolbox, the MATLAB Compiler, enables compilation of MATLAB code for integration with other laguages and applications. MATLAB can initiate and call any linked library, and can in itself create linked C and C++ libraries in addition to stand alone applications. On Windows, the resulting DLL-file can be called from C, C++, and C#. MATLAB is however an interpreted language, meaning it is translated to machine instructions at run time. Because of this, compiled code can not be run without MATLAB or MATLAB Component Runtime (MCR) installed. The MCR functions as a runtime interpreter of deployed code, and is freely available for download and distribution.

Interactive Data Language

IDL, short for Interactive Data Language, is a scientific programming language and environment sold and licensed by ITT Visual Information Solutions, formerly the subsidiary Research Systems Inc. The image analyser ENVI, previously mentioned used as a data viewer in Section 3.2.1, is fully implemented in IDL. The language is constructed specifically for interactive processing of complex data, such as hyperspectral and multispectral images. The interactivity, usage, and target user base are similar to that of MATLAB mentioned in the previous section. The syntax of IDL is similar to the programming language FORTRAN.

Several of this thesis planned algorithm implementations require vector and matrix math, and IDL is written and optimised around a vector and matrix math core. IDL is written for Windows, Mac OS, and Linux and supports 32-bit and 64-bit architectures. In addition some of its core functions support multithreading. Function arguments are all passed by reference, to conserve memory. All these factors are considered performance enhancing.

IDL is a commerical product and is directly supported by the developing company. The documentation for IDL is extensive, and contains detailed information, example usage, and known limitations. The user base of IDL, and often ENVI, are scientific users with high technical knowledge, and there exists a user knowledge base and contribution web site, infoNET. Compared to the number of files available and active users at MATLAB Central, infoNET is a considerably smaller user community.

IDL can be interfaced with external languages such as ActiveX control, Windows Component Object Model (COM), and Java. Programs and libraries in other languages can be called directly from IDL.

For the development of ENVI, several tools were developed for imaging purposes and later integrated into the base functionality of IDL. PCA, watershed, ratio, and difference exists in IDL or are available on the user community site. Clustering and PLS could not be found readily implemented in IDL. Data loading and storing is well implemented and documented in IDL, and the product information states that most formats known can be loaded and written.

One previously famous usage includes the management, analysis, and presentation of the multispectral imaging data from the two Mars Exploration Rovers. A set of IDL routines was developed by scientists and engineers at Cornell University called MERtools, used for monitoring camera health, data tracking, and data analysis.

The usage of ENVI and IDL required a license and a hardware USB-key. During this thesis, only one key was available, and this is seen as a limitation of development with IDL. Distribution is possible using the IDL Virtual Machine, a free runtime framework for compiled IDL code. There are clear similarites between the IDL Virtual Machine and the MATLAB Component Runtime.

SciCraft

SciCraft (Chemometrics and Bioinformatics Group, NTNU, Norway) is a open source data analysis software. The framework gives developers and users access to statistical, chemometrical, and partially imaging functions through a building-block user interface. At the time of writing, the software was listed by its developers as immature and incomplete. SciCraft is aimed at providing solutions for data analysis rather than technical issues such as loading and storing in different formats.

The documentation is comparable to the development of the main software. It can be characterised as unfinished, and has not been updated for the last big revision of SciCraft. The documentation is based on the Wiki technology, enabling active users to add, edit, and correct information.

The developers of SciCraft at this stage offer per user answers and support. There is no user community site beyond the documentation Wiki at the time of writing. SciCraft can however utilise many other language inputs such as MATLAB code files through a unified interface, and this gives it the combined input of several user communities.

Based on the programming language R, also known as GNU S, SciCraft has the basis for good performance. R is optimised for statistical computing, graphics, and large scale vector and matrix math.

SciCraft contains implementations of PCA and PLS, but not clustering, snakes, or image arithmetic operations. The ENVI file format is currently not supported in SciCraft.

The SciCraft user interface is based on GTK, a cross platform graphical user interface framework. No data viewer beyond plots is currenty implemented.

Evaluation Conclusion

The summary of each framework's fulfilment of the requirements is shown in Table 3.8. Based on the evaluation criteria in Table 3.7, MATLAB was chosen as environment for algorithm development. The compiled algorithms can be interfaced from most programming languages. C# was chosen as the programming language for its high level structure and easy user interface design, discarding C++ after a short test implementation period.

Table 3.8: *Framework comparison with weighted criteria from Table 3.7. The frameworks can fulfil the singular criterions completely (shown with a Y), partially (shown with a P), or not (shown with a N). The final ranks can be seen in the bottom row, where 1st is the top rank.*

Criterion	**Weigth**	**OpenCV**	**ITK**	**IDL**	**MATLAB**	**SciCraft**
C1	High	Y	P	Y	Y	Y
C2	High	P	P	Y	Y	P
C3	Medium	Y	Y	Y	Y	Y
C4	Medium	P	N	P	Y	P
C5	Medium	P	P	Y	Y	P
C6	Low	N	Y	Y	Y	P
C7	Low	N	N	Y	P	N
C8	Low	N	N	Y	Y	N
C9	Low	P	P	Y	Y	N
Ranked		3rd	4th	2nd	1st	5th

3.3 Implementation Details for Algorithms

This section presents the details surrounding the implementation of the three enhancement algorithms and the two segmentation algorithms described in Chapter 2. The concerns and changes of implementing the theoretical methods are presented here. Where deemed necessary, reader friendly pseudo code is presented in a listing.

Ratio and difference were implemented in C# and experimented with through the batch program also written in C#. Principal component analysis, K-means clustering, and watershed segmentation were implemented in MATLAB, compiled using the MATLAB Compiler, and experimented with through the batch application presented in Section 3.4.

3.3.1 Ratio and Difference

The arithmetic operations ratio and difference are performed on a pixel-by-pixel basis between two images in the presented implementation. For ratio, there is a common image processing problem when applying Equation 2.4 to the input. The problem arises when the value of the denominator pixel equals zero. Dividing by zero is not defined. The zero value must be meaned by its neighbours (called biased), or more commonly, incremented by a value, for example 1 if the image is of an integer data type [7]. Consider a denominator approaching zero. The resulting fraction will approach infinity regardless of nominator value. In a discrete image, infinity is undefined, and typically has a range of 0 through T, where T, for example, equals 65535 for unsigned 16-bit integers. The solution of the divide-by-zero problem is to reduce the range to 1 through T. The increment is hardly discernable from the zero intensity, and does not introduce the white noise pixels that can occur from setting the resulting divide-by-zero fraction to the maximum range value. Instead it will simply leave the value as it exists now if the divisor is zero. This approach is not mathematically accurate, but will produce acceptable results.

Conserving the image range from the original image to the resulting image is important with arithmetic operations. The result image of for example ratio often ends up with a significant number of low pixel values. Image detail variation can be lost if the values are naively multiplied and rounded. To minimise loss and to ensure an effective use of the available value range, the stretch shown in Equation 3.5 was used.

$$P_{out} = (P_{in} - c)(\frac{b-a}{d-c}) + a \tag{3.5}$$

P_{out} is the resulting pixel value, while P_{in} is the input pixel value. The lower and upper pixel value limits are a and b respectively, while c and d is the lowest and highest pixel value, respectively, currently found in the input image.

To preserve the dynamic range of images while applying Equation 2.3, a simplified version of Equation 3.5 is used [7]. This can be written as:

$$G_{diff}(i,j) = (b + G_1(i,j) - G_2(i,j))/2 \tag{3.6}$$

where G_{diff} is the resulting image, b is the largest value possible for any given pixel, G_1 and G_2 are the two input images, and i and j are spatial pixel position within each image. Using this simple equation prevents negative values and retains the same dynamic range in the result images as in the originals. It is assumed that both input images have the same dynamic range for both algorithms. No source code for the ratio and difference algorithms has been included in the appendix as it considered being sufficient with the formulas mentioned.

3.3.2 Principal Component Analysis

Memory limitations restricted the use of internal MATLAB principal component analysis algorithm. Therefore, interface code was written both for the MATLAB PCA method and a independent NIPALS algorithm. Both were compiled with the MATLAB Compiler and used through the batch application.

The NIPALS algorithm computes the principal components iteratively, and uses less memory than the calculation of covariance matrix and eigen vectors, as described in Section 2.3.2. However, the numerical approximation of the eigenvector projection is slow. The pre set limit (a small number, typically between 10^{-6} and 10^{-7}) is approached in smaller steps for each principal component, and therefore using the MATLAB PCA algorithm was preferred. The NIPALS algorithm is presented as pseudo code in Listing 3.1. The source code for the PCA and NIPALS PCA code can be found in Sections D.1.1 and D.1.2.

Listing 3.1: NIPALS Pseudo Code

```
% NxMxP input image is transformed into NMxP
% and stored in matrix X
```

```
X = reorderData(inputData)

for i := 1 to numberOfPrincipalComponents
{
   % set u to the first row vector of X
   u := X(1,all)
   currentError := Inf
   % run until error smaller than
   % limit (approaches zero)
   while currentError > limit
   {
      % project X onto u to find
      % corresponding loading v
      v := (X*u')/(u*u')
      % normalize the length of loading
      % vector v to 1.0
      v := v / |v|
      % store the score vector u into uold
      uold := u
      % project the matrix X onto v to
      % find new score vector u
      u := (X'*v)/(v'*v)
      % transpose u from 1xn to nx1
      u := u'
      % difference between the previous
      % scores and the current scores
      d := uold - u
      % convergence of scores
      currentError := |d|
   }
   % store the calculated loading
   scores(i) := u
   % store the calculated loading
   loadings(i) := v
   % find the residual
   E := X - ( v*u )
   X := E
}
```

3.3.3 K-means Clustering

As mentioned, K-means clustering was implemented in MATLAB. The variables for the experimentation are a range of clusters and a range of iterations. Simplified pseudo code is presented in Listing 3.2. By default and used in the implementation, MATLAB uses the Euclidean distance calculation for the clustering. The function script implemented in this thesis can be seen in Section D.2.

Listing 3.2: K-means Clustering Pseudo Code

```
% NxMxP input image is transformed into NMxP
% and stored in matrix X
X := reorderData( inputData )

% for all the values in CLUSTERS
for each cluster in CLUSTERS
{
   % for all the values in ITERATIONS
   for each iteration in ITERATIONS
   {
      % use the MATLAB kmeans algorithm
      clusterResult( i, j ) := kmeans( X , cluster , iteration )
      save( clusterResult )
   }
}
```

3.3.4 Watershed Segmentation

Watershed segmentation was implemented using thresholding, morphological opening, the dilation-erosion difference (called the Beucher-gradient), and the watershed transformation.

The morphological opening used a structural disk element with a radius of six. The element size was determined through inital testing on single bruise images to find at which size the features were sufficiently morphologically opened. For the Beucher-gradient, a square 3x3 structural element was used.

The input images were stretched between their minimum value and maximum value, and their range normalised to the range from 0.0 to 1.0. Listing 3.3 shows pseudo code for our watershed segmentation algorithm using MATLAB's watershed function. Code for the full implementation can be found in Section D.3.

Listing 3.3: Watershed Segmentation Pseudo Code

```
% the input image (a grey scale image) is normalised
greyImg := normimage( greyscaleImage )
% adjust the image range so that maximum value is
% pure white, and minimum black
adjImg := imadjust( greyImg )
% for all the pixels in the image
for each pixel i in adjImage
{
   % threshold the image with the predefined threshold input
   % resulting thresholdImg is a logical image with 1 or 0 (true
       or false)
   thresholdImg(i) := adjImage(i) > predefinedThreshold
}
% the two structural elements are defined
elementSquare := ones( 3, 3 )
```

```
elementDisk := disk( 6 )
% the threshold image is morphologically opened with the disk
    element
openImg := dilate( erode( thresholdImg,elementDisk ),elementDisk )
% find the Beucher−gradient using the 3x3 structural element
beucherImg := dilate( openImg,elementSquare ) − erode( openImg,
    elementSquare )
% find the distance from every black pixel to the nearest white
    pixel
distanceImg := distance( invert( beucherImg ) )
% use the watershed transform on the distance image
labelImg := watershed( distanceImg )
```

3.4 Application Design

This section gives an overview for the design of an application that was created and used during the experimentation. First, some background and a fictional story is presented to explain possible use. This is followed by a brief look at the underlying functionality.

It was decided to create a batch program that could set up as many tests as required and run these sequentially. This was done as some of the algorithms are computationally expensive. The batch program allowed all tests to be set up at the end of the day, and during the night. User input is given through a simple graphical user interface (GUI). The two windows in the GUI can be seen in Figure 3.4. An unobstructed view of the main window can be found in Section B.1. No data view functionality was implemented. A fictional user interaction is presented to give a simple example of possible use for the batch program:

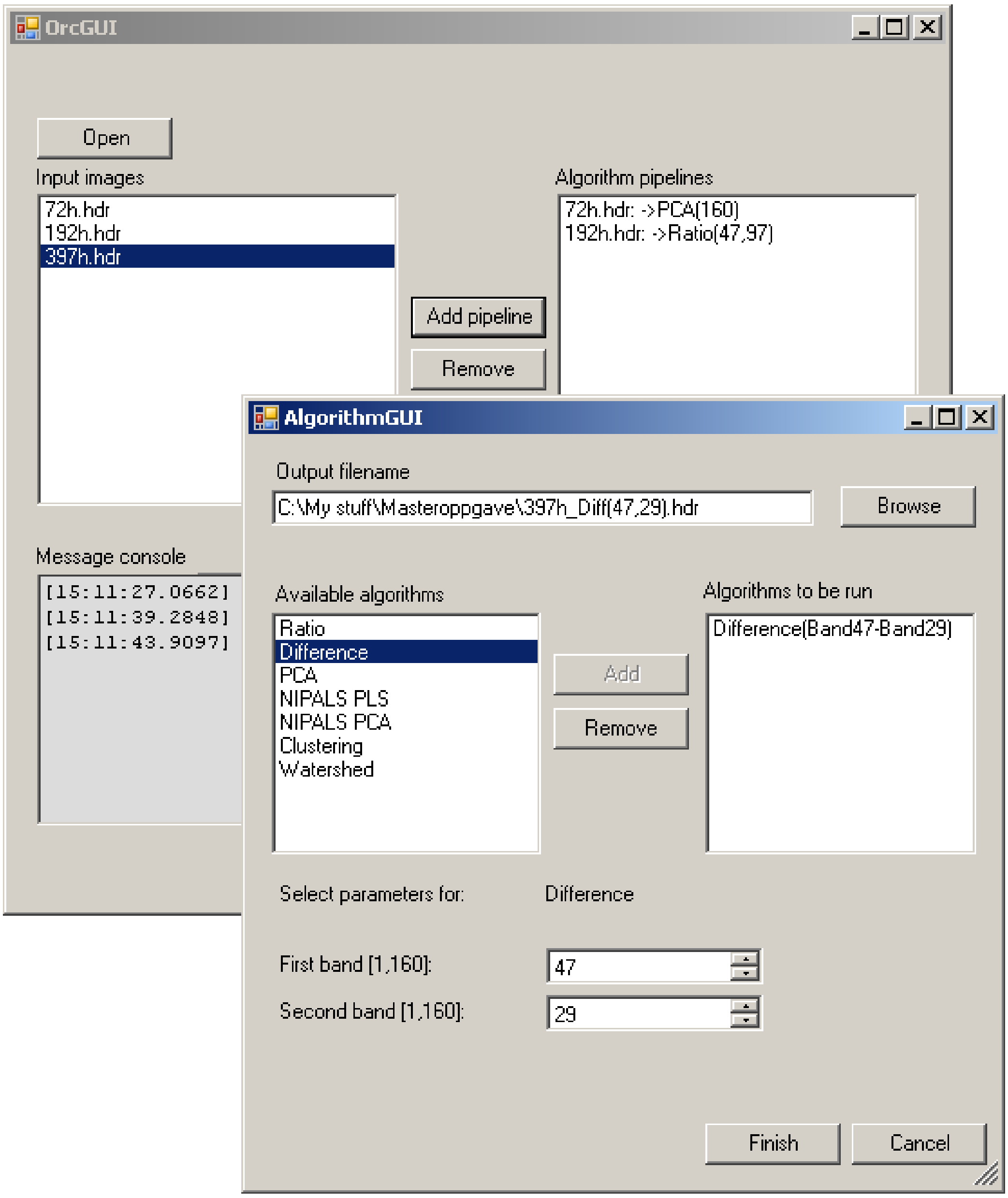
Meet Tim. Tim wishes to run some algorithms on a selection of images containing bruises. When the application starts, Tim is presented with the window in the back of Figure 3.4. He presses the 'Open' button and loads three image metadata files into the program. Tim can then see the available files in the 'Input images' box to the left. He selects '397h.hdr' and presses the 'Add pipeline' button. This opens the lower window. This window allows Tim to set up a processing pipeline. A pipeline is a series of algorithms run on the same image in a specified order with specific parameters.

Tim wants to run the difference algorithm, so he selects it from the 'Available algorithms' list. He sets the desired parameters and adds the algorithm to the pipeline by pressing the 'Add' button. Tim is now able to see the added algorithm with the specified parameters in the 'Algorithms to be run' list. The output filename can also be seen in the 'Output filename' box. After Tim adds the desired algorithm, he presses the 'Finish' button and is brought back to the first window. He is now able to see the pipelines in the 'Algorithm pipelines' list. When Tim is satisfied with the number of pipelines, he presses the 'Run All' button. After the program completes, Tim returns to find all his problems solved.

A simplified class diagram showing core functionality is presented in Figure 3.5. Only elements that help the reader to understand core functionality are presented, meaning that some classes present in the program are omitted from this diagram. All the classes presented, except the MATLAB package was implemented in C#. The program is built around a 'BatchController'. Among other things, the controller contains a list of available algorithms and images. The two most important functions of the controller are to start the graphical user interface (GUI), and to start the batch program.

The images are read and written through an I/O module, which currently only supports the ENVI file format. Support for other file formats can be included by implementing the 'Input/Output' interface. The 'Image' class contains the information related to images in two fields called 'metaData' and 'data'. Metadata includes height, width etc., while data is the actual image points stored in an array. Only ENVI headers are supported. If other file formats are needed, they should inherit from 'Metadata'. All the algorithms are implemented from the 'Algorithm' interface, which forces them to have a universal run method that the 'BatchController' can call. The run method takes an image as input and returns the treated image.

Of the algorithms, ratio and difference were implemented in C#, while PCA, NIPALS PCA, clustering, and watershed were implemented in MATLAB. The wrapper classes contain individual set of parameters for each algorithm along with a method that uses subprograms coded in MATLAB. These subprograms are made in the Microsoft Windows Dynamic-link library (DLL) format, which is a standard for intra- and subprogram communication and reusable components.



Input images

72h.hdr
192h.hdr
397h.hdr

Algorithm pipelines

72h.hdr: ->PCA(160)
192h.hdr: ->Ratio(47,97)

Message console

```
[15:11:27.0662]
[15:11:39.2848]
[15:11:43.9097]
```

Output filename: C:\My stuff\Masteroppgave\397h_Diff(47,29).hdr

Available algorithms

Ratio
Difference
PCA
NIPALS PLS
NIPALS PCA
Clustering
Watershed

Algorithms to be run

Difference(Band47-Band29)

Select parameters for: Difference

First band [1,160]: 47

Second band [1,160]: 29

Figure 3.4: *The two windows used in the batch program.*

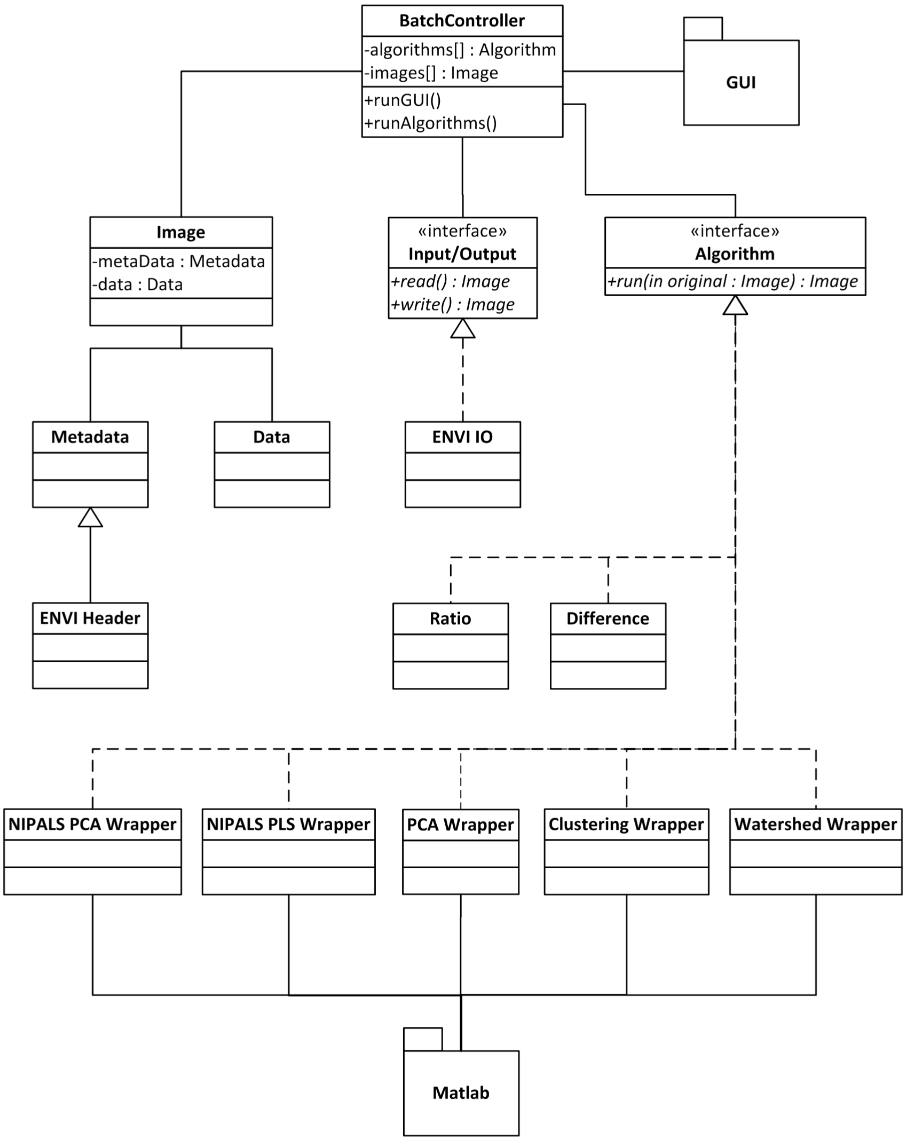


Figure 3.5: *A simplified class diagram showing core funtionality.*

Chapter 4

Experimentation

This chapter gives an overview of the tests that were done during experimentation. First, a list of the algorithms that were tested is presented to give a brief overview. Secondly, general guidelines for result analysis are presented before each algorithm is described in the same order as listed. The algorithms that were tested are:

- **Difference** - Initial tests were run on selected images, to determine parameters for further testing. These parameters were then applied to the images in Section A.2
- **Ratio** - Initial tests were run on selected images, to determine parameters for further testing. These parameters were then applied to the images in Section A.2
- **Principal Component Analysis** - Algorithm was applied to the images found in Section A.2. Also tested on three radiance images, to examine any differences between reflectance and radiance images
- **Clustering** - Initial tests were performed on a reflectance image and the principal components of the same image to determine parameters for further testing. Identified parameters, if any, were used on a reflectance image of another subject, and the principal components of that reflectance image
- **Watershed** - Segmentation was performed on four grey scale images with the intention of segmenting out the two (possibly three) bruises present in the image. Three of the four input grey scale images were the best results from the enhancement algorithms, and the fourth was a single band from the reflectance image were the bruise was clearly visible

Results are commented based on a visual analysis, and divided into two main categories. These are positive (+) and negative (-). This classification is subjective, meaning that other observers might get different results. When a result is judged to be positive it is mainly because bruises are clearly visible or because results seemed promising for further bruise segmentation. A result that does not clearly show bruises or contain a high level of noise is judged to be negative. If results are lacking bruises, but show other interesting features, like blood vessels, they were marked as negative. The features were commented upon, and some images that displayed interesting properties, other than bruises, are presented. Other results might be valuable for future work. Any further selection beyond (+) and (-) is specified were this is necessary. Unless stated otherwise, all images used for testing were reflectance images.

4.1 Difference

Input parameters for the difference algorithm are two bands from an image cube. Each image cube contains 128 or more bands, and it was decided not to test and analyse the complete range of permutations. An image cube with 128 bands would give 16256 permutations. The experimentation was split into two phases for testing of the difference algorithm. The first phase determined parameters on a reduced data set, and the second phase used these parameters on a larger set of images.

For the first phase, two images were selected from the larger set: A VNIR (Visible and Near Infrared) image (I1-1) and a SWIR (Short Wave Infrared) image (I2-1). Details on the full set can be found in Section A.2. The SWIR and VNIR images represent the total spectral range of the full data set. I1-1 is a reflectance image, while I2-1 is a radiance image. The reflectance image of I2-1 was not satisfactory for experimentation, as described in Section 3.1.4.

To select parameters (bands) for the first phase, two different approaches were used:

1. **Approach one:** Band selection based on absorption peaks.
2. **Approach two:** Band selection based on visual inspection of all bands.

The first approach utilise information from biomedical optics related to some of the different compounds that might be found in a bruise. Information on the compounds can be found in Table 2.1. Section A.3 details the connection between wavelength and band number. If an absorption peak falls between two bands, the bands above and below are listed. For experimentation, the band below was selected.

The absorption peaks in Table 2.1, are only considered valid for reflectance images, as the lighting is expected to alter amplitude in radiance images. Because of this radiance image I2-1 does not fit the first approach, and will be exempt from this test.

Table 4.1 shows the different permutations that were tested in the first phase. Note that only the first combination is listed for testing in reverse order. This was done to verify that the algorithm was running correctly. If parameters are interchanged in the difference algorithm, it will produce the mathematical inverse. Inverse images of already produced results do not provide new information. By avoiding the reverse order, the number of permutations was halved.

Bilirubin has one absorption peak which is affected by contributions from subcutaneous transport of bilirubin by lymphatic flow followed by diffusion into dermis [27]. Both possible absorption peaks found in Table 2.1 were tested.

The second approach is based on visual inspection of all bands in an image cube. This approach was used on both I1-1 and I2-1. Based on similarities regarding bruise visibility, bands were gathered into groups. The groups and their description can be found in Section A.4 and Section A.5 for image I1-1 and I2-1 respectively. One or two bands were chosen randomly from each group and then tested against one another.

The parameters found from the two approaches were used to run tests on I1-1 and I2-1. The results were then analysed and divided into positive and negative groups, following the guidelines presented in the introduction to Chapter 4. If more than four

Table 4.1: *The different permutations that were tested on image I1-1.*

Parameter one	**Parameter two**
Deoxyheamoglobin	Oxyheamoglobin
Oxyheamoglobin	Deoxyheamoglobin
Bilirubin	Deoxyheamoglobin
Bilirubin	Metheamoglobin
Bilirubin	Oxyheamoglobin
Deoxyheamoglobin	Metheamoglobin
Oxyheamoglobin	Metheamoglobin

positive results were obtained, four were chosen amongst the positive results. This selection was done by comparing bruise visibility. Those with the most visible bruises were selected, unless a result with similar details and more visible bruises already had been selected.

After deciding on parameters in the first phase, the second phase was to apply the selected parameters to all images listed in Section A.2 not already tested. Algorithm parameters were adjusted for variations between the different cameras in accordance with the information in Section A.3, which details the connection between wavelengths and bands for each camera model.

4.2 Ratio

The approach used for the ratio algorithm is very similar to that of difference, with minor changes. Ratio was also divided into two phases, and parameters were selected in the same manner. The only change was with regard to the inverse. Interchanging parameters does not produce the mathematical inverse, but it was suspected that the two images would contain more or less the same details with brightness variations similar to the inverse. For that reason, parameters for phase one was selected using Table 4.1 for ratio as well.

4.3 Principal Component Analysis

The principal component analysis (PCA) was tested on all images found in Section A.2. The principal components for each image were then analysed with respect to bruise visibility. The number of principal components computed is the same as the number of bands in the original image cube. Other interesting features were commented along with bruise visibility. Three radiance images was also tested and compared with their reflectance counterparts. This was done to see how results would differ if radiance images had been used as opposed to reflectance images.

4.4 K-means Clustering

Similarly to the difference and ratio algorithms, K-means Clustering was tested in two phases. The first phase was a test of a wide range of parameters on reflectance image I1-1 and a selection of principal components from reflectance image I1-1. The input parameters for the algorithm are the number of clusters in the output and number of iterations. The clustering parameters used for the first phase can be found in Section A.9.1. They range from 2-50 clusters and 1-125 iterations.

If the parameters proved successful on the input image, they were used on another image of the same type for verification with a different individual, i.e. successful parameters on the principal components of a VNIR image would be tested on the principal components of another VNIR image. Image I1-1 was used in the first phase, and image I4-3 in the second phase.

4.5 Watershed Segmentation

Details on the theoretical part of watershed segmentation can be found in Section 2.3.4, and implementation details in Section 3.3.4.

Wathershed segmentation was performed on four grey scale images. Three of the images were the subjectively evaluated best results from the three enhancement algorithms difference, ratio, and principal component analysis. The fourth was taken from band 47 of the reflectance image I1-1. This band has been visually identified as one of the bands with best feature contrast, i.e. clearly visible bruises. The band from the original image was tested to get an indication of whether the enhancements algorithm had a positive or negative impact on segmentation. The varying parameter for the implementation presented in this thesis is the thresholding value. For the four images, the thresholding value was varied from 0.0 to 1.0 with 0.05 in increments.

Results were evaluated based on area in the input image covered by the segmentations. For example if the segments covered only the upper half or only the centre of a bruise, the segmentation was regarded as less successful than a segmentation covering the entire visible bruise.

4.6 Summary

This chapter concludes with a quick summary of what has just been presented. Difference, ratio, and clustering all used two phases. The first phase was used to test parameters on a reduced number of images. The second phase then used these parameters on other images. PCA was tested on all the reflectance images and three radiance images. Watershed was used on three selected result images and one original image. An overview of the algorithms and the images they were applied to can be seen in Table 4.2. All the images, except three, are reflectance images. If a ✓is used, it means that the algorithm at the top of the column was used on the image in the corresponding during experimentation. Not tested (NT) means that an image was going to be tested but was omitted. This could happen if a preliminary requirement was not fulfilled, for example, image B was only tested if a positive result from image A was observed. Otherwise testing on image B would be omitted. At the bottom of Table 4.2 there are some rows that require additional explanation. PCA (image id) means the PCA result image from for example I1-1. The last three rows are the best results (BR) from the algorithms specified within parentheses.

Table 4.2: *The images with unique id (left column) and the algorithms that used them as input. If a ✓is given, it means that an image was used during experimentation of by an algorithm. Not tested (NT) means that an image was going to be tested but was omitted.*

Image ID	**Difference**	**Ratio**	**PCA**	**Clustering**	**Watershed**
I1-1	✓	✓	✓	✓	✓
I1-1 (radiance)			✓		
I1-2	✓	✓	✓		
I1-3	✓	✓	✓		
I2-1	✓	✓	✓		
I2-2	✓	NT	NT		
I2-3	✓	NT	NT		
I3-1	✓	✓	✓		
I3-2	✓	✓	✓		
I3-2 (radiance)			✓		
I3-3	✓	✓	✓		
I4-1	✓	✓	✓		
I4-2	✓	✓	✓		
I4-3	✓	✓	✓	NT	
I4-3 (radiance)			✓		
PCA (I1-1)				✓	
PCA (I4-3)				✓	
BR (ratio)					✓
BR (difference)					✓
BR (pca)					✓

Chapter 5

Results

This chapter presents the results from applying the algorithms described in this report to images. The chapter has been divided into sections, were each section is named after the algorithm used. For ease of reading, the algorithms are presented in the same sequence as in Chapter 4. To limit the number of images presented here, only a selection of important images is shown in this chapter. A textual presentation is given for all result images.

5.1 Difference

The results from the difference algorithm have been divided into three sections to facilitate reading. The two first cover results from phase one and phase two. Details on the phases can be found in Section 4.1. Each section contains positive and negative results. The final section presents some observations that are not directly related to bruise enhancement.

5.1.1 Phase One - Initial Test Results

An overview of the positive results observed during the initial testing (phase one) can be seen in Table 5.1. The complete list featuring all results from parameters tested can be found in Section A.6.1.

Four parameters were supposed to be selected for each image in phase one for further use in phase two. Only two parameters for image I1-1 matched the selection criteria used for selection, since the other positive results were similar to the two chosen. For I2-1 there was one parameter that gave a positive result. Table 5.2 shows the parameters that were used further in phase two. The two parameters used on image I1-1 were chosen based on absorption peaks, while the one used on I2-1 was chosen based on visual inspection of all bands.

Result images from the difference algorithm using the parameters from Table 5.2 can be found in Figure 5.1 and in Section B.2.1 for image I1-1 and I2-1 respectively.

Figure 5.1(a) shows band 47 of the original image, whose wavelength is approximately at the second absorption peak for oxyheamoglobin. The oxyheamoglobin wavelength is well suited for the display of bruises. Figure 5.1(b) shows two bright circular bruises, while (c) have dark bruises. Figure 5.1(b) displays a 'racoon' effect around the bruises, while (c) has a dark area below the bruises and in the upper left corner. Skin structure has been lost in both images.

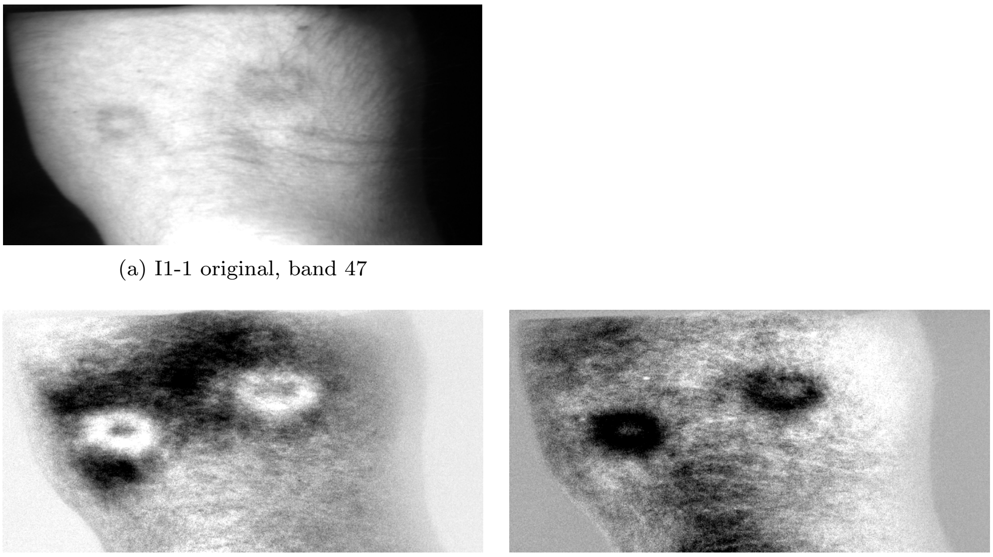
Table 5.1: *An overview of the positive results from the initial testing during phase one. Using absorption peaks for band selection is called approach one, while band selection based on visual inspection is called approach two. Approach one is not applicable to image I2-1, since it was captured using camera H3. The absorption peaks used for approach one are outside the range used by camera H3.*

Image	**Approach One**	**Approach Two**	**Comments**
I1-1	8 of 28(28,6%)	0 of 12(0%)	Two visible bruises
I2-1	N/A	1 of 6(16,7%)	Two visible bruises. Approach one not applicable. See caption for details.

Table 5.2: *Three parameters (bands and wavelengths) that gave the most promising result images for the difference algorithm. Excerpt from tables in Section A.6.1.*

UID	**Bands (wavelength)**	**Comments**
I1-1	22-38 (480nm-542nm)	Two visible bruises surrounded by a dark area
I1-1	47-29 (576nm-508nm)	Two clearly visible bruises. Loss of skin structure
I2-1	110-142 (1443nm-1604nm)	Linearly stretched. Two visible bruises

Section B.2.1 presents a single result image from the experimentation on SWIR image I2-1. The bruises show as bright circles against a dark arm.



(a) I1-1 original, band 47

(b) Band 22 and 38

(c) Band 47 and 29

Figure 5.1: *The original image and two positive results from phase one using the difference algorithm on image I1-1. Parameters that were used are listed below each image*

Figure 5.2 shows an example of two negative result images from I1-1, while Section B.2.1 presents the same for I2-1.

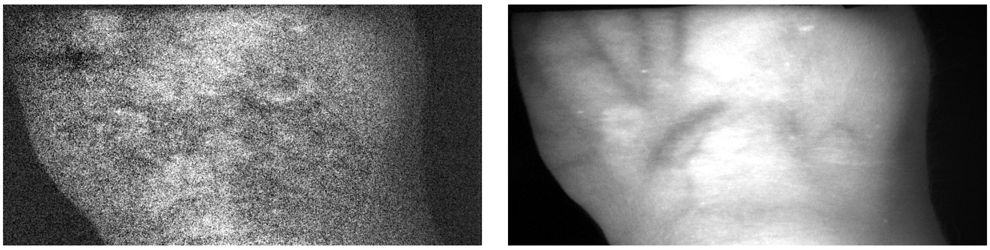
Figures 5.2(a) and (b) show results containing bruises that are difficult to discern from the rest of the image. Figure 5.2(a) also contains a high degree of 'salt-pepper' noise [1]. One bruise is visible in Figure 5.2(b), however the other bruise is indiscernible. Two bright areas can be seen below and approximately on the location of the missing bruise.

The negative results from I2-1 contained no discernable noise but the bruises are weakly visible. Blood vessels are visible in one of the images.

5.1.2 Phase Two - Extended Test Results

After parameters had been selected in phase one, they were applied to the rest of the images found in Section A.2. Results from that test can be found in Section A.6.2. Table 5.3 shows an overview of the number of positive results for each parameter. The result images have been grouped together based on similarities in visual contents. Images that contained known bruises are presented first, followed by images that were either taken before injury or before the injury was possible to observe visually.

[1] Salt-pepper noise are randomly occurring white and black pixels in images.



(a) Band 42 and 38

(b) Band 97 and 29

Figure 5.2: *Two negative results from phase one using the difference algorithm on image I1-1. Parameters that were used are listed below each image.*

Table 5.3: *An overview of the positive results from phase two using the difference algorithm. The table is divided into two rows. The first shows result from images with visible bruises, the second from images with no visible bruises. A positive result on an image with no visible bruise is considered a false positive, hence a low percentages is considered good. The last column is only valid for images from I2, because of the camera used (H3). There are no images captured by camera H3 that did not contain visible bruises.*

Content	**480nm-542nm**	**576nm/508nm**	**1443nm-1604nm**
With visible bruise	2 of 5(40%)	2 of 5(40%)	2 of 2(100%)
No visible bruise	0 of 3(0%)	0 of 3(0%)	-

Figure 5.3 shows nine images. The images in the top row are the originals, while the two rows below them are result images from the difference algorithm. Figure 5.3(a) shows a weakly visible square bruise, (b) shows a single circular bruise, while (c) shows two circular bruises. The uppermost bruise in (c) has a dark centre, making internal contours hard to see. Figures 5.3(f) and (i) both show clearly visible bruises and both internal and external contours are visible. These were the only two judged to be positive in this figure. Figure 5.3(e) shows a visible bruise, but its difficult to discern contours. The same can be said for (h), but it also shows a dark ring, in the area of the bruise, surrounding a brighter ring, which again surrounds a dark centre. The square shaped bruise is weakly visible in both (d) and (g). Striping effects are visible in all the result images.

The images used during phase one, I1-1 and I2-1, are both part of time series. Two additional images were also taken from each series. Both I1-1 and I2-1 were captured 72 hours after injury, while the other images are from 192 and 397 hours after injury. Section B.2.2 presents result images from I1-2, I1-3, I2-2, and I2-3. Bruises are visible for I1-2, but difficult to discern for I1-3. This is also true for the result images from the difference algorithm, where results from I1-2 shows more visible bruises compared to results from I1-3. I1-2 was judged to have positive result images, while they were judged to be negative for I1-3. The left bruise from I1-2 results has lost its contour compared to the right bruise.

The result images from I2-2 and I2-3 both have visible bruises, but they are weaker for I2-3. Both result images show bright bruises against a dark arm and they both contain striping effects. Result images were deemed positive, even for I2-3, since an improvement over the original can be seen.

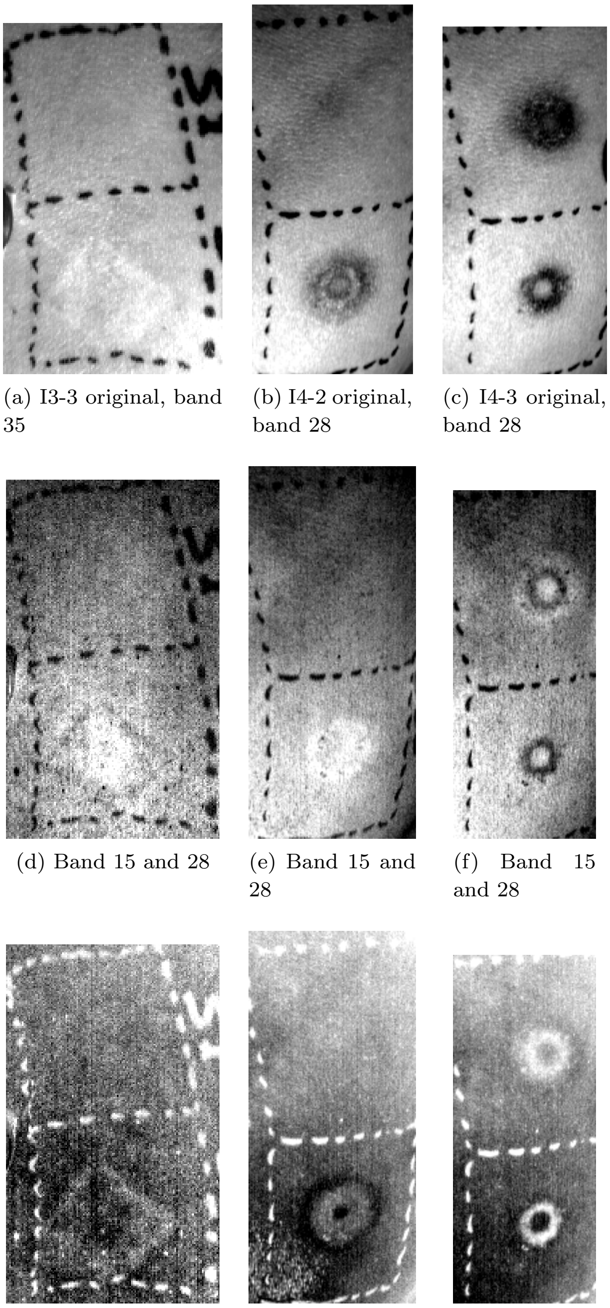
Section B.2.2 also covers results that had no clearly visible bruise in the original image. None of the tested parameters produced false positives. This means that there was no clear indication of bruises before or after testing. There are some marks that could be interpreted as a weakly visible bruise, both in the originals and the results, but since these cannot be determined to be bruises in either, they are considered to not contain visible bruises.

5.1.3 Other Observations

This section presents some observations that contained results not directly related to bruise enhancement. A brief example of what happens when the parameters for the difference algorithm are reversed is also presented.

Figure 5.4 shows a result image from I2-1. Blood vessels with a varying degree of thickness can be seen.

Two example result images from image I1-1, with their parameters reversed, are presented in Section B.2.3. The image to the left is similar to the inverse of the image to the right and vice versa, which is expected. A reversal of parameters in the difference algorithm produces the mathematical inverse. This example result was used as a visual confirmation of this theory.



(a) I3-3 original, band 35

(b) I4-2 original, band 28

(c) I4-3 original, band 28

(d) Band 15 and 28

(e) Band 15 and 28

(f) Band 15 and 28

(g) Band 35 and 21

(h) Band 35 and 21

(i) Band 35 and 21

Figure 5.3: *Three original images and six result images. The images at the top row are the originals, while the two rows below them are results from the difference algorithm using two sets of parameters. Parameters that were used are listed below each image.*

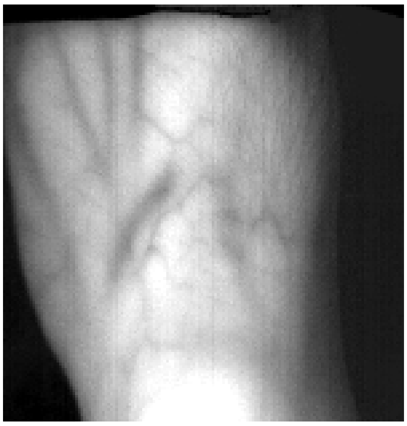


Figure 5.4: *A result from the difference algorithm showing visible blood vessels. Parameters used were band 20 and 62.*

5.2 Ratio

The ratio and difference algorithms have both been tested in two phases. For this reason, the results of the ratio algorithm and results of the difference algorithm are presented in the same manner.

5.2.1 Phase One - Initial Test Results

An overview of the positive results observed during the initial testing (phase one) can be seen in Table 5.4. The complete list featuring all results from phase one can be found in Section A.7.1 Four parameters were selected from the positive results using image I1-1, but no positive results were observed for image I2-1. Since there were no parameters selected from testing of image I2-1, phase two was not conducted on the images I2-2 and I2-3. The four selected parameters that were used in phase two can

Table 5.4: *An overview of the positive results from phase one using the ratio algorithm. Using absorption peaks for band selection is called approach one, while band selection based on visual inspection is called approach two. Approach one is not applicable to image I2-1, since it was captured using camera H3. The absorption peaks used for approach one are outside the range used by camera H3.*

Image	Approach One	Approach Two	Comments
I1-1	22 of 28(78,6%)	6 of 12(50%)	Two visible bruises
I2-1	N/A	0 of 6(0%)	Two visible bruises. Approach one not applicable. See caption for details.

be found in Table 5.5. Result images from applying the selected parameters to image I1-1 can be found in Figure 5.5.

Table 5.5: *Four parameters (bands and wavelengths) that gave the most promising result images for the ratio algorithm. Excerpt from tables found in Section A.7.2.*

UID	Bands (wavelength)	Comments
I1-1	22/38 (480nm/542nm)	Two bright obvious bruises with dark areas surrounding them
I1-1	47/29 (576nm/508nm)	Two obvious bruises. Lost some skin structure
I1-1	47/97 (576nm/760nm)	Two bruises and a possible bruise are emphasised
I1-1	70/47 (747nm/576nm)	Two clearly visible bruises and an additional possible bruise

Figure 5.1(a) shows band 47 of the original I1-1 image with two dark circular bruises. All result images show these two bruises, either as dark or bright circles. Figure 5.5(a) shows a 'racoon' effect around the bruises, while (c) and (d) show a triangle shape below and between the two bruises. Possible blood vessels are visible on the left side of (c) and (d), while some hair is weakly visible on the right side of (a), (c), and (d). There is noise in the background of all result images.

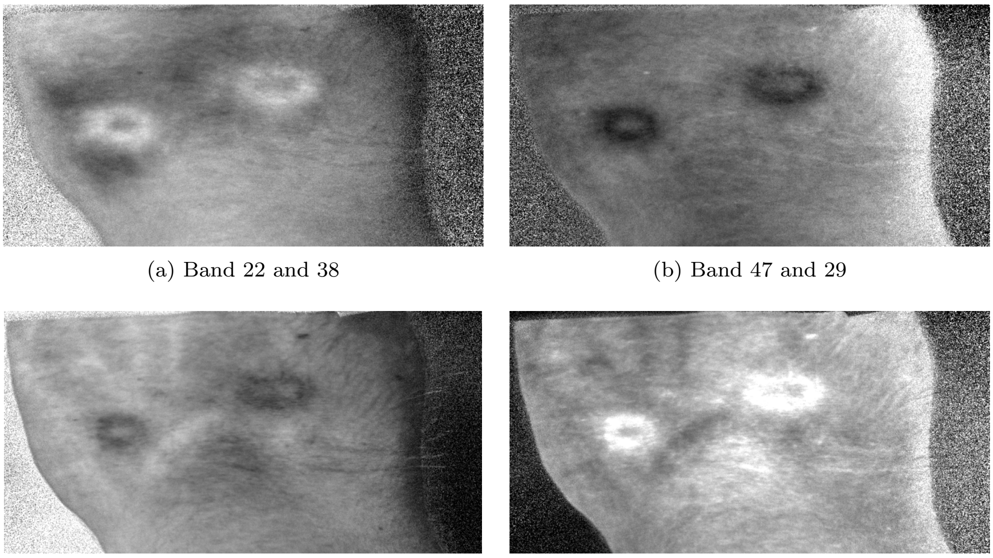
Examples of some of the results that were deemed negative can be found in Figure 5.6 and in Section B.3.1. Figures 5.6(a) and (b) both have barley visible bruises. There is noise on the right side of both images, while (a) has noise on the left side as well. Figure 5.6(a) has also lost skin structure details.

Section B.3.1 presents two examples of negative results from the initial testing of image I2-1. Both images show bruises that are difficult to discern. Hair is visible in both images.

5.2.2 Phase Two - Extended Test Results

The parameters that where selected in phase one, were applied to the rest of the images. These can be found in Section A.2. A table showing results from this test can be found in Section A.7.2. Table 5.6 shows an overview of the number of positive results for each parameter. The images presented in this section have been grouped together based on similarities in visual content. Images with bruises are presented first, followed by some examples of images that did not contain clearly visible bruises.

To facilitate comparison, images of tissue on a pig with visible bruises are gathered into Figure 5.7 which contains 15 result images. Column one (left column) shows the original images. From the top, these contain a bright square bruise, a dark circular bruise and finally two dark circular bruises. Each row shows four result images for each original image. The top row contains negative result images in addition to (l). Figure 5.7(l) was judged to be negative since the lower bruise is weakly visible. The



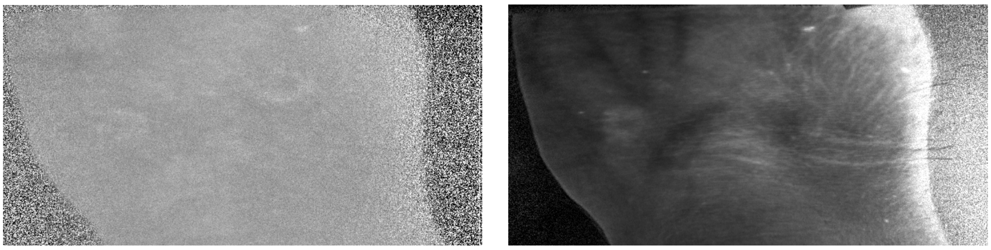
(a) Band 22 and 38

(b) Band 47 and 29

(c) Band 47 and 97

(d) Band 70 and 47

Figure 5.5: *Four positive results from phase one using the ratio algorithm on image I1-1. Parameters that were used are listed below each image.*



(a) Band 42 and 38

(b) Band 97 and 29

Figure 5.6: *Two negative results from phase one using the ratio algorithm on image I1-1. Parameters that were used are listed below each image.*

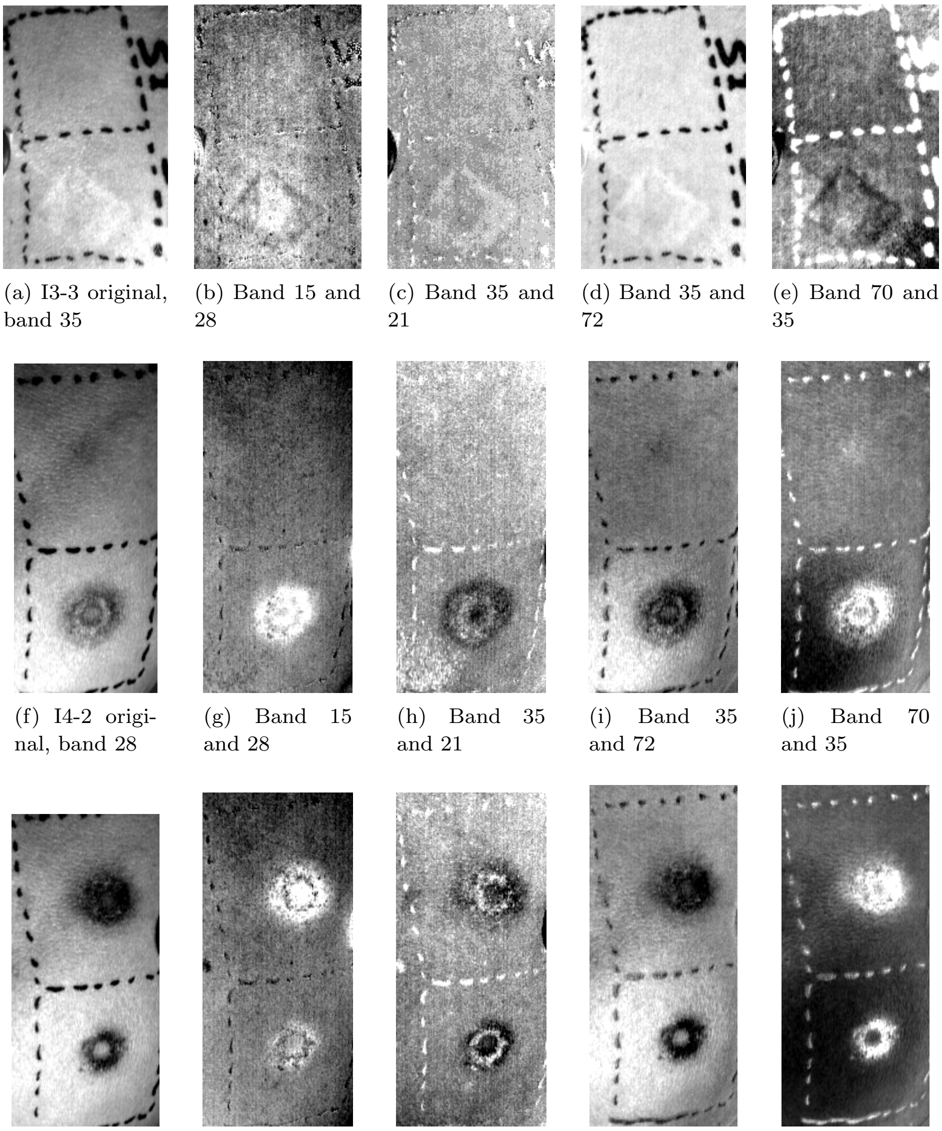
Table 5.6: *An overview of the positive results from phase two using the ratio algorithm. The table is divided into two rows. The first shows result from images with visible bruises, the second from images with no visible bruises. A positive result on an image with no visible bruise is considered a false positive, hence a low percentages is considered good.*

Content	**Wavelengths (nm)**			
	480/542	**576/508**	**576/760**	**747/576**
With visible bruise	3 of 5(60%)	3 of 5(60%)	3 of 5(60%)	4 of 5(80%)
No visible bruise	0 of 3(0%)	0 of 3(0%)	0 of 3(0%)	0 of 3(0%)

rest were judged to be positive result images. All result images contain some striping effect, but images in column two the most. Result images in the fourth column are similar to their respective originals. Most bruises in the result images show as either a bright or dark version of the original bruise. Figure 5.7(m) shows an example where parts of the bruise have a different brightness than the immediately surrounding area which is also part of the bruise.

The result images from I1-2 and I1-3 can be found in Section B.3.2. Bruises are less visible in the result images from I1-3 compared to those from I1-2. All the results from I1-3 were judged negative, while those from I1-2 were judged positive. There is however a difference in bruise visibility in the result images from I1-3, ranging from non-existent to close to positive. Band 22 and 38 along with 47 and 29 gave less visible bruises than band 47 and 97 along with band 70 and 47. The same can be said for I1-2, but it is not as obvious. Blood vessels seem to show equally well when comparing result images from I1-2 to I1-3.

Section B.3.2 also covers results from images of the pig that had no clearly visible bruise in the original image. None of the tested parameters produced false positives. This means that there was no clear indication of bruises before or after testing. There are some marks that could be interpreted as a weakly visible bruise, both in the originals and the results, but since these cannot be determined to be bruises in either, they are considered to not contain visible bruises.



(a) I3-3 original, band 35

(b) Band 15 and 28

(c) Band 35 and 21

(d) Band 35 and 72

(e) Band 70 and 35

(f) I4-2 original, band 28

(g) Band 15 and 28

(h) Band 35 and 21

(i) Band 35 and 72

(j) Band 70 and 35

(k) I4-3 original, band 28

(l) Band 15 and 28

(m) Band 35 and 21

(n) Band 35 and 72

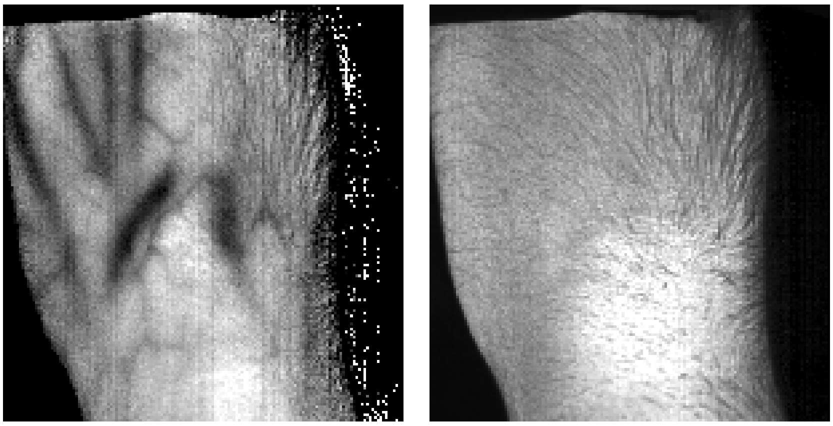
(o) Band 70 and 35

Figure 5.7: *Three original images and 12 result images. All the images are of a pig with visible bruises. The images in the left column are the originals, while each row show results from using the ratio algorithm with different parameters on the original images. Parameters that were used are listed below each image.*

5.2.3 Other Observations

This section presents some results not directly related to bruise enhancement. These might prove valuable for later research both with bruises and in other areas. A brief example of reversed parameters for the ratio algorithm is also covered.

Figure 5.8 shows some results from the testing with image I2-1. Figure 5.8(a) shows possible blood vessels, some thick and dark, others a bit thinner. Striping effects and some noise is visible. Figure 5.8(b) shows hair, but no blood vessels or bruises. A bright spot can be seen on the lower part of the arm.



(a) Band 20 and 62

(b) Band 62 and 110

Figure 5.8: *Two results from I2-1. The one to the left shows blood vessels while the one to the right show hair with no blood vessels or bruises. Parameters that were used are listed below each image.*

Section B.3.3 presents two result images from image I1-1 with their parameters reversed. The image to the left is similar to the inverse of the image to the right and vice versa.

5.3 Principal Component Analysis

This sections presents the results from applying the principal component analysis on the images described in Section A.2. All the images are reflectance images, except those from the I2 image series, which are radiance images. Three images where tested using both the reflectance version along with the radiance version. All the reflectance images results are presented first, followed by the radiance images results. Finally some results not directly related to bruise enhancement are presented. A complete list of all images tested and their comments can be found in Section A.8. The complete list also shows comments on results that have been excluded, since they did not contain any visible bruises. These are not shown in the result section. Table 5.7 gives an overview of the number of positive result images that were observed for both reflectance images and radiance images. Principal components that were judged to contain too much noise is not included in the percentages. This means that if the first 10 components out of 160 are the only ones with an acceptable signal-to-noise ratio, only the 10 first are considered when calculating the percentages. These 10 principal components can contain visible bruises or no bruises at all.

Table 5.7: *An overview of the positive results from testing with PCA. The table is divided into two rows and two columns. The first shows result from images with visible bruises, the second from images with no visible bruises. The first column shows the positive results for the reflectance images, while the second shows the same for radiance images. A positive result on an image with no visible bruise is considered a false positive, hence a low percentages is considered good.*

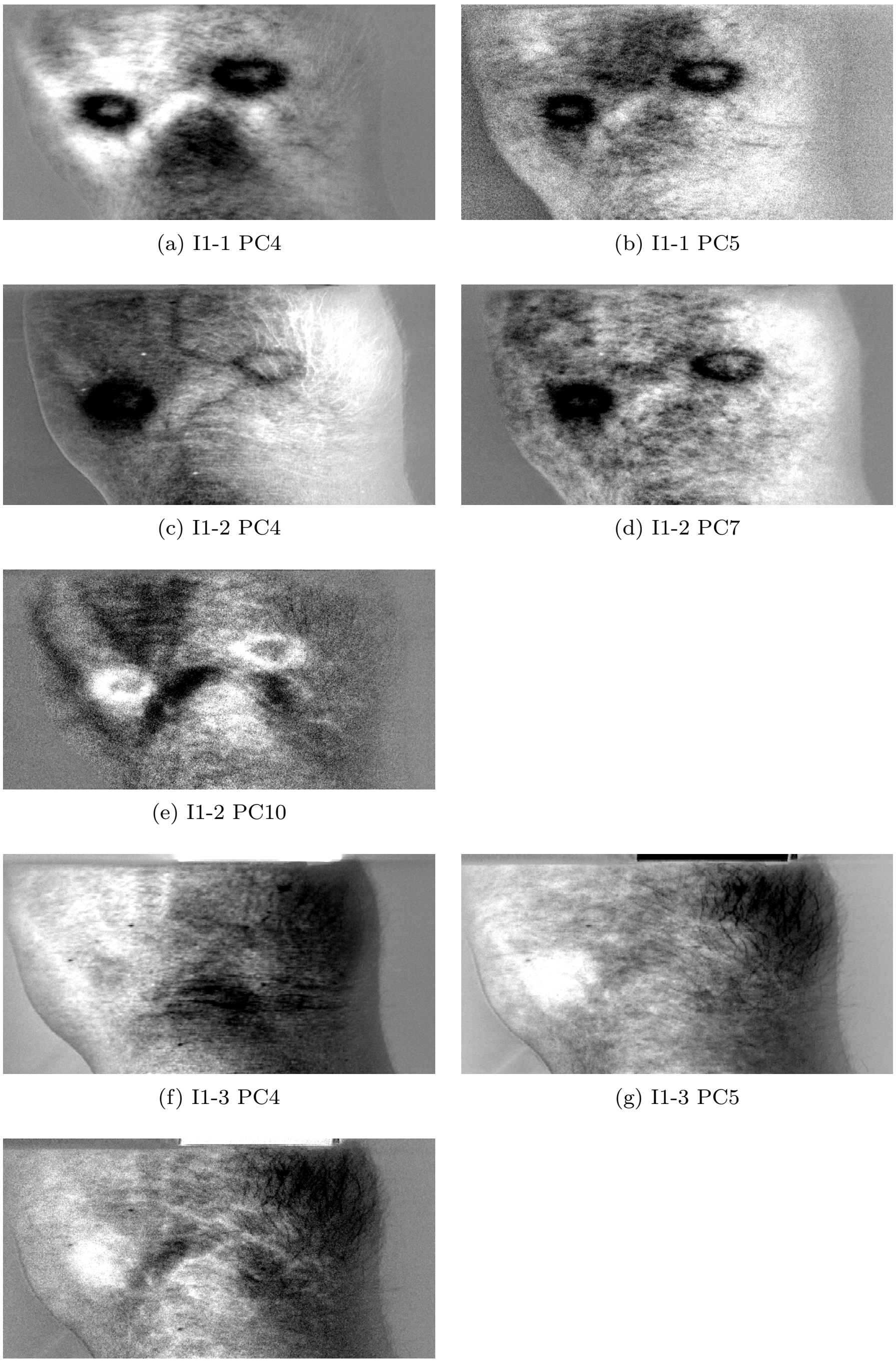
Content	**Reflectance Image**	**Radiance Image**
With visible bruise	12 of 49(24,5%)	17 of 55(30,9%))
No visible bruise	0 of 10(0%)	0 of 6(0%)

5.3.1 Reflectance Image Results

The reflectance images results have been grouped together, based on similarities in visual contents. Some result images from image series I1 is presented first followed by result images from images series I3 and I4. A couple of examples of negative results conclude this section.

Figure 5.9 shows results from image I1-1, I1-2, and I1-3. The two first had positive result images, but I1-3 had only negative result images. The three result images from I1-3 were included to show the result images that were considered best among the negative result images. They were judged to be negative since only one bruise is visible. The originals can be found in Figure 5.1(a) for the first image and in Section B.2.2 for the two last. Almost all the images show at least one clearly visible bruise. Figure 5.9(f) does not show any bruises clearly, but shows the left bruise reasonably well compared to the original. The right bruise is either not possible to see or weakly visible in Figures 5.9(f), (g), and (h). Figure 5.9(c) shows a thinner right bruise compared to (a), (b), (d), and (e). Blood vessels are visible in (a), (c), (e), (f), and (h), but (c) and (h) show thinner vessels connected to the bruises. Hair is clearly visible in (c) and visible in (f), (g), and (h). A triangle shaped area can be seen below and between most of the bruises, but is most clearly in (a). Figure 5.9(h) shows blood vessels in this area. Some noise can be seen in (b) and (e). Principal component (PC) 4 gave positive results for all the images, while PC5 gave positive results for two. No other PCs gave similar results.

Results from I3-3, I4-2, and I4-3 are shown in Figure 5.10. All bruises are clearly visible, except Figure 5.10(g) which is similar to the original. For this reason (g) was judged to be negative, but was included as it was the best result image from I3-3. The result images from the other two images were judged to be positive. The same



(a) I1-1 PC4

(b) I1-1 PC5

(c) I1-2 PC4

(d) I1-2 PC7

(e) I1-2 PC10

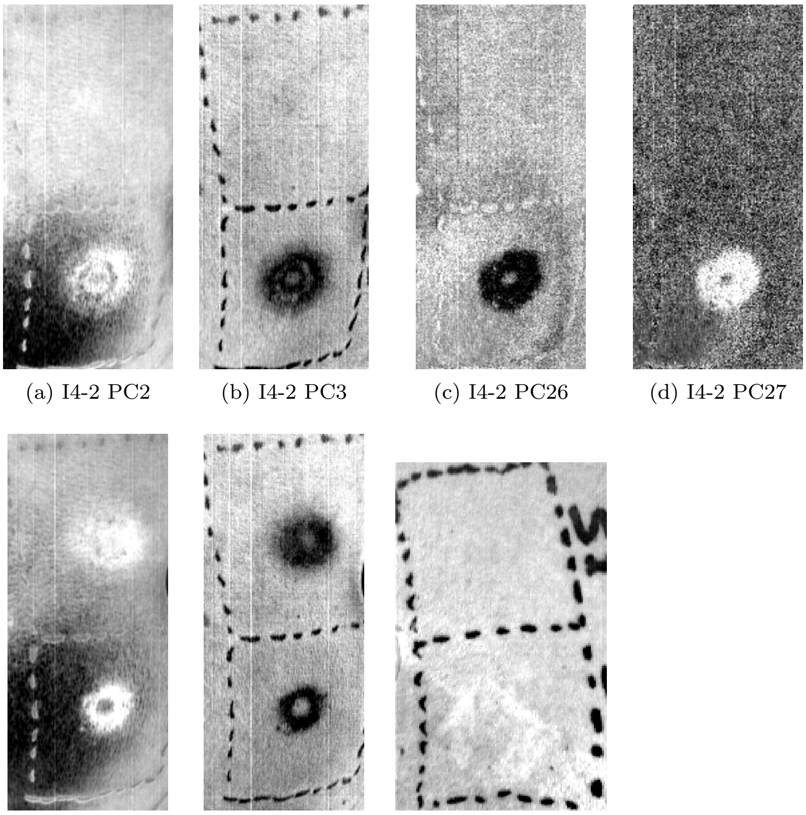
(f) I1-3 PC4

(g) I1-3 PC5

(h) I1-3 PC8

Figure 5.9: *Eight result images from using PCA on I1-1, I1-2 and I1-3. The original image used and the principal component (PC) presented are listed below each image.*

goes for striping effects, which is clearly visible in all the images except (g). Figure 5.10(d) has lost most skin structure details, but was considered a positive since it clearly separates the bruise from the rest of the image. The area of the bruise varies in the different principal components. PC2 gave positive results for all the images, while PC3 gave positive results for two. PC26 and 27 was only considered positive for one image. No other PCs gave similar results.

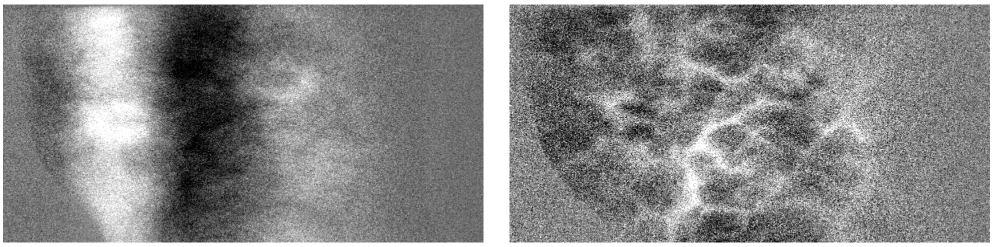


(a) I4-2 PC2 (b) I4-2 PC3 (c) I4-2 PC26 (d) I4-2 PC27

(e) I4-3 PC2 (f) I4-3 PC3 (g) I3-3 PC2

Figure 5.10: *Seven result images from using PCA on I3-3, I4-2, and I4-3. The original image used and the principal component (PC) displayed are listed below each image.*

Figure 5.11 presents two example result images that were considered to have a too low signal-to-noise ratio. The bruises are visible in (a) and weakly visible in (b). Both images were still judged to be unacceptable since both images show too much noise all across the images. The heavy stripes in (a) also contributes to the verdict. Figure 5.11(b) show possible blood vessels. General for most principal components higher than 5 is heavy to all consuming salt/pepper noise.



(a) PC7

(b) PC13

Figure 5.11: *Two result images from using PCA on I1-1. They were judged to have too low signal-to-noise ratio. The principal component (PC) displayed are listed below each image.*

5.3.2 Radiance Image Results

The radiance image results can be divided into two parts. The first covers the three radiance images that were tested to give a foundation for comparing with their reflectance counterparts. The second part covers the testing of images from I2, which was not converted to reflectance. This is a summary of the radiance results. The images and a more detailed description of them can be found in Section B.4.

The radiance images from I1 had most interesting information in PC3, 4 and 5, but some additional components had interesting features to. Bruises were clearly visible in all the result images, with minor differences to the right bruise. The right bruise looks a bit smaller, but has better contours. Two, or maybe three, different kinds of blood vessel networks can be seen. These have varying thickness and level of branching. PC4 shows hair clearly visible. PC3 and PC4 seem to contain much of the same information, but with seemingly inverted colours.

Result images from radiance image I4-3 had good results in PC3 and PC4, even though one bruise in each was slightly difficult to discern. Surprisingly, some positive results were found in PC107, 108, and 109 as well. These have bruises more visible than PC3 and 4. The bruises have different sizes in the components.

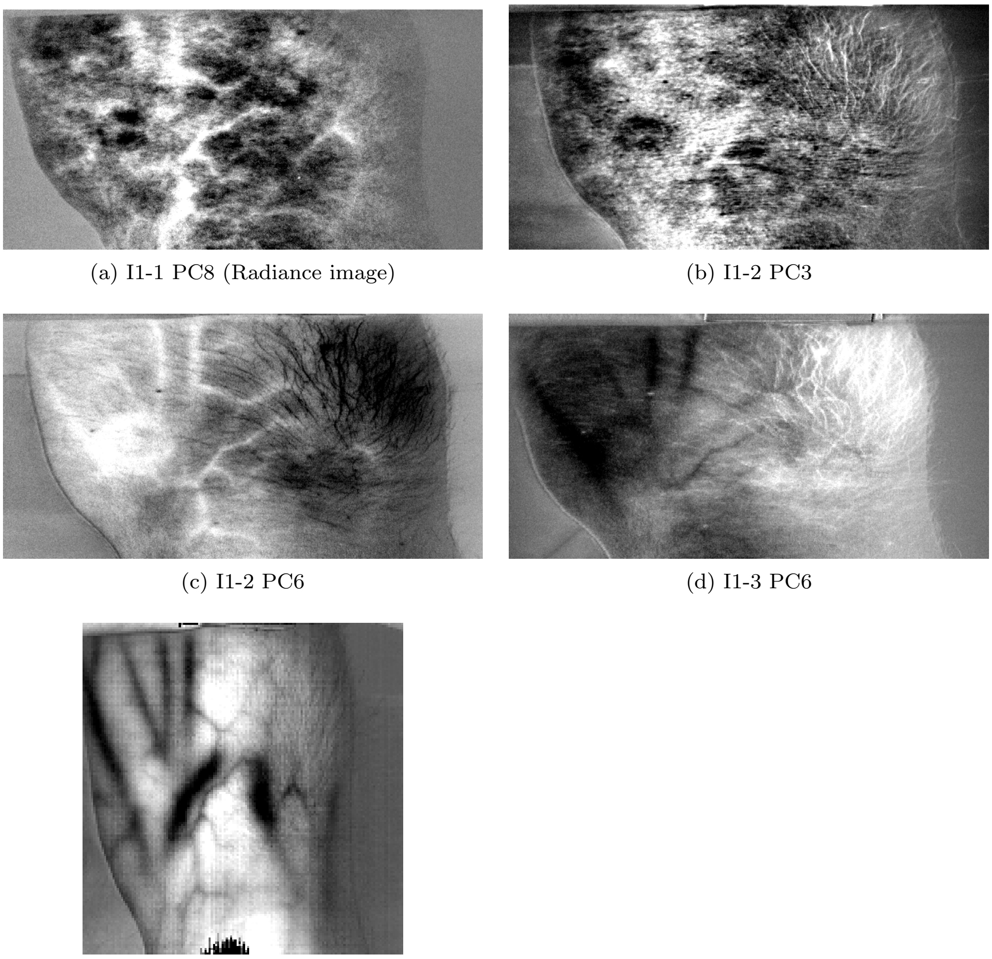
Radiance image I3-2 was also tested, but none of the principal components produced positive results. No images were included from this test.

Three radiance images were originally supposed to be tested from set I2, in a similar manner to set I1. Heavy striping effects and image corruption meant that only I2-1 was included in the report, since the other had inadequate quality. Positive results comes later compared to I1-1, and the first came at PC7. After that, results came spaced out between PC15 and PC29. The bruises are clearly visible in all the images, but the right bruise has lost some contour and is difficult to discern in most of the positive results. Blood vessels are visible in all result images, but it seems like two different networks. One has thicker vessels than the other. Hair is visible in PC7. All result images contain heavy striping effects and some corruption in the lower part.

5.3.3 Other Observations

This section presents some observations not directly related to bruise enhancement.

Figure 5.12 shows some of the other interesting results. Figures 5.12(a), (c), (d), and (e) show visible blood vessels. Hair is visible in (b), (c), and (d). Figure 5.12(b) shows possible skin surface details. It is also possible to make out a network which might be capillaries or individual skin cells. Figures 5.12(c) and (d), which are from the same PC show similar details, but with different brightness. The hair and blood vessels have inverted colours. The left bruise is clearly visible in (c) and weakly visible in (d). In (a) and (b), the left bruise is difficult to discern and no bruises are visible in (e). Some noise is present in all the images.



(a) I1-1 PC8 (Radiance image)

(b) I1-2 PC3

(c) I1-2 PC6

(d) I1-3 PC6

(e) I2-1 PC10

Figure 5.12: *Five result images showing blood vessels and hair. The original image used and the principal component (PC) displayed are listed below each image.*

5.4 K-Means Clustering

The results of the K-means clustering testing are presented in two parts. First the results from the initial testing phase are presented, and results from phase two are presented in further testing. Details of the experimentation can be found in Section 4.4.

5.4.1 Phase One - Initial Test Results

The initial testing were as described performed on reflectance image I1-1 and the results from the principal component analysis of image I1-1. The latter will be abbreviated PCA-I1-1. The results for the reflectance image will be presented first. The parameters *clusters* and *iterations* were varied. All the details on the specific variations can be found in Section A.9.1.

In all tests on both images, a variation between the initial iteration and the following iterations can be observed. The variation becomes less noticeable as the iterations increase in number as less and less pixels change clusters and the clustering approaches convergence. The initial variation and reduction of variation was expected as described in 2.3.3.

The results for reflectance image I1-1 with low cluster values (1-6) was characteristically undersegmented, but the background and foreground was clearly and comparing to the original image fairly accurate. For high cluster values (25-50), the result was an oversegmentation of the image. An example of undersegmentation can be seen in Figure 5.13(a), and oversegmentation is shown in Figure 5.13(b).

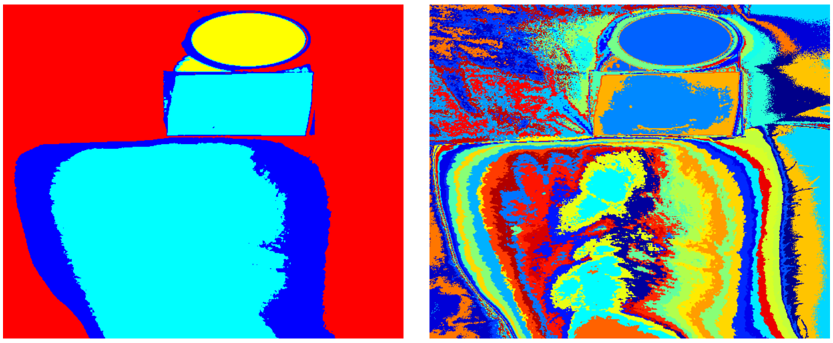
Cluster values between 8 and 16 all show an acceptable number of continuous segmented regions, but only three segmentations in this parameter range produced bruise segments for the reflectance image, and none were considered adequate for further processing. These three parameter pairs are listed in Table 5.8. A table of the remaining result details can be found in Section A.9.

Table 5.8: *The parameters that provided bruise segments to some degree on K-means clustering of reflectance image I1-1.*

Clusters	**Iterations**	**Comments**
10	1	Part of left edge of the left bruise in a single segment. Right bruise not discernable.
14	1	Part of right edge of the left bruise in a single segment. Right bruise not discernable.
16	8	Small part of left edge of the left bruise in a single segment. Right bruise not discernable.

The clearest and most complete segmentation was achieved with 40 clusters after one iteration. This was, however, an oversegmented image and the result was not considered useful. Image I1-1 at 40 clusters with 1 iteration is shown in Figure 5.14.

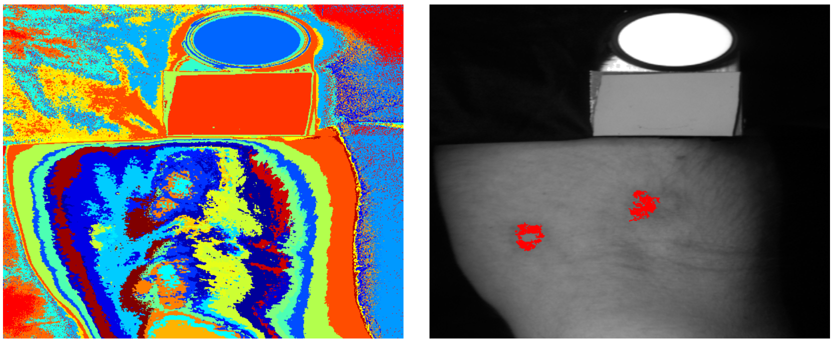
In 5.14(a) each colour represent a region. Figure 5.14(b) shows the bruise region in red overlaid the original image.



(a) Undersegmentation

(b) Oversegmentation

Figure 5.13: *Undersegmentation and oversegmentation examples for the clustering algorithm on reflectance image I1-1.*



(a) I1-1 with individual regions coloured randomly

(b) I1-1 with bruise regions overlaid on original image

Figure 5.14: *Reflectance image I1-1 oversegmentation example and overlay of bruise region on original image. Number of clusters was 40 at one iteration.*

The results for clustering on the 10 first principal components of image PCA-I1-1 were similar to the results from the original reflectance image I1-1. Undersegmentation was clear on the results from using 2 to 6 clusters. Oversegmentation dominates the results from 25 to 50 clusters. For cluster amounts from 8 to 20 the segmentation again have large continuous regions and the amount of regions seems adequate for further processing. For this input image there are six pairs of parameters within that range that resulted in partly segmented bruises. These are listed in Table 5.9. None of the results in the range separate both bruises in separate regions. In Figure 5.15 the edges of the left bruise are shown in black. This image is the result of 16 iterations with 20 clusters. The image has been edited to enhance the visibility of the region.

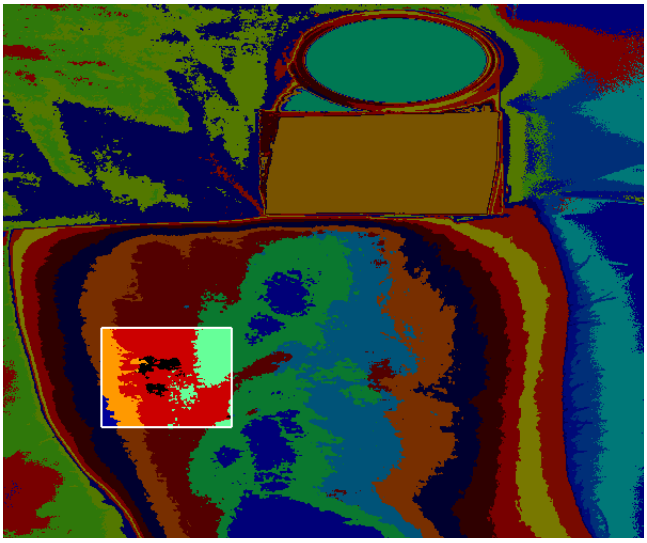
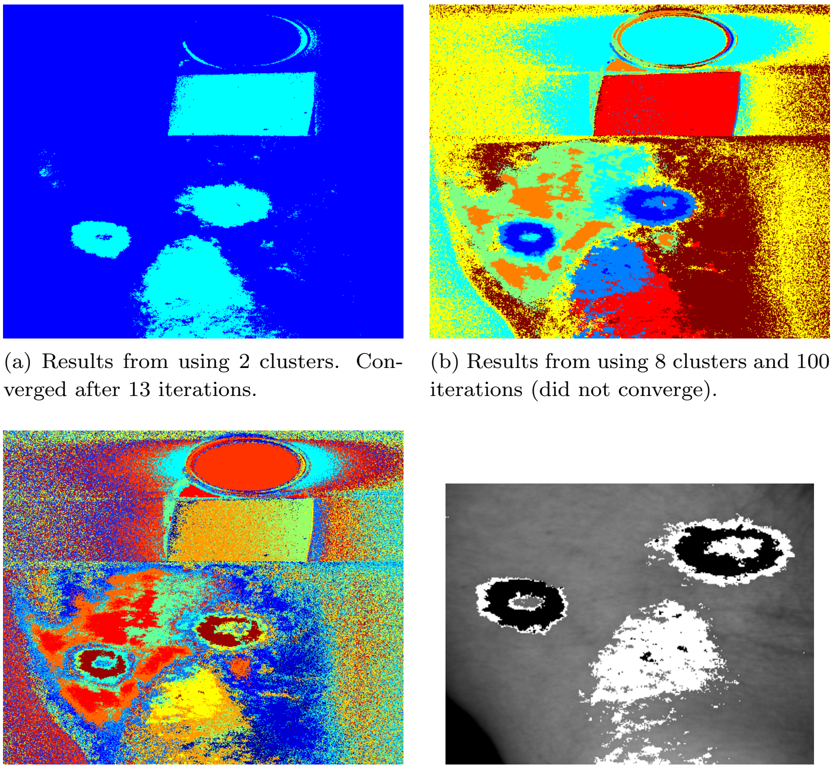


Figure 5.15: *The image shows the best selection of the left bruise with 16 iterations in 20 clusters. The image has been enhanced for focus on the region in question.*

Table 5.9: *The parameters that provided bruise segments to some degree on K-means clustering of 10 first components of PCA on reflectance image I1-1.*

Clusters	**Iterations**	**Comments**
8	4	Part of left edge of the left bruise in a single segment. Right bruise not discernable.
16	1	Part of right edge of the left bruise in a single segment. Right bruise not discernable.
20	1	Full left edge of the left bruise in a single segment. Right bruise not discernable.
20	8	Part of left edge of the left bruise in a single segment. Right bruise not discernable.
20	16	Full upper edge and part of lower edge of the left bruise in a single segment. Right bruise not discernable.
20	32	Full upper edge and part of lower edge of the left bruise in a single segment. Right bruise not discernable.

Using visual inspection of the principal components of reflectance image I1-1, the bruise was found to be predominant in the fourth and fifth components. Using this a priori information as input for the clustering algorithm, the bruises were clearly segmented with intact and single or multiple regions. Equally to testing on reflectance image I1-1 and the first 10 principal components of that image, small amounts of clusters, from 2 to 5 resulted in undersegmentation. Oversegmentation began earlier with this input data, at 20 clusters. Indications of region splits in areas apparently continuous began at 16 clusters. For all iteration values larger than 1 (beyond random starting point) and for all cluster values (2-50), the bruises are clearly segmented in few separate regions. Table 5.10 summarises the results. A cluster value of 8 was the first parameter to segment the bruises and surrounding tissue out without including the triangle shape in the lower image. The noise levels with 8 clusters were low. Figure 5.16 shows three result images from the clustering, and an overlay image showing only bruise segmentation on the original image.



(a) Results from using 2 clusters. Converged after 13 iterations.

(b) Results from using 8 clusters and 100 iterations (did not converge).

(c) Results from using 20 clusters and 16 iterations (did not converge).

(d) Overlay of bruise and tissue regions from Figure 5.16(b) on original image.

Figure 5.16: *The images shown depict the best segmentations of the bruises when clustering on a selected subset fom the principal components of reflectance image I1-1.*

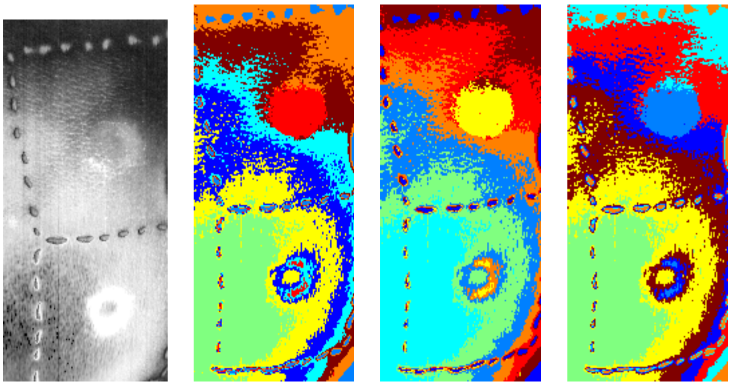
Table 5.10: *The parameters that provided full and good segmentation of the two bruises present in the image using principal components 4 and 5 of reflectance image I1-1 as input. The results are given for iteration values > 2. The quality of the results are summarised in the last column.*

Clusters	**Comments**
2-4	Undersegmented. One segment covered bruises, centre, and some surrounding tissue. The triangle region below the two bruises was marked in the same segment.
5	Undersegmented. One segment covered bruises, centre, and some surrounding tissue. Upper part of the triangle region below the two bruises was marked in the same segment.
6-7	One segment covered bruises, and one segment centre and some surrounding tissue. Points on the triangle below the two bruises were marked in the bruise segment.
8	One segment covered bruises, and one segment centre and some surrounding tissue. Points on the triangle below the two bruises were marked in the *tissue* segment, therefore differing from the last result.
10	Elevated noise levels. One segment covered bruises, and one segment centre and some surrounding tissue. Points on the triangle below the two bruises were marked in the bruise segment.
12-14	Elevated noise levels. One segment covered bruises, and one segment centre and some surrounding tissue. Points on the triangle below the two bruises were marked in the bruise segment.
16-25	Oversegmented. High noise levels. One common segment for the two bruises, one for the surrounding tissue. Points on the triangle below the two bruises were marked as a tissue segment with an unclear border.
30-50	Oversegmented. High noise levels. One common segment for the two bruises, one for the surrounding tissue. Points on the triangle below the two bruises were marked as a tissue segment with an unclear border.

5.4.2 Phase Two - Extended Test Results

From the results of the initial testing of K-means clustering, the most distinct segmentation was achieved using a relative small number of principal components from the reflectance image I1-1. The parameters that gave the clearest segmentation of the two bruises were eight clusters and more than one iteration. These parameters were therefore tested on the 10, five, and three first principal components of the reflectance image I4-3. The results can be seen in Figure 5.17. The number of iterations parameter was intentionally set very high to achieve algorithm convergence.

The three principal component images used as input for the K-means clustering algorithm all produced similar clustering results after converging; The upper bruise was marked in all results as a large contiguous region. Some spatially separate points on the lower bruise were labelled like the upper bruise. The lower bruise has a spatially separate centre region, framed by a wide circle. Inside the circle, a central part of the bruise edge forms an incomplete circle.



(a) PC3 of reflectance image I4-3 for reference.

(b) Converged clustering of the first 10 principal components.

(c) Converged clustering of the first 5 principal components.

(d) Converged clustering of the first 3 principal components.

Figure 5.17: *The images shown depict the third principal component for reference and the results of using 8 clusters and letting the algorithm converge for a number of principal components from reflectance image I4-3.*

5.5 Watershed Segmentation

The results of the watershed segmentation on the four input images are presented in this section. As described in Section 4.5, the four input grey scale images were:

(i) Band 47 of the reflectance image I1-1

(ii) The difference of band 47 and 29 from reflectance image I1-1

(iii) The ratio of band 70 and 47 from reflectance image I1-1

(iv) Principal component 4 of reflectance image I1-1

For all the input images and all threshold level values (in the range 0.0 to 1.0 with 0.05 increments), the final watershed segmentation was calculated and overlaid on the input image. The overlaid images can be seen in row four of Figure 5.18 which also shows some of the steps in the segmentation for each of the input images.

For image (i), a thresholding value of 0.3 gave the best final watershed result. Figure 5.18(a) shows the black and white thresholded image. The geodesic morphological opening with the thresholded image as a mask resulted in Figure 5.18(e). The opening removed all small and thin single features. The Beucher-gradient and following distance image are not shown in the figure. Watershed on the distance image resulted in Figure 5.18(i) and the following overlay on the original image, Figure 5.18(e). Using a threshold level of 0.3 was the only level that yielded separate regions for the bruises, seen as two circular regions covering a small area in the watershed and overlay pictures.

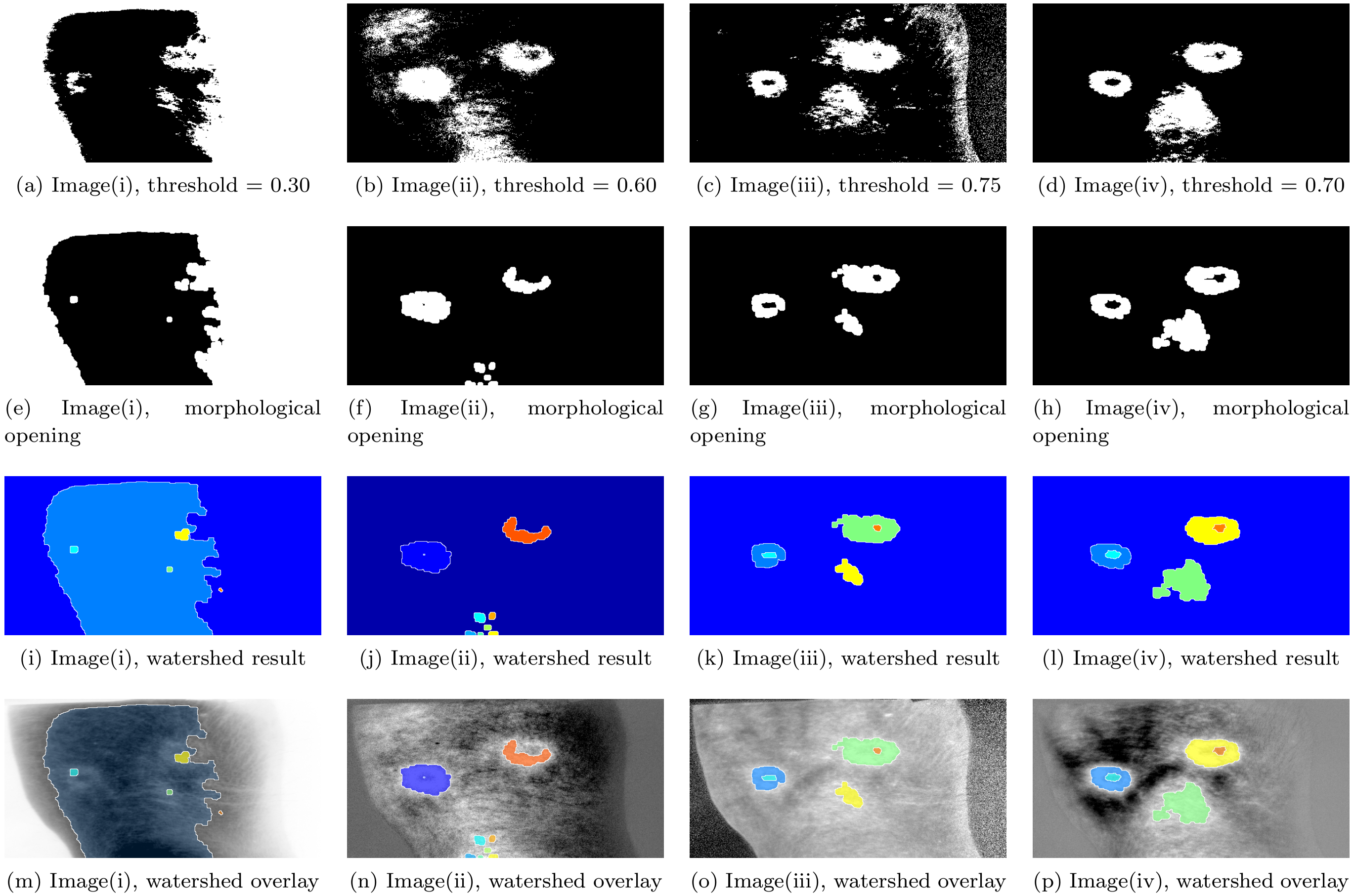
Example results for image (ii) can be seen in column two of Figure 5.18. A threshold value of 0.60 resulted in a good watershed region for the left bruise, and a partial segmentation of the right bruise. The geodesic morphological opening with the thresholded image as a mask resulted in Figure 5.18(f), and this opening removed much of the surrounding un-sharp thresholded points, and left the bruise regions, one closed section, and one open.

For image (iii), column three of the figure, a thresholding value of 0.75 provided the best separation of arm, bruise edges, and bruise centres. A third region was also segmented out, shown in yellow in Figure 5.18(k). The segmentation resulted in a slightly small region for the left bruise, and a slightly oversized region for the right bruise.

Image (iv), the last column, a threshold value of 0.70 resulted in the best bruise segmentation. The results overlay can be seen in Figure 5.18(p). The segments covered the central and surrounding parts of the bruise, and the edges seemingly covered the bruised area completely. A larger version of this image can be found in Section B.5.

Table 5.11: *The threshold values that gave the best watershed segmentation for each input image.*

Image	**Value**
(i)	0.30
(ii)	0.60
(iii)	0.75
(iv)	0.70



(a) Image(i), threshold = 0.30

(b) Image(ii), threshold = 0.60

(c) Image(iii), threshold = 0.75

(d) Image(iv), threshold = 0.70

(e) Image(i), morphological opening

(f) Image(ii), morphological opening

(g) Image(iii), morphological opening

(h) Image(iv), morphological opening

(i) Image(i), watershed result

(j) Image(ii), watershed result

(k) Image(iii), watershed result

(l) Image(iv), watershed result

(m) Image(i), watershed overlay

(n) Image(ii), watershed overlay

(o) Image(iii), watershed overlay

(p) Image(iv), watershed overlay

Figure 5.18: *Steps, results, and overlays for watershed segmentation on four input images. Each row show the same step for each image, and each column show all the steps for a single image.*

Chapter 6

Discussion

The difference, ratio, and PCA algorithms are capable of enhancing bruise visibility. The enhancements can prove valuable in visual inspection and classification. Given good parameters, the enhancement algorithms clustering and watershed can provide very good results for further segmentation of image features. This chapter discusses strengths and weaknesses of the project and algorithms and provides a brief comparison between the methods. Other interesting observations are presented at the end of the chapter.

For the sake of brevity, this chapter sometimes refers to difference, ratio, and principal component analysis as the 'three algorithms'.

6.1 Validation

This section presents a validation of the project and the experimentation. It is based on the validity evaluation described in Wohlin et al. [37]. Only the topics that were deemed most relevant for this project have been included and discussed. The two main areas covered are conclusion validity and construct validity. Conclusion validity is concerned with the relationship between the treatment and the outcome, while construct validity is concerned with the relation between theory and observation. The validity threats for this project are as follows:

- **Low statistical power:** The power of a statistical test is the ability of the test to reveal a true pattern in the data. A low statistical power increases the risk for an erroneous conclusion and makes it difficult to reject an erroneous hypothesis

- **Fishing:** Searching for specific results are considered a threat, since the analyses are no longer considered independent and the researchers may influence results by looking for specific outcomes

- **Reliability of measures:** The validity of an experiment is highly dependent on the reliability of the measures. The basic principle is that when you measure a phenomenon twice, the outcome shall be the same. Objective measures are more reliable than subjective ones

- **Reliability of treatment implementation:** There is a risk that the implementation is not similar between different subjects or different occasions. The implementation should be as standard as possible

- **Mono-method bias:** Using a single type of measures involves a risk that if measure gives a measurement bias, then the experiment will be misleading. Different types of measures can be cross-checked against each other

The last validity threat relates to construct validity, while the rest relates to conclusion validity. The validity threats are presented in the same order as listed.

The data set is considered to yield a low statistical power for prediction of future algorithm results. This is because of the low number of independent subjects and images that was tested. To remedy this, all the images in time series could have been used to increase the number of images, but this was not done. The similarities between the subject and the injuries would not increase the statistical power related to general bruise detection on humans. It would give more information on time sensitivity for the algorithms, but this was not the focus of this project. New images with a greater variance in bruise shape and colour were captured to increase the statistical power for this thesis. Unfortunately, several factors that were beyond the control of this thesis led to the exclusion of all new images from the test set. These factors are described in Section 3.1.3. Because of the actual size of the data set, conclusions must be treated more as indications or guidelines for what might be successful in later work. It is recommended to extend the data set for further work and to include more variation with regard to bruise type and skin colour.

Fishing could influence results because of the subjective nature of visual inspection and because of knowledge available to the observers. The observations were done by the writers and it was known in advance if an image contained a visual bruise or not. This could for example lead an observer to see bruises that an independent observer would be unable to identify. To reduce this validity threat, both writers inspected the initial result images, while two independent observers were used on selected images. Future work related to visual enhancement of bruises could conduct larger tests using external observers to reduce this threat further and as a success measurement.

Visual inspection is not known to be a reliable measure because of its subjective and therefore error prone nature. Result images that were deemed positive for one observer might be classified as negative by another. As was mentioned in the previous paragraph, two observers were used on all the result images and two independent observers were used on a selected number of result images. This increase the reliability of the measurements but there is room for improvement. A larger observer group would improve this measurement still. The reliability of treatment implementation is considered to be a validation threat to this project. This is because of the many cameras, light sources, and environments used during image acquisition. It is believed that these factors can be considered as variables in the experimentation, and therefore might influence results. In previous work related to bruise detection on apples and cucumbers, these factors were kept the same for all images [2, 20, 39, 15]. Two DC 150W halogen light sources, situated at different angels compared to the object of interest, were used in the previous work on apples and fruits. For the most part, only a single light source or two with almost identical angel were used on the images in the available data set. It is believed that the setup that uses two light sources at different angels will reduce shadow effects and give better images. One type of light source using the setup described in previous work on apples and cucumbers, along with one type of camera and one environment is believed to be the best basis for further work on bruise enhancement and segmentation on humans.

Visual analysis was the only measurement used in this thesis which increases the validation threat of mono-method bias. If other types of measurements had been done, they could have been cross-checked against the visual inspection measurements. The low statistical power provided by the available data set means that most results only can be treated as indications. Alternative measurements, objective or subjective, would not change this fact. A larger data set may make this validation threat more important to address.

6.2 Individual Algorithm Evaluation

In this section, the algorithms and their results are discussed individually.

6.2.1 Difference

A higher percentage of good results were observed for approach one compared to approach two during the initial testing (phase one) for image I1-1. Approach one is band selection using absorption peaks, while approach two is band selection through visual inspection of the image cube. The higher percentage can probably be explained by the bands used in the first approach, since they are based on absorption peaks of important compounds. Some of these contribute to bruise visibility. Related work has been done on cucumbers, where absorption peaks for known compounds were successfully used to enhance surface damage [2, 15]. This corresponds with the higher number of positive results from approach one. Even though the percentage of positive results using approach one is higher than that of approach two, it is still considered to be a low value. More exhaustive experimentation should be carried out on a bigger image test set. This is expected to alter the percentages in favour of using absorption peaks (approach one).

For tests of the difference algorithm on image I2-1, a single positive enhancement was observed. The two specific bands input to the algorithm were identified by visual inspection. A different choice of bands might increase the percentage of positive results. Future work could include more extensive tests involving alternative bands.

The following parameters are recommended for future studies. They represent the set of wavelengths that provided the best results across the images tested.

- Bilirubin (480 nm) and oxyheamoglobin (542 nm)
- Oxyheamoglobin (576 nm) and metheamoglobin (508 nm)

The parameters listed gave the best enhancement results for image I1-1 in phase one, but mixed negative and positive results for the rest of the images in phase two. I1-2 and I1-3 both show bruises, but they become difficult to discern from surrounding tissue as the time after injury increases. Result images from I3-3 and I4-2 show no significant improvement over the original input, and in some cases visibility deterioration can be observed. Results for image I4-3 show bruises as well as the original, however the contours are better for the upper bruise.

The bands (a single parameter pair) found using visual inspection of bands (approach two) provided marginally better bruise visibility compared to the original I2 images. Both images resulted in a positive enhancement of the bruise features. This resulted in a 100% positive rate for I2-2 and I2-3. This is not expected to hold for a test on larger data set. Only two images were tested using this one parameter. In total, this gives a very low statistical power for predicting further positive results.

A reversal of parameters in the difference algorithm produces the mathematical inverse. A small test was performed during experimentation which confirmed that this holds for the implemented algorithm.

An interesting result is shown in Figures 5.3(f) and 5.3(i). In the original bands, the upper bruise is shown as dark with a faint, light internal ring. The lower, smaller bruise is dark with a clear white centre. In both figures mentioned, the bruises have clear centres and a clear ring surrounding the bruise. The central spot most likely covers the area of the most severely damaged tissue. This is in accordance with Randeberg et al. [26, 27].

6.2.2 Ratio

As was observed with the difference algorithm, a higher percentage of positive results were observed for approach one compared to approach two during the initial testing (phase one). This was as expected since a priori knowledge of important compounds were used to identify bands in the first approach. This seems to be consistent with work done on cucumbers where specific absorption peaks corresponding to different compounds found in the object of interest were used [2, 15]. A different selection of bands or a larger test set for approach two might result in higher percentage, but it will most likely only match approach one at best.

No positive results were observed for image I2-1 using specific bands selected by visual inspection. The selection of alternative bands could give positive results. A more exhaustive study involving different image bands is needed before any conclusions can be drawn.

The reversal of parameters for the ratio algorithm was expected to result in approximate inverse images, or more accurately an enhancement of the same features in the opposite intensity direction. A small test was carried out to visually confirm the assumption. As can be seen by the example in Section B.3.3, the dark features of the first image (left) are similar to light features of the second image (right) and vice versa. As the results were found to be similar for the reversed parameters, it was possible to half the amount of parameters that were to be tested. The image inverse is easy calculated later if needed. However, this appearance of inversion is not a mathematical inverse, as it is for the difference algorithm.

The combination of wavelength 576 nm and 760 nm gave result images with similarities to the original bands for images I3-3, I4-2, and I4-3. This can be seen by comparing column one and column four in Figure 5.7. A more interesting result was obtained for I1-1 during the initial testing with these parameters. As for the mentioned images, there are some similarities between the original image and the result image. A reversal of the parameters would most likely give a result image with bright bruises on a dark arm, probably similar to that of Figure 5.7(o). Because of this, it might be possible to remove the parameters used from future tests, as it would probably give similar results to Figure 5.7(n) with the parameter order reversed. It could be argued that the parameters applied to 5.7(n) should be removed from future testing instead, but these are based on absorption peaks and the parameters used on image 5.7(o) are based on visual inspection.

By comparing the result images from I1, it is possible to see how the different parameters respond to bruise ageing. Using the wavelengths 480 nm and 542 nm and the wavelengths 576 nm and 508 on image I1-3, it was difficult to discern any signs of the bruises. This is probably related to the choice of wavelengths, which corresponds to the absorption peaks of bilirubin, oxyheamoglobin, and metheamoglobin. Both the concentration of bilirubin and the oxygenation in a bruise decrease with time [28]. This is probably why the bruises go from clearly visible in I1-1 to almost gone in I1-3. Influence from metheamoglobin is probably the reason that the bruises show slightly better using the last two wavelengths.

With this in mind, the following parameters are recommended if the ratio algorithm is used in future work:

- Bilirubin (480 nm) and oxyheamoglobin (542 nm)
- Oxyheamoglobin (576 nm) and metheamoglobin (508 nm)
- Deoxyheamoglobin (760 nm) and oxyheamoglobin (576 nm)

However, if bruises are recorded over longer periods, for example more than 192 hours after injury, or if a bruise is old, it seems like the last set of wavelengths would give the best results. The ratio between deoxyheamoglobin and oxyheamoglobin has been applied to human bruises before, although using the lower absorption peak for deoxyheamoglobin (555 nm) [26].

An interesting observation can be made in Figure 5.7(m). The image shows a noticeably brighter ring inside a dark bruise region. The bruise centre is also dark. This ring follows the contours of the most severely damaged skin tissue, where the point of impact caused vessel damage [26]. The bruise features can be extracted and used to describe and identify the impacting object. This could be a useful tool in forensic science or in a diagnostical application.

6.2.3 Principal Component Analysis

The principal component analysis (PCA) produced some good bruise enhancements for visual inspection, and as an input image for further segmentation. The results are highly dependent on which principal component (PC) is chosen. PC1 never produced positive results in the conducted tests. This might be because the bruise was treated as noise and because PC1 can be considered as a noise-reduced 'mean' image [6]. Clear, visible bruises were predominantly found in PC2 to PC5, although not consistently for all images. This seems to be in accordance with previous experiments were PC3 and a combination of PC2 and PC3 were used as foundation for bruise detection on apples [39, 20]. Bruises and chilling injuries on cucumbers were found in PC1 [2, 15]. A clear exception to this pattern can be seen in PC107, PC108, and PC109 from PCA on radiance image I4-3. Even though these components retain very low variance, they still showed clearly visible bruises. This reflects the importance of visually inspecting all principal components using one's knowledge of the study area when selecting principal components[17]. For bruise detection and enhancement this involves looking for principal components where the bruise is clearly visible. Using the magnitudes of the eigenvalues as an indicator of information content might not produce satisfactory results.

It was not possible to identify a single principal component which produced positive results for all the images tested. A possible reason for this could be the changing light sources, different capture devices, different subjects (human and pig), and different environments. In related work, when a principal component was chosen for bruise detection on apples and cucumbers, the same camera and light setting were used in a controlled environment for all the captured images [15, 2, 20, 39]. The subjects under inspection were also quite similar, being either apples or cucumbers. There were some variations in apple skin colour. Literature confirms that the properties of a PCA model often is problem-dependent [6]. Control of these parameters is important and desirable for a future system. It should also be noted that fruit have less complex skin composition compared to humans (crosstalk and scattering) and this could impact robustness with regard to choice of principal component. Taking these limitations into account might imply that PCA is not well suited for automated bruise enhancement and detection in the field. Further testing on a wider more controlled set should be carried out.

A bruise might turn out bright or dark in a principal component depending on the direction of the eigenvector. Figures 5.12(c) and (d) show two result images displaying PC6 from image I1-2 and I1-3 respectively. Care should be taken if simple operations like thresholding are used in automated systems, since bruises might end up below or above any pre set threshold. User input might be required.

The colour of a bruise is not only dependent on the heme catabolism, but also the depth of injury [3]. Contents found in the different principal components might suggest that a non-invasive approach using PCA could record depth information of various features. Some principal components show blood vessels, while others show bruises. Bruises and blood vessels exist at different depths, and this could possibly be seen in different PCs. PCs that show smaller or enlarged extents of a bruise could also be considered. The PCs that show both bruises and blood vessels in the same principal component, at the same time could suggest that this is not possible. Further work is required before any conclusions can be drawn. In vitro studies involving tissue phantoms with known structures could be used to investigate the relation between PCA components and depth resolution. This approach might have several applications within e.g. dermatology or plastic surgery.

Comparing principal components of reflectance images and radiance images, it is easy to see that they contain different information. This is a natural consequence of the adjustments done to the reflectance image compared to a radiance image. The radiance image seems to show certain blood vessel networks that are not visible in the reflectance image (I1-1), and seemingly more components contain useful information. This can perhaps be reflected by the slightly higher percentages (30,9% compared to 24,5%) of positive results found in the radiance images. These percentages should be considered with some care, as they are based on a visual analysis with bruise visibility in mind and the number of images is considered low (4 radiance and 9 reflectance images). If reflectance images contain interesting information in fewer components, they are more suited for data reduction. For bruise enhancement and detection it seems like both radiance and reflectance images could be used giving different results.

6.2.4 K-means Clustering

The results for the K-means clustering algorithm are presented in Section 5.4. This section will first discuss the clustering parameters, and then the results are compared.

A significant difference was observed between the first and following iterations on all input images. This corresponds with the theoretical background and implementation details, as MATLAB's clustering method uses random starting centroids. As the number of iterations increased, and thus approached stability and convergence, the segmentations changed decreasingly each iteration. Because fewer points are reassigned per iteration in clustering, there is reason to believe that results with medium to high clustering numbers are accurate enough for visual determination of good results. For further development it should be considered to let the algorithm always converge. This would also enable reproducible results, but would introduce a significant increase in algorithm run time.

As clustering is a per-pixel algorithm, it was expected and shown that high values of clustering, generally more than 20 clusters, resulted in many small regions, appearing as noise, termed oversegmentation. Equally, undersegmentation gave good information on the general shape of the foreground and background, but failed to provide any relevant information on the smaller bruise features.

For the reflectance image I1-1, no results were considered adequate, and very few results showed fragments of bruises. The three best results are listed in Table 5.8.

For 10 and 14 clusters with one iteration, the positive result can be classified as a coincidence. The results appeared only for the first iteration, and following iterations showed no bruise segments.

The most reliable result for reflectance image I1-1 was found with 16 clusters and 8 iterations, the maximum number of iterations tested for this amount of clusters. The results show only parts of the left bruise, and no discernable features of the right bruise. No pair or range of parameters was considered valid for further testing on reflectance image I4-3. The bruises were not predominant enough in the images to simply be labelled as the same region.

The clustering results, for the valid range of cluster numbers, all grouped bruises and surrounding tissue. In earth observation by hyperspectral imaging, clustering has been used successfully on reflectance images [35, 17]. Unlike previous known usage on vegetation and inorganic materials, the image point vectors on a bruised forearm are spectrally very similar to each other. This similarity between image points is believed to be the cause of the inadequate clustering results for the reflectance images.

The results from the clustering algorithm on the first ten principal components of image I1-1 gave few good positive results. The results are presented in Table 5.9. As written in the above paragraph, a iteration value of *one* hold little credibility, as the starting centroid positions are random, and therefore the results are not reproducible. As none of the results for the first ten principal components showed the right bruise, the test had no usable resulting parameters. The bruise was clearly visible in 2 of the 10 components, and it might be reasoned that the parameter range tested did not cover values that could result in a good segmentation. The results from this test can therefore not be considered conclusive.

Using principal components *four* and *five* from reflectance image I1-1 as a two banded input image, the resulting segmentations were significantly better. For a wide range of cluster values, including values that gave over- and undersegmentation, the bruises and surrounding tissue were extracted. The high contrast regions of the bruises in all of the input image bands are the reason for this successful segmentation. The best result of all clustering parameters was the result from using 8 clusters and the default of 100 iterations. The lower triangle shape seen in the two principal component images did not end up in the same segment as the bruises, but were grouped with the immediate surrounding bruise tissue. In Figures 5.16(b) and 5.16(d) the quite accurate segmentation is shown.

The results indicated that 8 clusters were a sensible cluster count, with a high number of iterations, thus approaching clustering convergence. Testing these values on the *ten, five*, and *three* first principal components of image I4-3 resulted in good segmentations of the bruises for all three inputs. This is an indication that the first principal components contained enough bruise features enabling a good segmentation. For reflectance images, choosing the three or five first principal components could give good results, but this requires further testing. Automatic settings of the parameters is not recognised as a good solution [35], and trial and error testing for each application is considered a good solution.

A clear problem using the principal component approach is the selection of principal components with high contrast features. There was never an intention of using the algorithm unsupervised, but keeping a low number of parameters is believed to have resulted in a more thorough experimentation for the relatively few inputs. The minimum parameters are a sensible number of clusters and a suitable data subset. The algorithm iterations are considered less important as the algorithm should be allowed to converge. Based on the testing, the number of clusters may need a per-image setting.

As mentioned, the clustering algorithm facilitates segmentation, or grouping, and also has a possible usage in substance identification depending on the input data [35]. Clustering can provide a understanding of relationships that may exist between image points. A clustering method targeted at clustering spectroscopic data for tissue analysis could be developed and tested in future work. A spectral and spatial signal smoothing could help reduce the observed noise in the clustering results[39], providing a better spatial grouping, but may also influence the spectral identification process to an unacceptable degree.

6.2.5 Watershed Segmentation

The results from the watershed segmentation experiments show that using input images with low bruise contrast will result in segmentations that can be characterised as incomplete, not covering, or too large. This is the case for the (i) input band 47 and the (ii) difference of band 47 and 29 from reflectance image I1-1. The first provided a minimal coverage of the bruises at a thresholding value of 0.30, and the latter segmentation was considered incomplete as it only partly covered the bruises.

The (iii) ratio of band 70 and 47 and (iv) principal component 4 of reflectance image I1-1 were the two inputs with the highest bruise feature contrast, and provided the best two segmentations. They both covered the bruise, had separate centres, and a small separate region for the lower triangle region. This indicates that higher feature contrast directly leads to a better segmentation result. A less naive thresholding approach could be used preparing the input for watershedding, for example local adaptive thresholding. The possibility for other algorithms that could provide a parameterless implementation should be investigated.

6.3 Comparisons and General Observations

A comparison of the algorithms, and observations relating to more than one algorithm, are discussed in this section. Some of the topics covered are enhancement of already visible bruises, detection sensitivity related to age of bruises, and possible bilirubin presence, dubbed the 'racoon' effect.

It is important to understand that operations on physical properties, such as wavelength, are dependent on simple connections to produce for example good bruise enhancements. Enhancement methods that use simple arithmetic's (add, subtract, divide, and multiply) on physical properties can not be expected to enhance beyond the simple connections in the data. Methods that work on latent statistical properties of the physical variables can bring forward and enhance variables, connections, and statistical patterns in the data [4, 6]. PCA is the only tested enhancement algorithm based on statistical principles. PCA provided the best improvements for further segmentation. However, a visual inspection was needed to identify the principal components that showed an enhancement. The PCA is as mentioned therefore considered problem-dependent [6].

Positive results were found using ratio, difference, and PCA on original images that contained clearly visible bruises. For the image I3-3, the enhancement algorithms gave no positive results. This image contains visible damage from a square pendulum. The damage can be characterised as a bruise, however this bruise is more difficult to see compared to the circular bruises found in e.g. image I4-3. This damage was caused by high velocity paintballs, and the difference can most likely be attributed to the difference in mass and speed of the impacting object. No other images with weak or undetermined bruising resulted in improved bruise visibility.

One possible exception was observed. The triangle shown in Figures 5.5(c) for ratio and 5.9(a) for PCA, is possibly a bruise. The cause of this possible bruising is unknown, and it could have been inflicted prior to the experiment. As there were no other observations that indicated possible bruising, and the visibility and contrast of the bruise in I3-3 were not enhanced, the algorithms ratio, difference, and PCA only enhanced already visible bruises. The two injury causes (paintballs and square pendulum) were the only two represented in the data set, and thus represented in the experimentation. Additional types of bruising with different haemorrhage depths, shapes, and sizes could help establish a possible relationship between the enhancement capabilities of the algorithms and type of bruise. Further research might define a clear connection.

Seemingly, the algorithms ratio, difference, and PCA, with the following watershed segmentation and partly clustering are sensitive to ageing of bruises. When a bruise occurs, the immune system responds with an inflammatory reaction. Red blood cells and free haemoglobin molecules are engulfed and the heme oxygenase system starts to break down the haemoglobin and produce bilirubin and haemosiderin [26]. In short, this gives changing colours over time until the damage is repaired and the bruise is no longer visible. If ratio, difference, and PCA are influenced by the initial visibility of a bruise, they would also be affected by the diminishing visibility observed as a bruise ages. This seems to correspond with result images observed during experimentation. Result images from I1-1 show clearly visible bruises for all three algorithms. The bruises are less visible in the result images from I1-2 and clearly weaker in the result images from I1-3. Previous work on cucumbers also shows that the detection accuracy for ratio, difference, and PCA was affected by time passed after bruising [2, 15]. The visibility reduction can not be seen for other features such as blood vessels in the ratio result images. This could be yet another indication that ratio, difference, and partly PCA are only able to enhance already visible bruises. The sensitivity to bruise age might also indicate that the algorithms are unsuited for detection of older, healed injuries.

When the results from this thesis are compared with previous work related to cucumbers and apples, it is important to keep in mind that there are important differences between the bruised subjects. Humans tend to vary more than for example apples, even though both can display different skin colours. The human skin and underlying tissue is more complex than that of apples. Effects like scattering and cross talk can probably affect results on humans to a greater degree than on apples (if any).

There are some interesting observations to be made when comparing result images from I1. The 'racoon' effect seen in Figure 5.1(b) and in Figure 5.5(a) might be attributed to bilirubin, which give bruises a yellowish colour. If a bruise has presence of bilirubin it can be an indication of the age of a bruise [12]. The 'racoon' effect might be useful in further work to determine bruise age by showing existence and extent of bilirubin presence. If a series of images are captured it could also be used to determine diffusivity of various compounds in skin. The 'racoon' effect can be observed in result images from I1-1 (72 hours after injury) but it is practically gone in the result image from I1-2 (192 hours after injury) using the same parameters. By observing the rate of decline of concentration and extent, it could be possible to determine diffusivity. However, according to Randeberg et al. [28] the bilirubin concentration should not be much lower after 192 hours compared to 72 hours. The concentration was found to increase up to about 168 hours (seven days) and then decrease after that, but it did not disappear completely after 192 hours. After 397 hours it would probably be significantly lower and not show very well. This last observation seems to correspond better with the result images. The discrepancy for I1-2 can be explained by individual variations in the subjects.

Previous work related to bruise detection on cucumbers and apples have among other things compared ratio, difference, and PCA. As mentioned above, it seems like the three algorithms are sensitive to the ageing of bruises. In Ariana et al. [2] ratio and difference was preferred over PCA, because their classification accuracy was more stable over time. Ratio also showed a better accuracy compared to difference during the first couple of days, but that changed on day three. Other work within the same field of research shows that the detection accuracy of ratio and PCA are comparable, but ratio showed slightly better discrimination rate during the two first days [15]. The experimentation in this report does not contain enough samples to verify any of these results, but some interesting observations were done. The left bruise in I1-3 can be seen in some of the result images from the three algorithms, while the right bruise is difficult to discern in all the result images from I1-3. This could indicate that the three algorithms have similar detection potential for human bruises, but a broader data set is needed before anything conclusive can be said.

The difference algorithm was the only one that gave positive results for all the images in I2. Therefore it is not possible to compare result images with the two other algorithms. Result images from the difference algorithm show that both bruises can be enhanced, but they are difficult to discern in the result image from I2-3. A better light source could provide better PCA results and alternative bands could give better results for the ratio algorithm. Further work is needed to investigate these possibilities.

Ratio and difference is much faster than PCA and provides similar results for some parameters. They are also easier to implement. This is not unexpected and corresponds with results from work related to bruises and chilling injuries on cucumbers [2, 15]

According to the percentages showing positive results in Chapter 5, it seems like the ratio algorithm is better than the difference algorithm during phase one. For phase two it looks like it outperforms both difference and PCA. Care must be taken while interpreting these numbers however. They are not meant as a performance measurement, but a summary of the tests that were done on each of them. For ratio and difference the percentages show how many positive results were observed during each phase. The percentages for PCA show how many principal components with an acceptable signal-to-noise ratio contained visible bruises. A direct comparison is therefore not applicable, but some interesting information can still be gathered. The enhancement quality is not directly reflected through the percentages, but they give some indications on how many positive results can be found through visual analysis of certain parameters. Based on the parameters tested, there are more positive results for ratio than for instance the difference algorithm. For this thesis, however, it is more important with one really good result, than many just acceptable ones. Based on this criterion, all the algorithms show potential. When the result images from ratio, difference, and PCA was used further by clustering and watershed, PCA seemed to provide a better base for segmentation. For visual enhancement of bruises the three algorithms all gave some satisfactory result images. To get a performance measurement, an extended test using suited input parameters for clustering and watershed should be done. A larger data set and an automated system for bruise detection and classification would be beneficial. The algorithms tested in this thesis may provide important components in such a system.

The enhancement results used for segmentation experimentation were chosen based on bruise visibility. This is also the case for the band from the reflectance image I1-1 tested with both K-means clustering and the watershed segmentation algorithm. Both methods for segmentation provided several good bruise segmentations on the three enhanced images given a wide range of parameters. However, used directly on the reflectance image, neither clustering nor watershed segmentation provided good results.

The best result from clustering is directly comparable to the best results of the watershed segmentation, as can be seen in Figures 5.16(d) and 5.18(p). The common trait for these two images is the prior use of PCA for enhancement. The grey scale (two dimensional) results from ratio and difference were not used on the multidimensional clustering algorithm, but when tested with the watershed segmentation, both provided segmentations superior to those of the original reflectance image.

6.4 Other Observations

Difference, ratio, and PCA all produced result images with content that might provide valuable information for later research. This research might not necessarily be related to bruises. Two topics are covered in this section; blood vessels and hair.

All three algorithms gave result images which emphasized blood vessels for images from I1 and from I2. Blood vessels of varying thickness can be observed in Figures 5.12(c) and (e). The varying thickness probably means that the algorithms can show different kinds of blood vessels like veins and arterioles. These blood vessels exist at separate depths, which is another indication that depth information could be found using the three algorithms. The images showing blood vessels could be used to map the layout of for example arteriole networks before surgery.

Hair is observed using all three algorithms as well. The bright hair in Figure 5.12(d) is one such example. Since the hair has a clearly different colour compared to the arm, it would be fairly easy to separate the hair using for example thresholding. With some additional work, this could be useful to, for example, inspect results after a hair removal treatment. It could be used as a measurement for success by comparing results before and after or to detect regorwing hair. Treating the same area sequentially as hairs begin to regrow increase the number that are permanently removed [36].

Chapter 7
Conclusion and Further Work

Hyperspectral imaging has in this thesis proven to be a useful technology platform for a more accurate characterisation of bruises on human skin. The tested algorithms provided a better description of the bruises, the extent of the bruising, and the severity of damage. However, the algorithms tested are not considered robust for consistency of results. It is therefore recommended that the image acquisition setup is standardised for all future hyperspectral images. A larger, more varied data set would increase the statistical power of the results, and improve test conclusion validity.

Results indicate that the ratio, difference, and principal component analysis (PCA) algorithms can enhance bruise visibility for visual analysis. However, images that contained weakly visible bruises did not show significant improvements in bruise visibility. Non-visible bruises were not made visible using the enhancement algorithms. Future work may include the development or use of enhancement algorithms that find and improve latent statistical relationships and connections present in the data.

Results from the enhancement algorithms were segmented and compared to segmentations of the original reflectance images. The enhancement algorithms provided results that gave more accurate bruise regions using K-means clustering and the watershed segmentation. Both segmentation algorithms gave the overall best results using principal components as input. Watershed provided less accurate segmentations of the input from the difference and ratio algorithms.

The enhancement algorithms provided interesting results that should be considered in future studies. Skin structures like fine lines, blood vessels and hair density were visualised with good contrast compared to the original images. The distribution of skin chromophores like bilirubin or oxyhemoglobin was also clearly visualised using the present methods. These features might have several diagnostic applications within medicine, e.g. in diagnosis and treatment monitoring of vascular skin disorders.

Bibliography

[1] H. Abdi. Partial least squares (pls) regression. *SAGE Encyclopedia of Social Sciences Research Methods*, 2003.

[2] D. P. Ariana, R. Lua, and D. E. Guyer. Near-infrared hyperspectral reflectance imaging for detection of bruises on pickling cucumbers. *Comput Electron Agr*, 53:60–70, 2006.

[3] M. Bohnert, R. Baumgartner, and S. Pollak. Spectrophotometric evaluation of the colour of intra and subcutaneous bruises. *Int J Legal Med*, 113:343–348, 2000.

[4] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* Wiley, New York, 2001.

[5] A. A. Faust, R. H. Chesney, Y. Das, J. E. McFee, and K. L. Russell. Canadian teleoperated landmine detection systems. part ii: Antipersonnel landmine detection. *Int J Syst Sci*, 36(9):529–543, 2005.

[6] P. Geladi and H. Grahn. *Multivariate Image Analysis.* Wiley, Chichester, 1996.

[7] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Prentice-Hall, New Jersey, 2002.

[8] Intel. Open source computer vision library. `http://www.intel.com/technology/computing/opencv/`, retrieved 27.02.07.

[9] ITTVIS. Envi - the remote sensing exploitation platform. `http://www.ittvis.com/envi/`, retrieved 05.04.07.

[10] E. Kreyszig. *Advanced engineering mathematics.* Wiley, New York, 1999.

[11] D. S. Krolls. Wikipedia. `http://en.wikipedia.org/wiki/Image:Spectrum4websiteEval.png`, retrieved 25.04.07.

[12] N. E. I. Langlois and G. A. Gresham. The ageing of bruises: A review and study of the colour changes with time. *Forensic Sci Int*, 50(2):227–238, 1991.

[13] S. C. Liew. Principles of remote sensing. `http://www.crisp.nus.edu.sg/~research/tutorial/image.htm`, retrieved 04.04.07.

[14] Y. Liu, Y. Chen, C. Wang, D. Chan, and M. Kim. Development of hyperspectral imaging technique for the detection of chilling injury in cucumbers. *Proc SPIE Int Soc Opt Eng*, 5587:18 – 28, 2004.

[15] Y. Liu, Y. Chen, C. Y. Wang, D. E. Chan, and M. S. Kim. Development of a simple algorithm for the detection of chilling injury in cucumbers from visible/near-infrared hyperspectral imaging. *Appl Spectrosc*, 59(1):78–85, 2005.

[16] D. B. Malkoff and W. R. Oliver. Hyperspectral imaging applied to forensic medicine. *Proc. SPIE Int. Soc. Opt. Eng.*, 3920:108–116, 2000.

[17] P. M. Mather. *Computer Processing of Remotely-Sensed Images: An Introduction.* Wiley, Chichester, 2004.

[18] Mathworks. Matlab - the language of technical computing. `http://www.mathworks.com/products/matlab/`, retrieved 18.04.07.

[19] P. M. Mehl, K. Chao, M. Kim, and Y. R. Chen. Detection of contamination on selected apple cultivars using reflectance hyperspectral and multispectral analysis. *Proc SPIE Int Soc Opt Eng*, 4206:201–213, 2001.

[20] P. M. Mehl, Y. Chen, M. S. Kim, and D. E. Chan. Development of hyperspectral imaging technique for the detection of apple surface defects and contaminations. *J Food Eng*, 61(1):67–81, 2004.

[21] Microsoft. Comparison of 32-bit and 64-bit memory architecture. `http://support.microsoft.com/kb/294418/`, retrieved 07.05.07.

[22] Microsoft. Ram, virtual memory, pagefile and all that stuff. `http://support.microsoft.com/kb/294418/`, retrieved 07.05.07.

[23] NASA. Aviris free standard data products. `http://aviris.jpl.nasa.gov/html/aviris.freedata.html`, retrieved 05.04.07.

[24] NEO. Products: Hyperspectral imaging. `http://www.neo.no/products/hyperspectral.html`, retrieved 01.05.07.

[25] S. Ozan and S. Gümüstekin. A case study on logging visual activities: Chess game. *Lect. Notes Comput. Sci.*, v 3949 LNAI:1–10, 2006.

[26] L. L. Randeberg, I. Baarstad, T. Løke, A. Winnem, E. Larsen, P. Kaspersen, O. Haugen, and L. Svaasand. Hyperspectral imaging of bruised skin. *Progr. Biomed. Opt. Imaging Proc. SPIE*, 6078:100–110, 2006.

[27] L. L. Randeberg, O. Haugen, R. Haaverstad, and L. Svaasand. A novel approach to age determination of traumatic injuries by reflectance spectroscopy. *Lasers Surg. Med.*, 38:277–289, 2006.

[28] L. L. Randeberg, A. Winnem, S. Blindheim, O. A. Haugen, and L. O. Svaasand. Optical classification of bruises. *Progr. Biomed. Opt. Imaging*, 5:54–64, 2004.

[29] J. A. Richards and X. Jia. *Remote Sensing Digital Image Analysis: An introduction.* Springer, Berlin, 2006.

[30] J. B. T. M. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41:187–228, 2000.

[31] C. N. Roger. Usgs spectroscopy lab - about imaging spectroscopy. `http://speclab.cr.usgs.gov/aboutimsp.html`, retrieved 26.03.07.

[32] R. Schalkoff. *Pattern Recognition - Statistical, Structural and Neural Approaches*. Wiley, Singapore, 1992.

[33] Sourceforge.net. Opencv wiki. `http://opencvlibrary.sourceforge.net/`, retrieved 27.02.07.

[34] G. N. Stamatas and N. Kollias. Blood stasis contributions to the perception of skin pigmentation. *J Biomed Opt*, 9:315–322, 2004.

[35] T. N. Tran, R. Wehrens, and L. M. Buydens. Clustering multispectral images: a tutorial. *Chemometrics and Intelligent Laboratory Systems*, 77:3–17, 2005.

[36] R. G. Wheeland. Lasers for removing unwanted hair and rejuvenating skin. *West J Med*, 169(4):228–229, 1998.

[37] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in software engineering: An introduction*. Kluwer, Boston, 2000.

[38] H. Wold. Estimation of principal components and related models by iterative least squares. *In Multivariate Analysis (Ed., P.R. Krishnaiah), Academic Press, NY*, pages 391–420, 1966.

[39] J. Xing, C. Bravo, P. T. Jancsók, H. Ramon, and J. D. Baerdemaeker. Detecting bruises on 'golden delicious' apples using hyperspectral imaging with multiple wavebands. *Biosyst Eng*, 09:27–36, 2005.

Appendix A

Tables

This chapter presents tables that provide additional information for the report.

A.1 Description of Available Data Set

Table A.1 shows details of the available data set. Details include an id for the image or image collection, the sex (Male(M)/Female(F)) and age (in years) of the subject. Images of items were marked not applicable (N/A) where necessary. Finally the comments give information on what the images contain, where and how old potential injuries are and which camera and light source was used.

Table A.1: *This table shows details on the available hyperspectral data set for this thesis.*

ID	Sex	Age	Comments
I1	M	32	A collection of 17 images taken at times ranging from 66 to 421 hours after injury. Images show two visible bruises on the volar side of the right elbow caused by paintball bullets. Camera:H1a Light:LS1
I2	M	32	A collection of 17 images taken at times ranging from 66 to 421 hours after injury. Images show two visible bruises on the volar side of the right elbow caused by paintball bullets. Camera:H3 Light:LS1
I3	Pig	-	A collection of 13 images taken at times ranging from before injury to about 1 minute after. Images show two bruises caused by pendulum with two differently shaped weights; a rounded tip and square shape. Camera:H2 Light:LS2
I4	Pig	-	A collection of 16 images taken at times ranging from before injury to about 1 hour after. Images show two bruises caused by paintball bullets. Camera:H2 Light:LS2
I5	M	24	Image show bruises on the dorsal side of the right upper arm caused by American football play. Image captured about 60 hours after injury. Camera:H1b Light:LS1

Continued on next page

ID	Sex	Age	Comments
I6	M	24	Image show bruises on the dorsal side of the left forearm caused by American football play. Image captured about 16 hours after injury. Camera:H1b Light:LS1
I7	M	24	Image show a bruise on the right lower back caused by American football play. Image captured about 16 hours after injury. Camera:H1b Light:LS1
I8	F	8	Image shows multiple bruises on the left calf caused by a moped accident. Image captured about 4 hours after injury. Camera:H1b Light:LS3
I9	M	24	2 images that show a possible bruise on the volar side of the left upper arm. Injury occurred during consumption of alcohol. Image captured about 12 hours after injury. Inspection 4 days later showed no visible bruise. Camera:H1b Light:LS3
I10	M	25	2 images that shows the volar side of the left hand. This is a reference image and it is not supposed to contain any bruises. Camera:H1b Light:LS3
I11	M	25	Image show the dorsal side of the right forearm. This is a reference image and it is not supposed to contain any bruises. Image contains some birthmarks. Camera:H1b Light:LS3
I12	M	23	Image show the dorsal side of the left forearm. This is a reference image and it is not supposed to contain any bruises. Some unidentified marks can be seen at the elbow. Camera:H1b Light:LS3
I13	F	30	Image shows the dorsal side of the right forearm. This is a reference image and it is not supposed to contain any bruises. Camera:H1b Light:LS3
I14	N/A	N/A	3 images of a colour reference chart. Camera:H1b Light:LS4

A.2 Selected Images for Experimentation

Table A.2 shows the images that were used for experimentation. The table gives names and comments for the selected images. A unique id (UID) is assigned to each image that belongs to an image collection as can be seen in Table A.1. For example, if two images come from image set I1, the first will be labeled I1-1 and the second I1-2.

Table A.2: *Details on the images selected for experimentation*

ID	UID	File name	Comments
I1	I1-1	72h	Visible bruises
I1	I1-2	192h	Visible bruises
I1	I1-3	397h	Visible bruises
I2	I2-1	72h_swir	Visible bruises
I2	I2-2	192h_swir	Visible bruises
I2	I2-3	397h_swir	Visible bruises
I3	I3-1	g7_s1_s2_pre_2006_06_02_ 10_16_59_VNIR_640_SN4_c.img	Before injury
I3	I3-2	g7_s1_s2_post_2006_06_02_ 10_44_58_VNIR_640_SN4_c.img	Possible bruise
I3	I3-3	g7_s1_s2_post_2_2006_06_02_ 11_09_35_VNIR_640_SN4_c.img	Visible bruises
I4	I4-1	g7s5s6post1_2006_06_02_ 13_28_05_VNIR_640_SN4_c.img	Before injury
I4	I4-2	g7s5s6post1_2006_06_02_ 13_41_27_VNIR_640_SN4_c.img	Visible bruise
I4	I4-3	g7s5s6post1_2006_06_02_ 14_46_40_VNIR_640_SN4_c.img	Visible bruises

A.3 Absorption Peaks and Corresponding Bands

Table A.3 shows the connection between absorption peaks and bands. If a wavelength is between two bands, both the one above and below is included.

Table A.3: *This table shows which band corresponds to important wavelengths for the different cameras used. Note that the compound bilirubin has only one peak, one for liquids (460 nm) and one for dermis (480 nm)*

Material	**Absorbtion peak (nm)**	**VNIR-1600 H1a**	**VNIR-1600 H1b**	**VNIR-640**	**SWIR-320i**
Bilirubin	460	16,17	14,15	11,12	-
Bilirubin	480	22	20	15,16	-
Deoxyheamoglobin	555	42	40,41	31	-
Deoxyheamoglobin	760	97	97	72,73	-
Metheamoglobin	508	29,30	27,28	21,22	-
Metheamoglobin	630	62	61,62	46,47	-
Oxyheamoglobin	542	38,39	37	28,29	-
Oxyheamoglobin	576	47,48	46,47	35,36	-
Random band	747	70	93,94	70	-
Random band	1443	-	-	-	110
Random band	1604	-	-	-	142

A.4 Visual Inspection of Bands in Reflectance VNIR Image

Table A.4 shows the result after a visual inspection of reflectance image I1-1. The comments serve as an aid for band selection when parameters for ratio and difference where chosen. The group shows which band range is described in the comments.

Table A.4: *Details on the contents found in the different bands of image I1-1.*

Group	Comments
$[1, 22]$	Weakly visible bruise
$[23, 54]$	Increasingly visible bruises, with peaks at band 38 an 47. The visibility of the bruises diminishes after that
$[55, 105]$	No visible bruises. Some changes in brightness between the different bands. Blood vessels are visible in these bands
$[106, 160]$	One bruise weakly visible. Increasing level of noise towards the final bands. Otherwise similar to $[55, 105]$

A.5 Visual Inspection of Bands in Radiance SWIR Image

Table A.5 shows results of visual inspection of radiance image I2-1. The comments serve as an aid for band selection when parameters for ratio and difference where chosen.

Table A.5: *Details on the contents found in the different bands of radiance image I2-1.*

Group	Comments
$[1, 14]$	Dark with varying amounts of banding effects
$[15, 47]$	One weakly visible bruise. Blood vessels visible
$[48, 103]$	Two visible bruises. Blood vessels are visible in the first layers
$[103, 117]$	No visible bruises or blood vessels
$[118, 153]$	Two visible bruises
$[154, 170]$	Dark with varying amounts of banding effects

A.6 Difference

Additional tables related to experimentation with the difference algorithm are presented in this section.

A.6.1 Phase One - Initial Test Results

Table A.6 shows results from phase one using absorption peaks as parameter selection criteria. Table A.7 shows results when parameters were selected based on visual inspection. The results are commented based on a visual inspection which serves as the basis for further use in phase two. If the parameters give an interesting result

(Int.), it is marked with a +, otherwise a - is given. Table A.6 is divided into sections, each starting with the compounds used as the basis for parameter selection.

Table A.6: *The parameters tested on image I1-1 using the difference algorithm.*

Parameters	**Comments**	**Int.**
*	*Deoxyheamoglobin, Oxyheamoglobin and Oxyheamoglobin, Deoxyheamoglobin*	*
42-38	Plenty of noise. Difficult to spot bruises	-
42-47	Lost skin structure, but the bruises are easy to see. Quite noisy	-
97-38	Some skin structure, but only one visible bruise. Bright areas around the bruises. Can see some blood vessels. Similar to (97-47).	+
97-47	Some skin structure, but only one visible bruise. Bright areas around the bruises. Can see some blood vessels. Similar to (97-38)	+
38-42	Loss of skin structure. No bruises visible	-
38-97	Loss of skin structure. One bruise barley visible. Arm is dark against bright background. Similar to (47-97)	-
47-42	Loss of skin structure, but easy to observe two bruises. Much noise	-
47-97	Loss of skin structure. One bruise barley visible. Arm is dark against bright background. Similar to (38-97)	-
*	*Bilirubin, Deoxyheamoglobin*	*
16-42	Loss of skin structure. Two bruises visible	+
16-97	No visible bruises. Some visible blood vessels. Similar to a brighter (22-97)	-
22-42	Loss of skin structure. Two bruises visible.	-
22-97	One weakly visible bruise. Some visible blood vessels. Similar to (16-97)	-
	Bilirubin, Metheamoglobin	*
16-29	Two bruises weakly visible. Loss of some skin structure	-
16-62	No visible bruises. Visible blood vessels and hair	-
22-29	Loss of some skin structure. Weakly visible bruises	-
22-62	No visible bruises. Visible blood vessels	-
*	*Bilirubin, Oxyheamoglobin*	*
16-38	Similar to (22-38)	+
16-47	Similar to a darker (22-38)	+
22-38	Two visible bruises surrounded by a dark area	+

Continued on next page

Parameters	Comments	Int.
22-47	Loss of some skin structure. Two strongly visible bruises surrounded by a dark area	+
	Deoxyheamoglobin, Metheamoglobin	*
42-29	Two clearly visible bruises. Loss of some skin structure. Similar to (47-29)	+
42-62	Weakly visible bruises. Some blood vessels are visible	-
97-29	One weakly visible bruises. Some blood vessels are visible	-
97-62	No visible bruise. Some blood vessels are visible	-
*	*Oxyheamoglobin, Metheamoglobin*	*
38-29	Two visible bruises. Lost some skin structure. Similar to (47-29)	+
38-62	Similar to (47-62), but with slightly less visible bruises	-
47-29	Two clearly visible bruises. Loss of skin structure	+
47-62	Two weakly visible bruises. Visible blood vessels and some hair	-

Table A.7: *Different parameters that were tested using the difference algorithm. The parameters where chosen based on a visual inspection of all the layers in image I1-1 and I2-1.*

UID	Parameters	Comments	Int.
I1-1	70-38	Similar to (90-47)	-
I1-1	70-47	Similar to (90-47), but the least visible bruise is showing better	-
I1-1	90-38	Similar to (90-47)	-
I1-1	90-47	One visible bruise, but the second is difficult to see	-
I1-1	121-38	Somewhat similar to (121-47), but less visible bruise	-
I1-1	121-47	One weakly visible bruise and some blood vessels	-
I1-1	140-38	Similar to (121-47), but less visible bruise	-
I1-1	140-47	Similar to (121-47), but with more noise on the arm	-
I1-1	121-70	Two visible bruises and visible blood vessels, but they are difficult to discern	-
I1-1	121-90	Blood vessels and bruises are dark and unclear	-

Continued on next page

UID	Parameters	Comments	Int.
I1-1	140-70	Two weakly visible bruises	-
I1-1	140-90	Two weakly visible bruises, but they are difficult to discern	-
I2-1	20-62	No visible bruises. Visible blood vessels	-
I2-1	20-110	One weakly visible bruise	-
I2-1	20-142	No visible bruises	-
I2-1	62-110	Two weakly visible bruises	-
I2-1	62-142	No visible bruises	-
I2-1	110-142	Linearly stretched. Two visible bruises	+

A.6.2 Phase Two - Extended Test Results

Table A.8 shows the results from phase two using parameters slected in phase one. Each image tested is identified by a unique id (UID), and a brief explanation to each image is given before it is tested. The interesting coloumn (Int.) shows if a result is judged to be positive or negative.

Table A.8: *Results from the extended testing using the difference algorithm*

UID	Parameters	Comments	Int.
I1-2	*	*192 hours after injury. Bruises are visible*	*
I1-2	22-38	Two bruises visible. Left bruise is more visible, but right bruise has more detail. Loss of some skin structure	+
I1-2	47-29	Two bruises visible. Left bruise is more visible, but right bruise has more detail. Loss of some skin structure	+
I1-3	*	*397 hours after injury. Bruises are weakly visible*	*
I1-3	22-38	Two bruises difficult to dicern. Noise and loss of skin structure	-
I1-3	47-29	Two bruises difficult to dicern. Right bruise slightly more visible than right. Dark spots on left side and some bright spots on right side	-
I2-2	*	*192 hours after injury. Two bruises visible*	*
I2-2	110-142	Two bruises weakly visible	+
I2-3	*	*397 hours after injury. Two bruises weakly visible*	*

Continued on next page

UID	Parameters	Comments	Int.
I2-3	110-142	Two bruises weakly visible	+
I3-1	*	*Before injury. No visible bruises, but a mark or a discolouration can be seen in square s1*	*
I3-1	15-28	No visible bruises	-
I3-1	35-21	No visible bruises. Dark	-
I3-2	*	*Right after injury. No visible bruises, but possible invisible bruises*	*
I3-2	15-28	No visible bruises	-
I3-2	35-21	No visible bruises	-
I3-3	*	*A square shaped bruise from a pendulum is visible in square s2*	*
I3-3	15-28	Weakly visible bruise	-
I3-3	35-21	Weakly visible bruise	-
I4-1	*	*One possible bruise in square s6. Strong specular light*	*
I4-1	15-28	No visible bruise	-
I4-1	35-21	No visible bruise	-
I4-2	*	*One circular bruise is visible. Strong specular light*	*
I4-2	15-28	Visible bruise, but slightly difficult to discern because of possible light influence (specular light)	-
I4-2	35-21	Visible bruise	-
I4-3	*	*Two circular bruises are visible. Strong specular light*	*
I4-3	15-28	Two bruises visible. The upper bruise is somewhat more difficult to spot	+
I4-3	35-21	Two bruises clearly visible	+

A.7 Ratio

Additional tables related to experimentation with the ratio algorithm are presented in this section.

A.7.1 Phase One - Initial Test Results

Table A.9 shows results from phase one using absorption peaks as parameter selection criteria. Table A.10 shows results when parameters were selected based on visual inspection. The results are commented based on a visual inspection which serves as the basis for further use in phase two. If the parameters give an interesting result

(Int.), it is marked with a +, otherwise a - is given. Table A.9 is divided into sections, each starting with the compounds used as the basis for parameter selection.

Table A.9: *The parameters tested on the base image I1-1 using the ratio algorithm.*

Parameters	Comments	Int.
*	*Deoxyheamoglobin, Oxyheamoglobin and Oxyheamoglobin, Deoxyheamoglobin*	*
42/38	Loss of some skin structure. Plenty of noise. Difficult to distinguish bruises from the rest	-
42/47	Loss of some skin structure. Plenty of noise. Bright bruises on dark background	+
97/38	One bruise visible and one slightly visible	+
97/47	Accentuation of bruises in addition to a possible bruise (somewhat fuzzy)	+
38/42	Loss of some skin structure. Plenty of noise. Difficult to see bruises	-
38/97	Accentuation of bruises in addition to a possible bruise. Similar to (47/97)	+
47/42	Loss of skin structure. Plenty of noise. Two dark bruises are visible	+
47/97	Two bruises and a possible bruise are emphasised. Similar to (38/97)	+
	Bilirubin, Deoxyheamoglobin	*
16/42	Two obvious bruises with dark areas around. Similar to (22/42)	+
16/97	No visible bruises, but dark areas where the visible bruises are supposed to be. Same goes for the possible bruise. Similar to (22/97)	+
22/42	Two obvious bruises, with dark areas around them. Similar to (16/42)	+
22/97	No visible bruises, but dark areas where the visible bruises are supposed to be. Same goes for the possible bruise. Similar to (16/97)	+
*	*Bilirubin, Metheamoglobin*	*
16/29	Similar to a brighter (22/29)	+
16/62	No visible bruises. Some visible blood vessels	-

Continued on next page

Parameters	Comments	Int.
22/29	Bright contours around two bruises. Dark areas around. Plenty of noise in the edges	+
22/62	No visible bruises, but dark areas where the bruises are supposed to be	-
	Bilirubin, Oxyheamoglobin	*
16/38	Similar to a brighter (22/38)	+
16/47	Similar to (22/38)	+
22/38	Two bright obvious bruises with dark areas surrounding them	+
22/47	Similar to a dark (22/38)	+
*	*Deoxyheamoglobin, Metheamoglobin*	*
42/29	Two visible bruises. Loss of skin structure	+
42/62	Two visible bruises and a possible bruise. Some blood vessels are visible. Dark	+
97/29	One vaguely visible bruise, in addition to some blood vessels	-
97/62	Difficult to spot bruises. Dark	-
	Oxyheamoglobin, Metheamoglobin	*
38/29	Similar to (47/29), but with some additional dark areas	+
38/62	Similar to (47/62)	+
47/29	Two obvious bruises. Loss of some skin structure	+
47/62	Two clearly visible bruises. Some visible blood vessels	+

Table A.10: *Different parameters that were tested using the ratio algorithm. The parameters where chosen based on a visual inspection of all the layers in image I1-1 and I2-1.*

UID	Parameters	Comments	Int.
I1-1	70/38	Similar to (70/47)	+
I1-1	70/47	Two clearly visible bruises and also a possible bruise	+
I1-1	90/38	Similar to a brighter (90/47)	-
I1-1	90/47	Two slightly visible bruises. Lost some skin structure	-

Continued on next page

UID	Parameters	Comments	Int.
I1-1	121/38	Two visible bruises and also a possible bruise. Not very clear	+
I1-1	121/47	Similar to (121/38)	+
I1-1	140/38	Similar to (121/38), but the bruises are smaller and less visible	+
I1-1	140/47	Similar to (121/38), but the bruises are smaller and less visible	+
I1-1	121/70	Similar to (140/70)	-
I1-1	121/90	Similar to (140/90)	-
I1-1	140/70	Two weakly visible bruises. Very dark.	-
I1-1	140/90	Two visible bruises and visible blood vessels, but very dark	-
I2-1	20/62	Linearly stretched. Clearly visible blood vessels	-
I2-1	20/110	No visible bruise. Similar to (62/110)	-
I2-1	20/142	Two visible bruises. Similar to (62/142)	-
I2-1	62/110	No visible bruise. Bright area might be heat related	-
I2-1	62/142	Two visible bruises	-
I2-1	110/142	Linearly stretched. No visible bruises	-

A.7.2 Phase Two - Extended Test Results

Table A.11 shows the results of applying a set of parameters on the selected image set. Details listed include which images have been tested and which parameters that were applied. More information on the images can be found in Section A.2 and A.1. The comments briefly explain the results using a certain set of parameters, while the interesting (Int.) column gives a summary of the comments. Two different categories are used in the Int. column: + (positive) and - (negative). Operations used to enhance further for visual inspection is also documented (i.e. additional linear stretching). All images had their spectralon cropped away before analysis. Each new test images is preceded with a brief explanation of what can be seen in the untreated image.

Table A.11: *Results from the extended testing using the ratio algorithm*

UID	Parameters	Comments	Int.
I1-3	*	*397 hours after injury. Bruises are weakly visible*	*
I1-3	22/38	No visible bruises	-
I1-3	47/29	Weakly visible bruises	-

Continued on next page

UID	Parameters	Comments	Int.
I1-3	47/97	Two weakly visible bruises. Bruise to the left is more visible than the one to the right. Dark triangel shape is visible between the bruises	+
I1-3	70/47	Two visible bruises, but details are difficult to observe. Bright triangle shape is visible between the bruises	+
I3-1	*	*Before injury. No visible bruises, but a mark or a discolouration can be seen in square s1*	*
I3-1	35/72	Linearly stretched. No visible bruises. The mark is slightly visible	-
I3-1	15/28	No visible bruise or mark. Some noise on the letters on the leg	-
I3-1	35/21	Linearly stretched. No visible bruises or mark.	-
I3-1	70/35	No visible bruises or mark	-
I3-2	*	*Right after injury. No visible bruises, but possible invisible bruises*	*
I3-2	35/72	Linearly stretched. No visible bruises	-
I3-2	15/28	No visible bruises	-
I3-2	35/21	Linearly stretched. No visible bruises	-
I3-2	70/35	No visible bruises	-
I4-1	*	*One possible bruise in square s6. Strong specular light*	*
I4-1	35/72	Linearly stretched. One possible bruise weakly visible	-
I4-1	15/28	No visible bruises	-
I4-1	35/21	Linearly stretched. No visible bruises	-
I4-1	70/35	One possible bruise weakly visible	-
I3-3	*	*A square shaped bruise from a pendulum is visible in square s2*	*
I3-3	35/72	Linearly stretched. Bruise is visible, but no improvements compared to original	-
I3-3	15/28	Bruise is weakly visible	-
I3-3	35/21	Linearly stretched. Bruise is visible, but no improvements to original	-
I3-3	70/35	Linearly stretched. Bruise is visible, but no improvements compared to original	-
I4-3	*	*Two circular bruises are visible. Strong specular light*	*

Continued on next page

UID	Parameters	Comments	Int.
I4-3	35/72	Two clearly visible bruises	+
I4-3	15/28	Two visible bruises, but s6 is clearer than s5	-
I4-3	35/21	Linearly stretched. Two visible bruises. Grey ring seems to follow the contour of the bruise	+
I4-3	70/35	Two bruises clearly visible	+
I4-2	*	*One circular bruise is visible. Strong specular light*	*
I4-2	35/72	One bruise clearly visible	+
I4-2	15/28	One bruise visible. The bruise and the surrounding area is much brighter than surrounding tissue. Some bruise details is lost	+
I4-2	35/21	One bruise slightly visible. An unknown dark circle surrounds the bruise	+
I4-2	70/35	One bruise visible. The bruise and the surrounding area is much brighter than other tissue in the cropped image. A dark area can be seen around half the bruise	+
I1-2	*	*192 hours after injury. Bruises are visible*	*
I1-2	22/38	Two bruises visible	+
I1-2	47/29	Two bruises visible	+
I1-2	47/97	Two bruises visible. Dark triangel shape is visible between the bruises. Some blood vessels visible	+
I1-2	70/47	One bruise clearly visible. The right bruise is visible, but more difficult to discern than the left one. Bright triangel shape is visible between the bruises. Some blood vessels visible	+

A.8 Principal Component Analysis

Table A.12 shows results from testing with PCA. Principal components (PC) that where judged to contained to much noise are not listed in the table.

Table A.12: *Results from a visual inspection of principal components (PC) of different images. PC indicates which component is beeing inspected. The radiance images follow directly after their reflectance counter parts.*

UID	PC	Comments	Sum
*	*	*Male arm with two circular bruises at 72 hours after injury*	*
I1-1	1	No visible bruises. Hair and some blood vessels are weakly visible. Dark	-
I1-1	2	One bruise is visible. Hair and some blood vessels are visible	-
I1-1	3	Two bruises are visible. Difficult to distinguish bruises from blood vessels. Hair and some blood vessels are strongly visible	-
I1-1	4	Two bruises strongly visible. Hair is weakly visible	+
I1-1	5	Two bruises are strongly visible. Loss of some skin structure. No hair visible	+
I1-1	6	No bruises visible	-
I1-1	10	No bruises visible. Some possible blood vessels are visible. Some noise	-
I1-1	*	*Radiance image*	*
I1-1	1	No visible bruises. Dark	-
I1-1	2	One weakly visible bruise. Similar to PC1, but with more details. Dark	-
I1-1	3	Two bruises visible, with different strengths. Some contour lost on left bruise. Hair is visible	+
I1-1	4	Similar to an inverted PC3	+
I1-1	5	Two clearly visible bruises. A triangle shape visible between and below the bruises. Hair is visible	+
I1-1	6	Two bruises visible, but the right one has lost its contour. A triangle shape visible between and below the bruises. Some hair visible	-
I1-1	7	One bruise weakly visible. Bright lines could be blood vessels. Some hair weakly visible	-
I1-1	8	Bruises are difficult to dicern from the rest of the image. Some bright 'branching' effects observed on the arm	-

Continued on next page

UID	PC	Comments	Sum
I1-1	9	Two bruises visible, but the left is difficult to discern	-
I1-1	10	Two bruises weakly visible. Some hair visible	-
I1-1	11	Two bruises clearly visible. Bright lines observed on the arm	+
I1-1	17	Two bruises clearly visible. Some noise	+
I1-1	18	Two bruises clearly visible. Some 'racoon' effect seen around the bruises. Some bright lines on the arm. Some noise	+
I1-1	45	Two visible bruises. Noisy	+
I1-1	143	Two visible bruises, with some dark areas scattered around them. Some noise	+
I1-2	*	*Male arm with two circular bruises at 192 hours after injury*	*
I1-2	1	No bruise visible. Dark	-
I1-2	2	One bruise weakly visible	-
I1-2	3	One bruise visible, but difficult to discern. Hair is clearly visible. Black spots similar to the bruise can be seen on the arm	-
I1-2	4	Two bruises visible. The left bruise has lost some contour- Hair and some blood vessels are vissible	+
I1-2	5	One bruise visible, but difficult to discern from arm. Hair visible	-
I1-2	6	One bruise visible. Possible blood vessels and hair visible	-
I1-2	7	Two bruises visible. Left bruise has lost some contour	+
I1-2	8	Two bruises visible. Bright spots elsewhere on the arm makes the bruises difficutl to discern	-
I1-2	9	No visible bruises. Some bright lines observed	-
I1-2	10	Two bruises visible. Some noise	+
I1-2	11	Two bruises visible, but they are difficult to discern. Noise	-
I1-2	12	Two bruises visible, but right bruise is difficult to discern. Noise	-
I1-2	13	Two bruises weakly visible. Noise	-

Continued on next page

UID	PC	Comments	Sum
I1-2	14	Two bruises weakly visible. Noise	-
I1-2	15	No visible bruises	-
I1-3	*	*Male arm with two circular bruises at 397 hours after injury*	*
I1-3	1	No bruises visible. Dark	-
I1-3	2	One bruise weakly visible	-
I1-3	3	One bruise visible. Dark lines could be blood vessels. Hair visible	-
I1-3	4	One bruise visible	-
I1-3	5	Left bruise very visible, but with some contour losses. Right bruise is weakly visible. Hair is visible	-
I1-3	6	One bruise barley visible. Some blood vessels visible, but the hair is clearly visible	-
I1-3	7	Similar to PC6, but with less blood vessel details	-
I1-3	8	Left bruise very visible, but with some contour losses. Right bruise is difficult to discern. Hair is visible	+
I1-3	9	No bruises visible. Possible blood vessels and hair visible	-
I1-3	21	No visible bruises	-
I1-3	22	No visible bruises. Some possible blood vessels visible. Noise	-
I1-3	24-26	No visible bruises	-
I1-3	29	No visible bruises	-
I2-1	*	*Male arm with two circular bruises at 72 hours after injury*	*
I2-1	1	Two weakly visible bruises	-
I2-1	2	Two weakly visible bruises. Some blood vessels visible	-
I2-1	3	No visible bruises. Some blood vessels visible	-
I2-1	4	One bruise weakly visible. Blood vessels visible	-
I2-1	5	No visible bruises. Visible blood vessels	-
I2-1	6	One bruise visible. Heavy striping effects	-
I2-1	7	One bruise visible. Some hair and blood vessels visible. Striping effects	+
I2-1	8	Similar to PC7, but brighter. One bruise visible, but difficult to discern	-

Continued on next page

UID	PC	Comments	Sum
I2-1	9	No visible bruises. Some visible blood vessels	-
I2-1	10	No visible bruises. Visible blood vessels	-
I2-1	11	One visible bruise, but difficult to discern. Some blood vessels visible	-
I2-1	12	Similar to PC10, but with less details	-
I2-1	13	One weakly visible bruise. Some striping effects	-
I2-1	14	Two weakly visible bruises. Some blood vessels. Heavy striping effects	-
I2-1	15	Two visible bruises	+
I2-1	23	One clearly visible bruise and one difficult to discern. Some blood vessels visible. Heavy striping effects	+
I2-1	25	Similar to PC23. Heavy striping effects	-
I2-1	28	Similar to PC23, but with slightly less visible bruises	-
I2-1	29	One bruise very visibleThe other has lost some contour, but is still visible. Blood vessels visible	+
I2-1	71	One bruise visible. Some blood vessels visible. Heavy striping effect	-
I2-2	*	*Male arm with two circular bruises at 192 hours after injury*	*
I2-2	13	Two visible bruises, but image is corrupted by vertical lines	-
I2-2	*	*No further tests done on image I2-2 because of corruption in all principal components*	*
I2-3	*	*Not evaluated because of heavy corruption in all principal components*	*
I3-1	*	*Pig before injury. No visible bruises*	*
I3-1	1	No visible bruises	-
I3-1	2	No visible bruises. No or little shadow effects. Striping effect	-
I3-1	3	No visible bruises. Dark, wiht a bright spot at the bottom. Stripping effect	-
I3-2	*	*Pig right after injury. No visible bruises*	*
I3-2	1	No visible bruises	-
I3-2	2	No visible bruises. No or little shadow effects	-

Continued on next page

UID	PC	Comments	Sum
I3-2	3	No visible bruises. No or little shadow effects. Striping effect	-
I3-2	*	*Radiance image*	*
I3-2	1	No visible bruises	-
I3-2	2	No visible bruises. Striping effects	-
I3-2	3	No visible bruises. Striping effects	-
I3-2	4	No visible bruises. Dark area at the bottom	-
I3-2	5	No visible bruises. No or little shadow effects. Heavy striping effects	-
I3-2	6	No visible bruises. Bright spot where there is a dark spot in PC4	-
I3-3	*	*Pig with a single square bruise*	*
I3-3	1	No visible bruises	-
I3-3	2	One visible bruise. No or little shadow effects	-
I3-3	3	One weakly visible bruise. Striping effects	-
I4-1	*	*Pig with possibel bruise*	*
I4-1	1	No visible bruises	-
I4-1	2	Linearly stretched. One possible bruise weakly visible. Striping effects	-
I4-1	3	Possibel bruise weakly visible. Heavy striping effects	-
I4-1	9	No bruise visible. Dark with a bright spot	-
I4-2	*	*Pig with one circular bruise*	*
I4-2	1	One weakly visible bruise	-
I4-2	2	One clearly visible bruise	+
I4-2	3	One clearly visible bruise. Striping effects	+
I4-2	26	One clearly visible bruise. Striping effects and heavy noise	+
I4-2	27	One clearly visible bruise. Striping effects and heavy noise. Most details have disapeared	+
I4-3	*	*Pig with two circular bruises*	*
I4-3	1	Two bruises visible. Possible shading effect	-
I4-3	2	Two bruises clearly visible. The lower bruise is partially surrounded by a dark area. This might be influence from the light source. Some striping effects	+

Continued on next page

UID	PC	Comments	Sum
I4-3	3	Two bruises clearly visible. Little or no visible influence from the light source with respect to shading or specular light. Some striping effects.	+
I4-3	4	Two bruises visible. Possible specular light from the light source	-
I4-3	*	*Radiance image*	*
I4-3	1	Two visible bruises, but the bottom one is weak	-
I4-3	2	One bruise visible. Top bruise dissapears in bright area	-
I4-3	3	Bottom bruise very visible, but top bruise is difficult to see. Some striping effect	+
I4-3	4	Two bruises visible. Top bruise is more visible than bottom bruise. Striping effect	+
I4-3	5	Two bruises weakly visible. Striping effect	-
I4-3	6-7	Two bruises weakly visible. Heavy striping effect	-
I4-3	8	Two bruises visible, but contours are difficult to see on the bottom bruise. Striping effect	-
I4-3	9	Similar to PC8, but much brighter	-
I4-3	10-16	Two bruises visible, but heavy striping effects	-
I4-3	107	Two dark clearly visible bruises. Striping effects	+
I4-3	108	Two bright visible bruises. Some striping effects and noise. Bottom bruise is brighter than top one	+
I4-3	109	Two dark clearly visible bruises. Striping effects	+

A.9 K-Means Clustering

Tabels related to experimentation with the K-means clustering algorithm is presented in this section.

A.9.1 Experimentation Parameters

The parameters used as input for the K-means clustering algorithm are listed in Table A.13. The table also shows if the clustering converged and at what iteration number this occured during the experimentation.

Table A.13: *Details on the input parameters for K-means clustering and notes on convergence*

Clusters	Iterations	Comments
2	1,2,4,8,16	Converged after 10 iterations
3	1,2,4,8,16	Converged after 15 iterations
4	1,2,4,8,16,32	Converged after 18 iterations
5	1,2,4,8,16,32,100	Converged after 61 iterations
6	1,2,4,8	No convergence
7	1,2,4,8	No convergence
8	1,2,4,8,250	Converged at 124 iterations
9	1,2,4,8,16	No convergence
10	1,2,4,8	No convergence
12	1,2,4,8,16	No convergence
14	1,2,4,8	No convergence
16	1,2,4,8,16	No convergence
18	1,2,4,8	No convergence
25	1,2,8	No convergence
30	1,2,8	No convergence
40	1,2,8	No convergence
50	1,2,8	No convergence

A.9.2 Initial Phase Results

This section presents parameters that gave positive clustering segmentations. Table A.14 shows parameters from testing on reflectance image I1-1, while Table A.15 shows parameters from testing on the first 10 principal components of reflectance image I1-1.

Table A.14: *The parameters that gave a positive clustering segmentation of one or more bruises for reflectance image I1-1.*

Clusters	**Iterations**	**Comments on Result Segmentations**
10	1	Upper half of left bruise marked, no marks for right bruise.
14	1	Right half of left bruise marked, no marks for right bruise.
16	8	Left edge of left bruise marked weakly, no marks for right bruise.
18	2	Less than half of area on upper and lower edge of left bruise marked, no marks for right bruise.
18	4	Less than half of area on upper and lower edge of left bruise marked, no marks for right bruise.
18	8	Less than half of area on upper and lower edge of left bruise marked, no marks for right bruise.
25	2	(Oversegmented) Half of left bruise selected with upper and lower edge, no marks for right bruise.
25	8	(Oversegmented) Less than half of left bruise selected with upper and lower edge, no marks for right bruise.
30	8	(Oversegmented) Left half of left bruise selected, no marks for right bruise.
40	1	(Oversegmented) Entire edge of bruise marked with unsharp edges in two regions. Left part of right bruise marked in one unsharp region..
40	2	(Oversegmented) Entire edge of bruise and center marked with unsharp edges. Left part of right bruise marked in one unsharp region.
40	8	(Oversegmented) Left half of left bruise and center marked. Left part of right bruise marked.
50	1	(Oversegmented) Edges on left half of right and left bruises selected.

Table A.15: *The parameters that gave a positive clustering segmentation of one or more bruises for the first 10 principal components of reflectance image I1-1.*

Clus.	**Iters.**	**Comments on Result Segmentations**
8	4	Some of right edge of left bruise marked.
16	1	Some of right edge of left bruise marked.
20	1	Full right edge of left bruise marked.
20	8	Some of left edge of left bruise marked.
20	16	Full upper edge and partial lower edge of left bruise selected.
20	32	Full upper edge and partial lower edge of left bruise selected.
25	2	(Oversegmented) Some of left edge of left bruise marked.
25	16	(Oversegmented) Some of right edge of left bruise marked.
25	32	(Oversegmented) Full left bruise marked in two regions.
30	1	(Oversegmented) Left half of left bruise marked. Left part of right bruise marked in one unsharp region.
30	4	(Oversegmented) Left half of left bruise marked. Left part of right bruise marked in one unsharp region.
30	16	(Oversegmented) Left half of left bruise marked. Center part of right bruise marked in one unsharp region.
30	32	(Oversegmented) Left half of left bruise marked.
40	4	(Oversegmented) Left half of left bruise marked. Left part of right bruise marked in one unsharp region.
40	8	(Oversegmented) Left half and upper edge of left bruise marked in two separate segments.
40	16	(Oversegmented) Left half and upper edge of left bruise marked in two separate segments.
40	32	(Oversegmented) Edge and center of left bruise marked in single segment. Segment slightly bigger than visual bruise.
50	2	(Oversegmented) Left half and upper edge of left bruise marked in three segments.
50	4	(Oversegmented) Left half and upper edge of left bruise marked in three segments. Left side of right bruise marked in single region.
50	8	(Oversegmented) Left half and upper edge of left bruise marked in two separate segments.
50	16	(Oversegmented) Left half and upper edge of left bruise marked in two separate segments.
50	32	(Oversegmented) Left half and upper edge of left bruise marked in two separate segments.

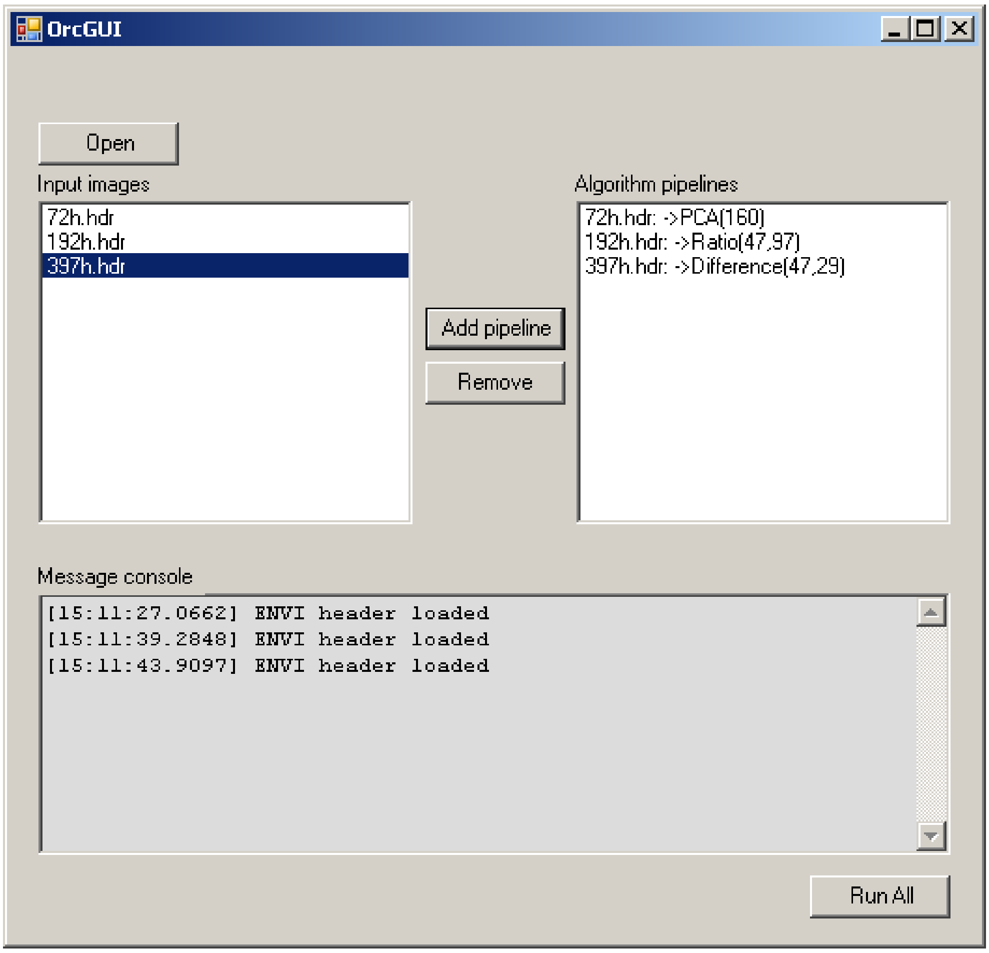
Appendix B

Figures

This chapter presents figures that provide additional information for the report.

B.1 Application Design

The main window for the batch program can be seen in Figure B.1. The white window to the left show available images, while the one to the right shows the images that has been added to the pipeline. The algorithm that will be applied to the images can be seen after the image names. The message console prints messages to the user while the program is running. The 'Run All' button starts the batch and runs all the algorithms currently in the pipeline.



Input images:

```
72h.hdr
192h.hdr
397h.hdr
```

Algorithm pipelines:

```
72h.hdr: ->PCA(160)
192h.hdr: ->Ratio(47,97)
397h.hdr: ->Difference(47,29)
```

Message console:

```
[15:11:27.0662] ENVI header loaded
[15:11:39.2848] ENVI header loaded
[15:11:43.9097] ENVI header loaded
```

Figure B.1: *The main window for the batch program.*

B.2 Difference

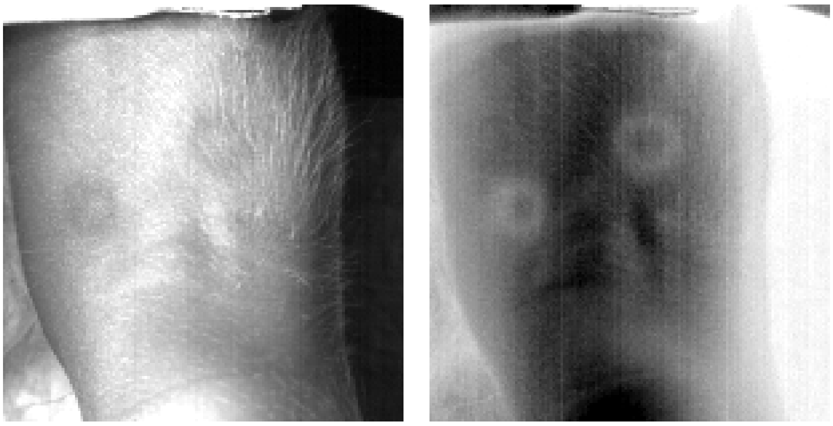
Additional figures related to experimentation with the difference algorithm are presented in this section.

B.2.1 Phase One - Initial Test Results

This section presents the result image from I2-1 using the selected parameter and a couple of negative results from the selection process.

Figure B.2(a) shows band 142 of the original I2-1 image, while (b) shows the positive result. The bruises show as bright circles against the black arm. Some stripping effects are observed.

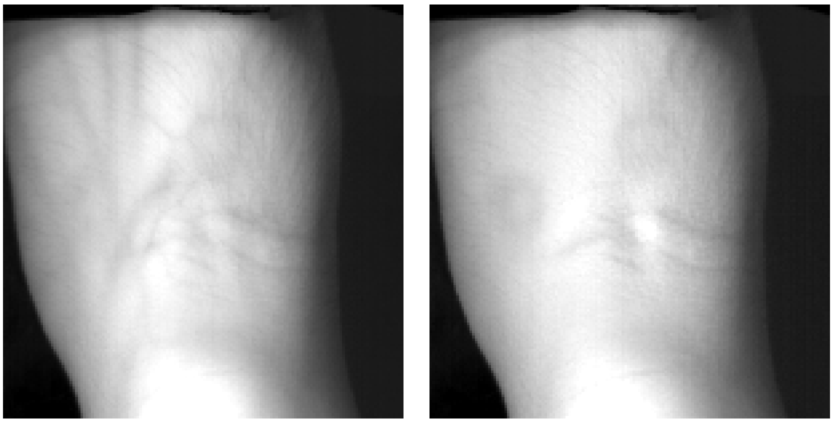
The negative results from I2-1 contained no discernable noise, but as Figures B.3(a) and (b) show, bruises are weakly visible. Blood vessels are visible in (a).



(a) I2-1 original, band 142

(b) Band 110 and 142

Figure B.2: *One positive result image from phase one using the difference algorithm on image I2-1. Parameters that were used are listed below each image*



(a) Band 20 and 110

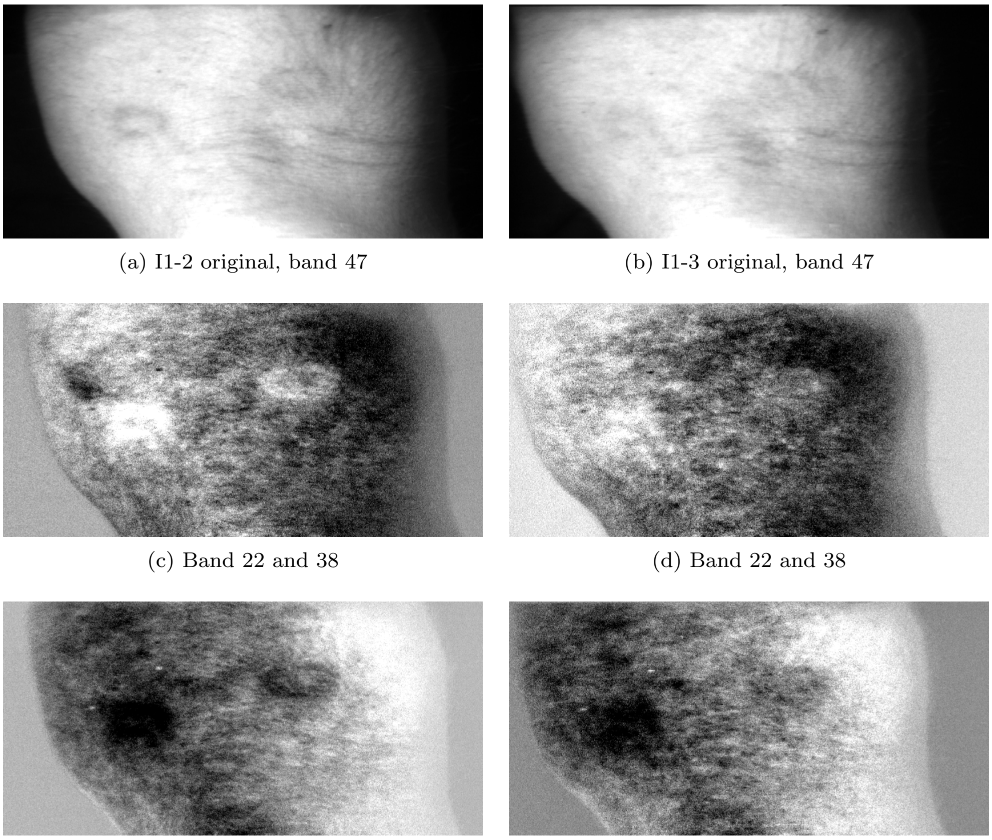
(b) Band 62 and 110

Figure B.3: *Two negative results from phase one using the difference algorithm on image I2-1. Parameters that were used are listed below each image*

B.2.2 Phase Two - Extended Test Results

This section presents results from the difference algorithm using I1-2 and I1-3, followed by the results using I2-2 and I2-3. Results from images containing no clearly visible bruises are presented at the end.

Figures B.4(a) and (b) show band 47 from the original reflectance images at 192 and 397 hours after injury respectively. Bruises are visible in (a), but difficult to discern in (b). Figures B.4(c) and (e) show two bruises, but the left has lost its contour compared to the right bruise. The bruises are difficult to see in both Figures B.4(d) and (f).



(a) I1-2 original, band 47

(b) I1-3 original, band 47

(c) Band 22 and 38

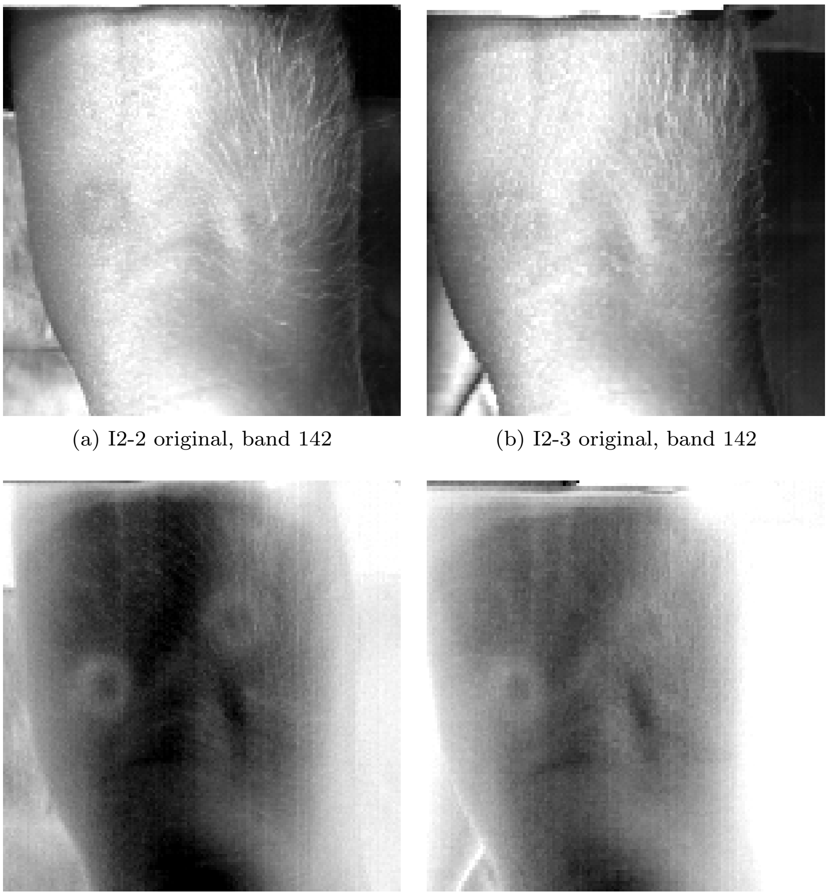
(d) Band 22 and 38

(e) Band 47 and 29

(f) Band 47 and 29

Figure B.4: *Four result images using I1-2(a) and I1-3(b). The left side show results from I1-2 captured 192 hours after injury. The right side show results from I1-3 captured 397 hours after injury.*

Figures B.5(a) and (b) show band 142 from the original radiance images at 192 and 397 hours after injury respectively. Bruises are visible in (a), but they are weakly visible in (b). Both (c) and (d) show bright bruises against a dark arm and both contain striping effect.



(a) I2-2 original, band 142

(b) I2-3 original, band 142

(c) Band 110 and 142

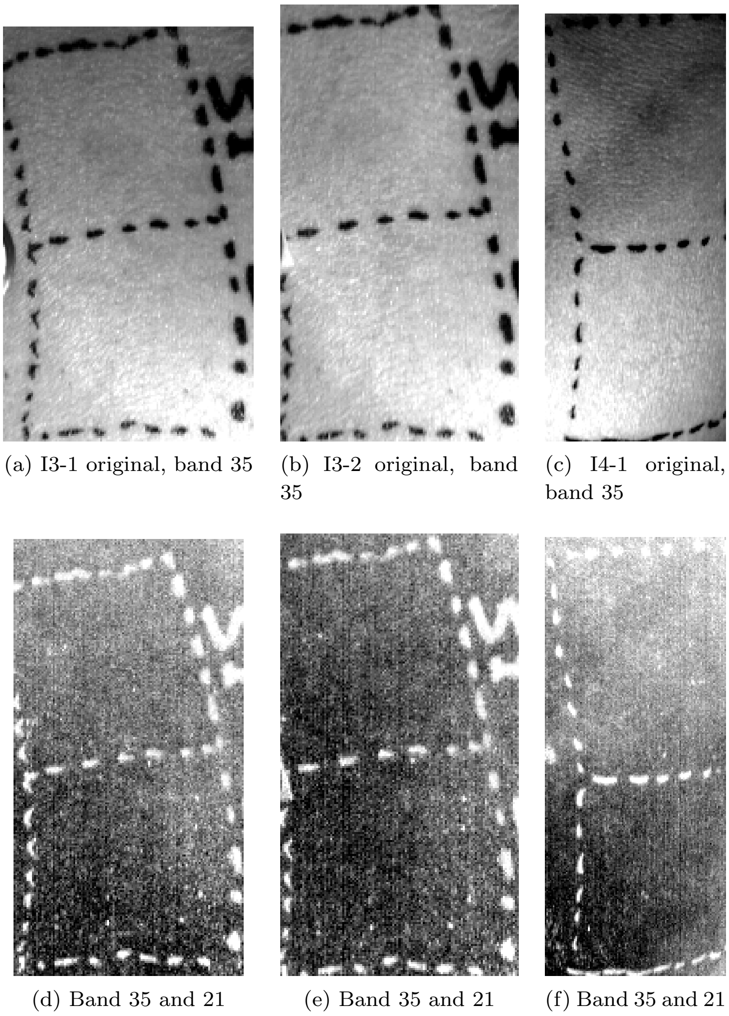
(d) Band 110 and 142

Figure B.5: *Image (c) shows results from I2-2(a) captured 192 hours after injury. Image (d) shows results from I2-3(c) captured 397 hours after injury.*

Figure B.6 shows results that had no clearly visible bruise in the original image. None of the tested parameters produced false positives. This means that there was no clear indication of bruises before or after testing. Three example parameters and their result images can be seen in Figures B.6(d), (e), and (f). None show any clear signs of bruises. The originals can be seen above each picture. Striping effects are visible in all the result images.

B.2.3 Other Observations

Figure B.7 shows two result images from image I1-1 with their parameters reversed. Figure B.7(a) shows dark bruises, while (b) show bright bruises. All bruises are clearly visible.



(a) I3-1 original, band 35

(b) I3-2 original, band 35

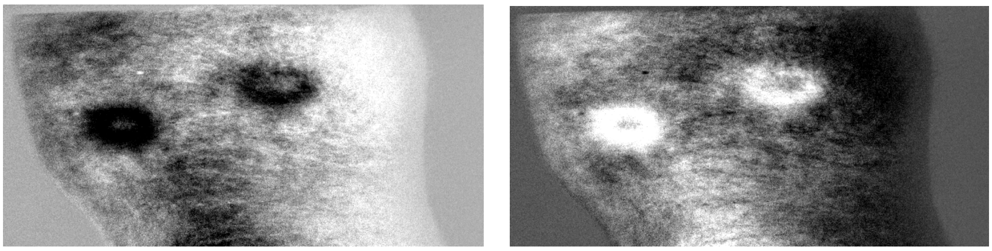
(c) I4-1 original, band 35

(d) Band 35 and 21

(e) Band 35 and 21

(f) Band 35 and 21

Figure B.6: *Six images taken from a pig were there are no clearly visible bruises. The originals show in (a), (b), and (c), while the images below them show three example parameters.*



(a) Band 47 and 29

(b) Band 29 and 47

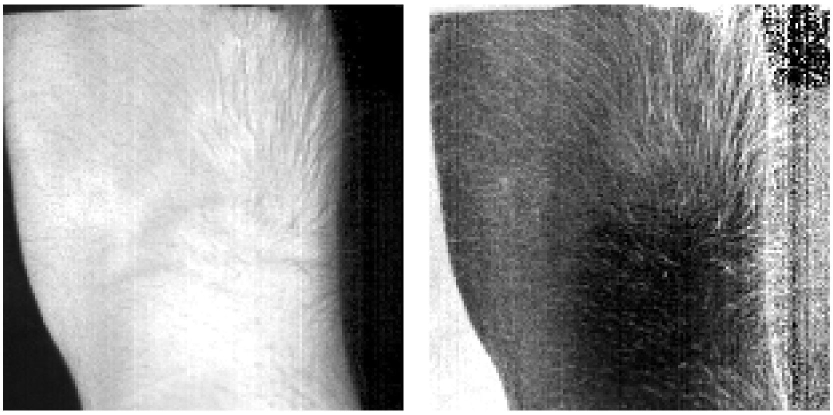
Figure B.7: *An example of two results with their parameters reversed.*

B.3 Ratio

This section presents additional figures related to experimentation with the ratio algorithm.

B.3.1 Phase One - Initial Test Results

This section presents two negative results from the initial testing of image I2-1. Figures B.8(a) and (b) show bruises that are difficult to discern. Hair is visible in both images, but most clearly in image (b).



(a) Band 62 and 142

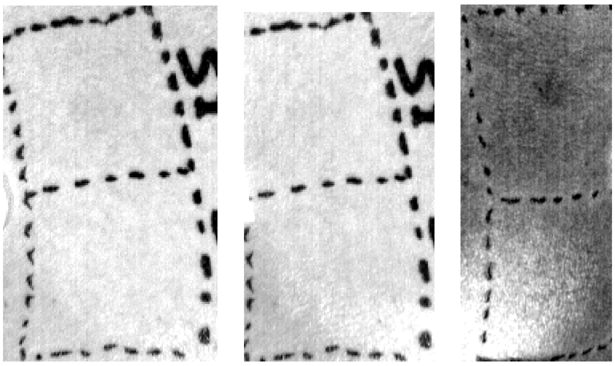
(b) Band 110 and 142

Figure B.8: *Two negative result images from phase one using the ratio algorithm on image I2-1. Parameters that were used are listed below each image.*

B.3.2 Phase Two - Extended Test Results

This section presents three example result images from images containing no clearly visible bruises, followed by results from testing on image I1-2 and I1-3.

Figure B.9 shows three result images from images of the pig that had no clearly visible bruises in the original image. None of the tested parameters produced false positives, meaning that there was no clear indication of bruises before or after testing. Some striping effect is observed in Figure B.9(c). The original images can be found in Figures B.6(a), (b), and (c).



(a) I3-1, band 35 and 72

(b) I3-2, band 35 and 72

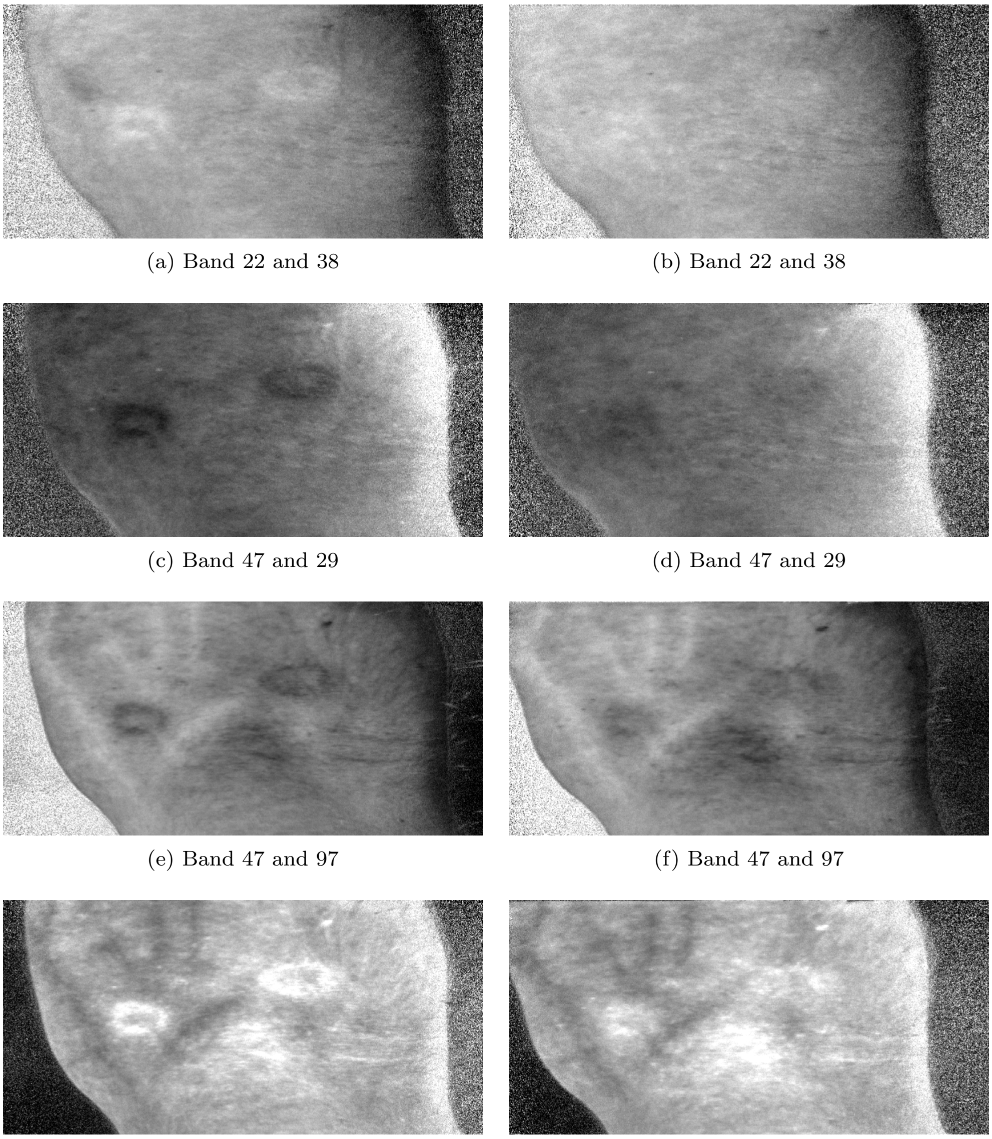
(c) I4-1, band 35 and 72

Figure B.9: *Three result images of a pig with no clearly visible bruises.*

The left column of Figure B.10 shows results taken at 192 hours after injury (I1-2), while the right column shows results taken at 397 hours after injury (I1-3). Figure B.4(a) shows band 47 of the original I1-2 while (b) shows the original of I1-3. Two visible circular bruises can be seen in the original of I1-2, but they are more difficult to discern for I1-3. The same goes for the corresponding result images. Bruises are visible in Figures B.10(c), (e), and (g), while they are a little weaker in (a). Figure B.10(b) does not show any bruises while (d) have two barely visible bruises. The bruises are slightly more visible in (f) and (h) compared to (d). Possible blood vessels are visible in (e), (f), (g), and (h), along with a triangle shape below and between the bruises. All images contain noise in the background.

B.3.3 Other Observations

Figure B.11 shows two result images from image I1-1 with their parameters reversed. Figure B.11(a) shows two dark circular bruises surrounded by a weak brighter 'racoon' effect. The same can be seen in (b), but with bright bruises and a darker 'racoon' effect.



(a) Band 22 and 38

(b) Band 22 and 38

(c) Band 47 and 29

(d) Band 47 and 29

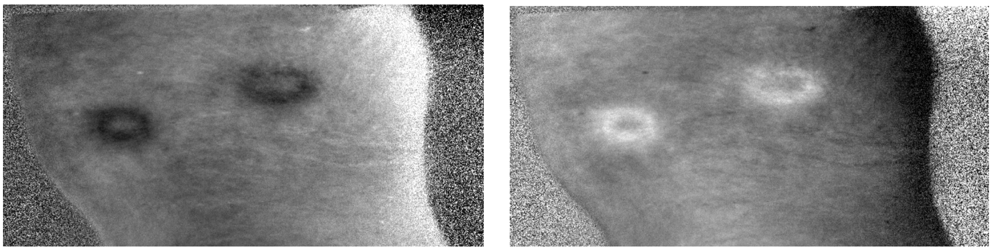
(e) Band 47 and 97

(f) Band 47 and 97

(g) Band 70 and 47

(h) Band 70 and 47

Figure B.10: *Eight result images using selected parameters on image I1-2 and I1-3. The left column show results from I1-2, while the right column show results from I1-3. Parameters that were used are listed below each image.*



(a) Band 47 and 29

(b) Band 29 and 47

Figure B.11: *Two results from I1-1 with reversed parameters.*

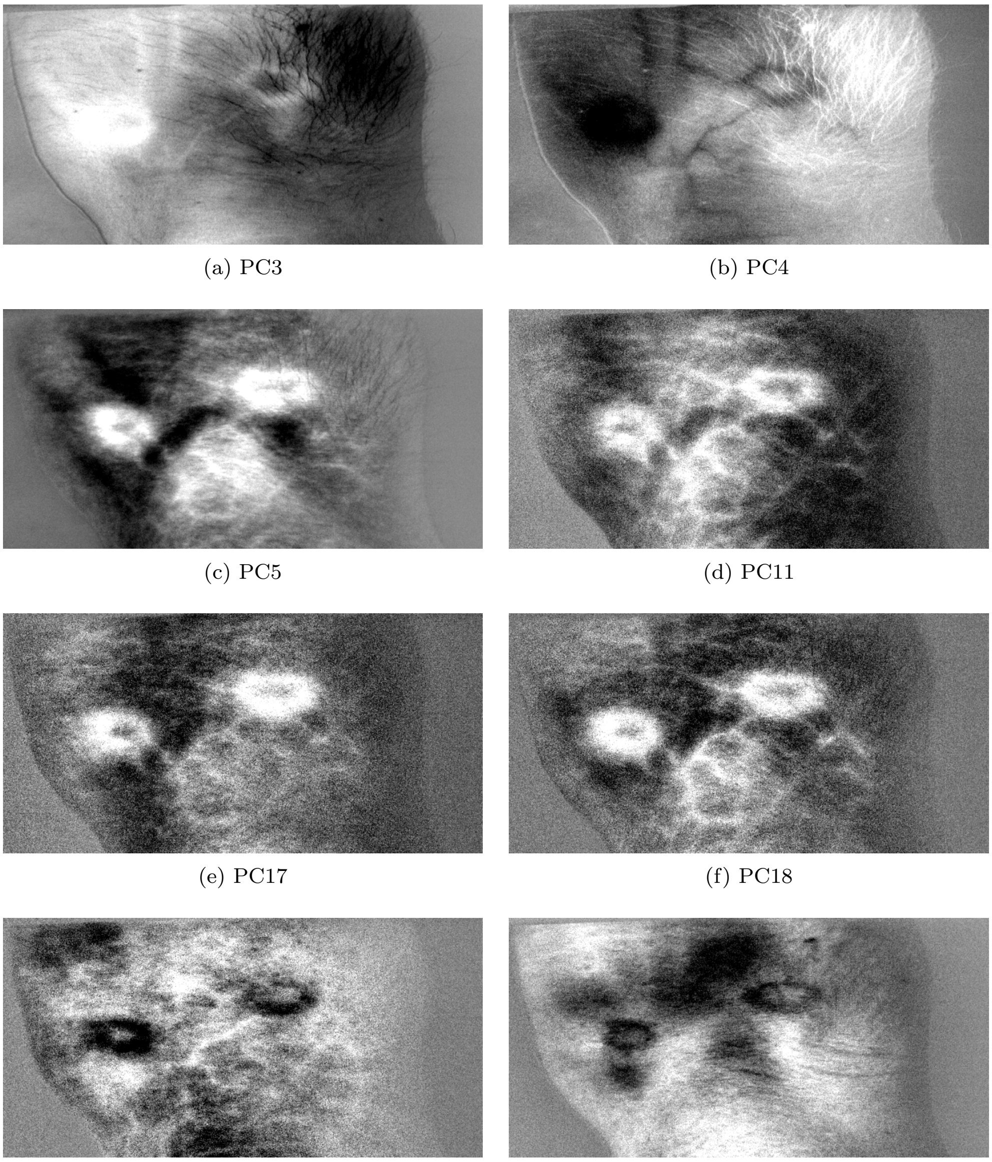
B.4 Principal Component Analysis

This section presents the results from using PCA on radiance images. Three radiance images were tested to build a foundation for a brief comparison between using radiance images and reflectance images. The results from these radiance images are covered first. Following this is the result images from testing on images from I2.

Figure B.12 shows eight positive results from testing on radiance image I1-1. Interesting results are found in PC3 to five, but also in later components. All the results display the bruises clearly, but with some minor differences to the right bruise in (a) and (b) and in both the bruises in (h). The bruises look a bit smaller in (h), but they have better contours. One set of blood vessels can be seen in (a) and (b), while (c) through (g) show a different network of blood vessels. Dark lines in (c) can be yet another set of blood vessels. The 'racoon' effect can be seen in (f) and partly in (e) and (h). Hair is clearly visible in (b) and a little less visible in (a) and (c). Figures B.12(a) and (b) contains much of the same details, but with seemingly inverted colours.

Figure B.13 shows positive results from testing with PCA on image I4-3. PC3 and PC4 gave good results, even though one bruise in each was slightly difficult to discern. Surprisingly, some positive results were found in PC107, 108, and 109 as well. These have bruises more visible than PC3 and 4. The bruises have different sizes in the principal components. Striping effects are visible in all components, but most clearly in PC4.

Three radiance images were originally supposed to be tested from set I2, in a similar manner to set I1. Heavy striping effects and image corruption meant that only I2-1 was included in the report, since the others had inadequate quality. Figure B.14 shows five positive results from image I2-1. Positive results comes later compared to I1-1, and the first came at PC7. After that, results came spaced out between PC15 and PC29. The bruises are clearly visible in all the images, but the right bruise has lost some contour and is difficult to discern in all but result (b). Blood vessels are visible in all result images, were Figures B.14(a) and (e) show two kinds of blood vessel networks. One has thicker vessels than the other. Hair is visible in (a). All result images contain heavy striping effects and some corruption in the lower part.



(a) PC3

(b) PC4

(c) PC5

(d) PC11

(e) PC17

(f) PC18

(g) PC45

(h) PC143

Figure B.12: *Eigth positive results from radiance image I1-1. The principal component (PC) displayed are listed below each image.*

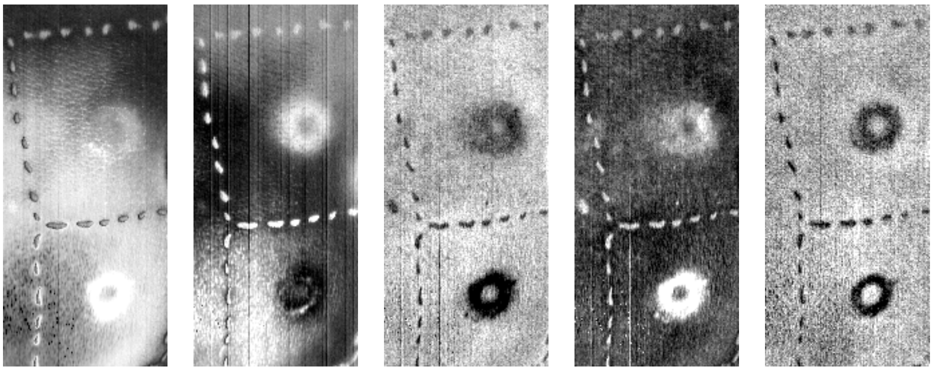
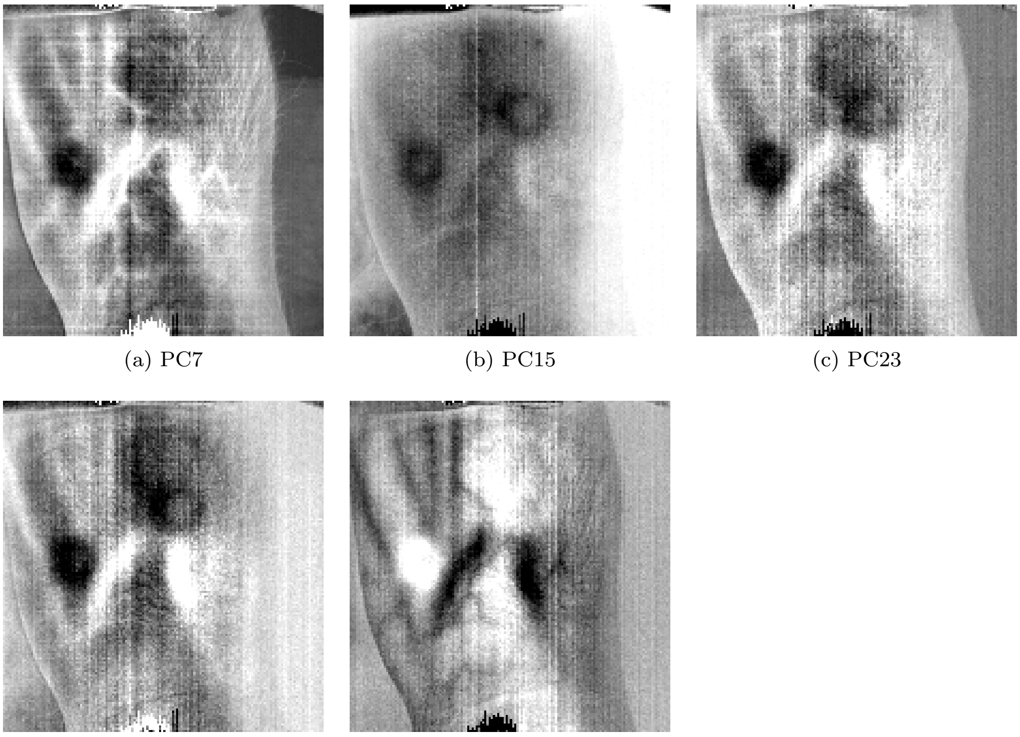

(a) PC3 (b) PC4 (c) PC107 (d) PC108 (e) PC109

Figure B.13: *Five positive results from radiance image I4-3. The principal component (PC) displayed are listed below each image.*


(a) PC7 (b) PC15 (c) PC23

(d) PC25 (e) PC29

Figure B.14: *Five positive results from radiance image I2-1. The principal component (PC) displayed are listed below each image.*

An example of one of the corrupted results that led to the exclusion of I2-2 and I2-3 can be seen in Figure B.15.

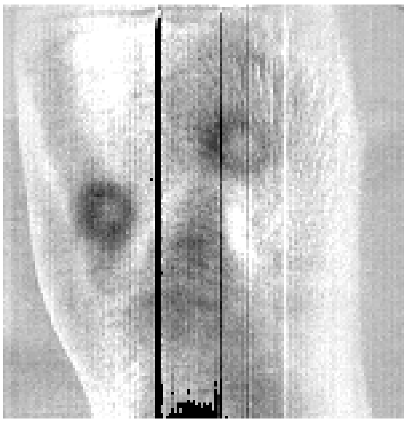


Figure B.15: *An example of a corrupted principal component (PC13) from image I2-2*

B.5 Watershed Segmentation

A larger version of the best watershed segmentation result can be seen in Figure B.16.

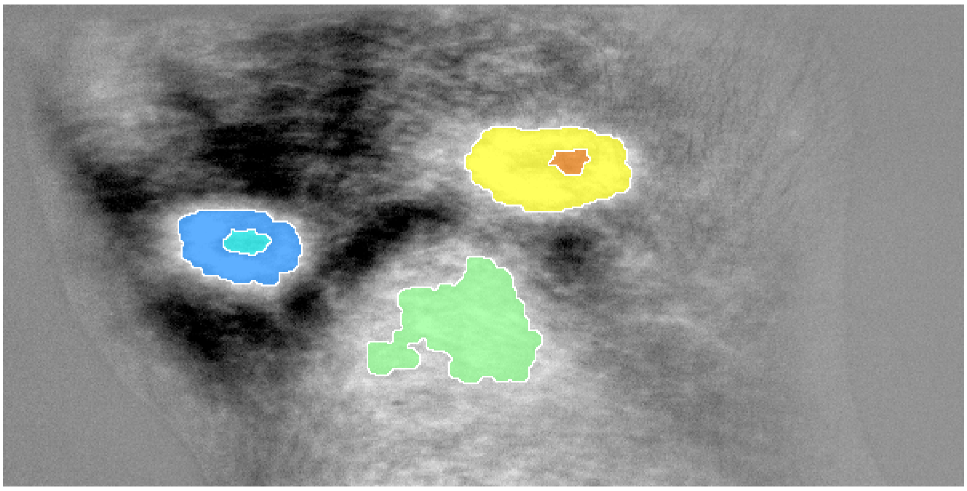


Figure B.16: *Watershed segmentation (threshold=0.7) overlay on PC4 of reflectance image I1-1.*

Appendix C

Format Descriptions

This appendix presents the file formats.

C.1 Interleave Formats

There are three common ways of organizing image data for storage in multispectral and hyperspectral images. Band Interleaved by Line (BIL), Band Interleaved by Pixel (BIP), and Band Sequential are the names of these storage schemes. Data stored like this is often preambled by metadata, most often a header file, that gives information about samples, lines, bands and interleave method. BIL, BIP, and BSQ are not iamge formats, but data storage patterns.

C.1.1 BSQ

Band sequential (BSQ) stores image points band by band. The entirety of band 1 is stored before any values of band 2 are stored and so on. This format is optimal for spatial X, Y access.

C.1.2 BIP

Band interleaved by pixel (BIP) is similar to BIL in that it alternates bands before all pixels in a band has been stored. In BIP pixel 1 of all bands are stored in sequence, followed by pixel 2 for all bands and so on. This format is optimal for spectral Z access of the data.

C.1.3 BIL

Band interleaved by line (BIL) stores image points band by band for each line (row). For example given a 500x600x160 hyperspectral image, the first 500 image points would be from the first row of band 1, the next 500 from the first row of band 2 and so on. This format is not optimized, and provides a compromise in performance between BSQ and BIP. Although this depends on application and data requested from the file.

Appendix D

Source Code

In this chapter, the source code for the algorithms principal component analysis (PCA), K-means clustering, and watershed segmentation are listed. In addition, helper functions are appended at the end of the seection.

D.1 Principal Component Analysis

The principal component analysis was implemented using the readily implemented PCA algorithm in MATLAB, and NIPALS. Both are listed in the following sections.

D.1.1 MATLAB PCA

The following listing shows the implementation using the MATLAB PCA method.

Listing D.1: PCA MATLAB Code

```matlab
function [msglength,msg] = pca( inname, outname, ncomponents )
%PCA This function will load a hyperspectral picture given by the path to
%the binary containing the data (not the header), and run PCA on it and
%store the components

% method timer start
e0 = cputime;

% attempt to read the image, calculate the PCs and write them out
% else, report an error
try
    % convert the argument to an 32-bit integer
    ncomps = int32(str2num(ncomponents));
    % read the data to X and convert it to single presicion
    % (32-bit floating point)
    X = hsiread(inname);
    X.data = single(X.data);


    % convert the two dimensional matrix of vectors to a one dimenional
    % vector of vectors. E.g. NxMxP becomes NMxP.
    X = columnize(X);

    % calculate the principal components using the MATLAB method
    [COEFF,SCORE] = princomp(X.data,'econ');
    X.data = SCORE;
    X.score = COEFF;

    % convert the data back to NxMxP from NMxP
    X = decolumnize(X);
    % normalise the data between 0 and 65535 and store it as unsigned
    % 16-bit integers
    X.data = uint16(normim(X.data,0.0,65535.0));

    % remove the pseudo color default bands from the image header
    if isfield(X.hdr,'defbands')
        X.hdr=rmfield(X.hdr,'defbands');
    end

    % if the number of requested components is smaller than the data
    % then limit the calculated data
    if ncomps<X.hdr.bands
        X.data = X.data(:,:,1:ncomps);
        X.hdr.bands = int32(ncomps);
    end

    % write the data to the give filename
    hsiwrite(X,outname);

    % return a tag formated message to the calling method
    msg = '<error>false</error>';
    msg = [msg,'<numprincomps>',ncomponents,'</numprincomps>'];
catch
    msg = '<error>true</error>';
    msg = [msg,'<lasterror>',lasterr,'</lasterror>'];
end

% append time elapsed to the output message
e0 = cputime - e0;
msg = [msg,'<elapsed>',num2str(e0),'</elapsed>'];
msglength = int32(numel(msg));
```

D.1.2 NIPALS

The following listing shows the implemented NIPALS algorithm for approximating the PCs.

Listing D.2: NIPALS for PCA MATLAB Code

```matlab
%function [ outname, nim, nim_cropped ] = pcaNipals( inname, n_comps )
%function outname = pcaNipals( inname, n_comps )
function pcaNipals( inname, n_comps )
%PCANIPALS Function for loading and performing PCA on an ENVI hyperspectral
%image.
%Arguments:
%   inname <string> - input filename with path of bin (not hdr)
%   outname <string> - output filename of image
%   nim <cell> - output loadings
%   nim_cropped <cell> - output cropped loadings

%% Quick arguments check
if nargin ~= 2
    error('Usage: pcaNipals(binaryfilename,num_components)');
end;

%% Load
disp('Step 1: Reading data');
try
    [i1,d00,dtype,outname] = hyperload(inname);
    clear d00 dtype;
catch
    disp('An error occured. pcaNipals aborting.');
    %s = lasterror
    return;
end

dims = size(i1);
%disp('Status:        File read');
disp(['   Dimensions:  ',num2str(dims(1)),'x',num2str(dims(2)),'x',num2str(dims(3))]);


%% Columnize
disp('Step 2: Columnize data');
%X = myim2col(i1);

t0 = 1;
X = zeros(dims(1)*dims(2),dims(3));
for i=1:dims(1)
    for j=1:dims(2)
        X(t0,:) = i1(i,j,:);
        t0 = t0+1;
    end;
end;
```

```

clear filename i j t0 i1;

%% NIPALS
disp('Step 3: NIPALS iterations');
dim = size(X); %dimensions of dataset
%n_comps = 15; %number of principal components
if ischar(n_comps)
    n_comps = str2double(n_comps);
end
scores = zeros(dim(2), n_comps); %score matrix (only as large as needed)
loadings = zeros(dim(1), n_comps); %loadings matrix (only as large as needed)
err = 1; %current error init
limit = 10^-7; %the cut off limit
%disp('Starting NIPALS iterations');

tid3 = cputime;
for i=1:n_comps
    tid1 = cputime; %timer for estimating runtime of iterations
    u = X(1,:); %setting u to a row vector of X
    while(err>limit) %run until error is approaching zero
        v = (X*u')/(u*u'); %project X onto u to find corresponding loading v
        v = v/norm(v); %normalize the length of loading vector v to 1.0
        uold = u; %store the score vector u into uold
        u = (X'*v)/(v'*v); %project the matrix X onto v to find score vector u
        u = u'; %transpose u from 1xn to nx1 (could have done this in previous step), but easier to understand code for others
        d = uold-u; %difference between the previous scores and the current scores
        err = norm(d); %convergence of scores
        %disp(['Error is: ',num2str(err)]);
    end;
    scores(:,i) = u;
    loadings(:,i) = v;
    err = 1;
    %X = X-(v*u);

    % matmult --> Out(i,j) = sum(r=1,n)( a(i,r)*b(r,j) )
    dv = size(v);
    mat_m = dim(1);
    mat_n = 1;
    mat_p = dim(2);
    rvec = 1:mat_n;
    %aj=1:mat_p;
    ai=1:mat_m;
    tid2 = cputime;
    %for ai=1:mat_m
    for aj=1:mat_p
        resm = (v(ai,rvec)*u(rvec,aj));
        X(ai,aj) = X(ai,aj) - resm;
    end;
    tid2 = cputime - tid2;

    pr = (i/n_comps) * 100;
    disp(['   ',num2str(pr),'% complete (',num2str(i),'/',num2str(n_comps),')']);
    tid1 = cputime - tid1;
    disp(['   One iteration took ',num2str((tid1)),' seconds.']);
    disp(['   Of this, matmult used ',num2str((tid2)),' seconds.']);
end;
tid3 = cputime - tid3;
disp(['   Total iteration time was ',num2str((tid3)),' seconds.']);

%% Cleaning
disp('Step 4: Cleaning');
clear ai aj an d dim dv err i limit mat_m mat_n mat_p pr rvec u uold v %resm

%% Output / utputt
disp('Step 5: Writing components to PNG');
close all;
nim = cell(n_comps);
nim_cropped = cell(n_comps);
for ai=1:n_comps
    temp1 = normim(mycol2im(loadings(:,ai),[dims(1) dims(2) 1]));
    %hardcoded cutting of spectralon
    temp2 = normim(temp1(220:1:dims(1),:));

    nim{ai} = temp1;
    nim_cropped{ai} = temp2;

    outfile = [inname,'outputCropped',num2str(ai),'.png'];
    imwrite(temp2,outfile,'PNG');
    outfile = [inname,'outputUncropped',num2str(ai),'.png'];
    imwrite(temp1,outfile,'PNG');
end;
%disp('Results written.');
clear ai outfile temp1 temp2;
```

D.2 K-means Clustering

The Listing D.3 shows the implementation of K-means clustering using the MATLAB kmeans method. Listing D.4 shows an example usage of the code in MATLAB.

Listing D.3: Clustering MATLAB Code

```matlab
function [clustCell,imgCell,descCell] = clustering(filename, clusters, iterations, bands)
% CLUSTERING
%   filename - path to the hyperspectral ENVI file
%   clusters - a parameter vector of cluster numbers
%   iterations - a parameter vector of iteration numbers
%   bands - vector of what bands in the input image should be used
input = hsiread(filename);
counter = 1;

for clus=clusters
    %pick out the data from the input that should be used
    data = input.data(:,:,bands);
    dims1 = size(data);
    for iter=iterations;
        runfor = 1;
        % a loop that will keep trying the current parameters
        % until they work (matlab clustering is temperamental)
        while runfor
            try
                %the kmeans results are stored in the cluster cell
                clustCell{counter}=kmeans(single(myim2col(data)), clus,...
                    'maxiter',iter,'display','iter');

                %the cluster results are coloured
                imgCell{counter}=label2rgb( mycol2im( clustCell{counter},...
                    [dims1(1) dims1(2) 1]));

                %COMMENTED OUT the cluster results are shown
                %figure, imshow(imgCell{counter});

                %the parameters are stored with the same index
                descCell{counter}=['Iterations=',num2str(iter),...
                    '-Clusters=',num2str(clus)];

                %if all this goes well, then end the while-loop
                runfor = 0;

                %increment the counter
                counter = counter + 1;
            catch
```

```
                lasterr
                runfor = 1;
            end;
        end;
    end;
end;
```

Listing D.4: Clustering MATLAB Code Usage

```
>> [clusterCell,colourImageCell,descriptionCell] = clustering('c:\
    prosjekt\datasett\VNIR\72h',[4 8],[10 100],[38 47]);
>> figure, imshow(colourImageCell{1},[]);
>> figure, imshow(colourImageCell{2},[]);
```

D.3 Watershed Segmentation

The Listing D.5 shows the implementation of watershed segmentation using the MATLAB watershed transform. Listing D.4 shows an example usage of the code in MATLAB. As implemented, the results will be written to image files on the harddrive in the location given by the prefix input.

Listing D.5: Watershed Segmentation MATLAB Code

```
function wshed(filename,band,outputprefix,invert,altlevel)
%WSHED watershed script core function
%
% Usage: wshed(filename,band,outputprefix)
% -->filename - path to the hyperspectral binary in ENVI format
% -->band - the band in the hyperspectral image that will be used
    as a
% greyscale input
% -->outputprefix - the path and filenameprefix of the outputed
    PNGs
% -->invert - invert the picture? 1 or 0
%
% This function will perform application adjusted watershed
    segmentation
% after mathematical morphology operations on the original image.

if nargin~=4 && nargin~=5
    error('ArgumentError');
end;
%
% Read the hyperspectral image (might be slow)
I = hsiread(filename);

%
% Get the band we wish to process
```

```
orgImage(:,:) = normim(I.data(200:end,:,band)); %with crop
%orgImage(:,:) = normim(I.data(:,:,band)); %without crop

%
% Adjust the image contrast
% (without loosing range, still double [0-1])
adjImage = imadjust(orgImage,stretchlim(orgImage),[]);
if invert
    adjImage = imcomplement(adjImage);
end

%
% Threshold the image
level = 0.75; %default threshold value
if nargin==5
    level = altlevel;
end
bwAdjImage = im2bw(adjImage, level);

%
% Mathematical morpholpgy
%structural element, a disk with radius of 6
radi = 6;
sqsize = 3;
seDisk = strel('disk',radi);
seSquare = ones(sqsize,sqsize);
%opening the image with struct.elem.
openImage = imopen(bwAdjImage,seDisk);
%masking the image (no pixels added, only removed)
openImage = openImage & bwAdjImage;
%Beucher-gradient
beucherImage = imdilate(openImage,seSquare) - imerode(openImage,seSquare);

%
% Watershed Segmentation (labeling of regions in binary image, trivial
% usage)
distImage = bwdist(~beucherImage);
waterImage = watershed(distImage);
labelImage = label2rgb(waterImage,'jet');

imwrite(normim(orgImage),[outputprefix,'_output1_original.png'],'PNG');
imwrite(normim(adjImage),[outputprefix,'_output2_adjusted.png'],'PNG');
imwrite(normim(bwAdjImage),[outputprefix,'_output3_thresholded(',num2str(level),...
    ').png'],'PNG');
```

```
imwrite(normim(openImage),[outputprefix,'_output4_thresholded(',num2str(level),...
    ')_morphOpened(disk,radius',num2str(radi),...
    ').png'],'PNG');
imwrite(normim(beucherImage),[outputprefix,'_output5_thresholded(',num2str(level),...
    ')_morphOpened(disk,radius',num2str(radi),...
    ')_morphBeucher(square,size',num2str(sqsize),...
    ').png'],'PNG');
imwrite(normim(distImage),[outputprefix,'_output6_thresholded(',num2str(level),...
    ')_morphOpened(disk,radius',num2str(radi),...
    ')_morphBeucher(square,size',num2str(sqsize),...
    ')_bwDistance',...
    '.png'],'PNG');
imwrite(normim(waterImage),[outputprefix,'_output7_thresholded(',num2str(level),...
    ')_morphOpened(disk,radius',num2str(radi),...
    ')_morphBeucher(square,size',num2str(sqsize),...
    ')_bwDistance_watershed',...
    '.png'],'PNG');
imwrite(labelImage,[outputprefix,'_output8_thresholded(',num2str(level),...
    ')_morphOpened(disk,radius',num2str(radi),...
    ')_morphBeucher(square,size',num2str(sqsize),...
    ')_bwDistance_watershed_rgbLabled',...
    '.png'],'PNG');
```

Listing D.6: Watershed MATLAB Code Usage

```
>> wshed('c:/prosjekt/datasett/VNIR_reflectance/72h_reflectance',... 47,'c:/prosjekt/png_wshed/72h_reflectance');
```

D.4 Helper Functions

This section presents the code implemented as a framework around, or helper functions of the experimentation. This includes funcationality such as reading and loading the hyperspectral data, reordering the hyperspectral data from NxMxP to NMxP (called columnize and decolumnize), and other structural implementations.

D.4.1 ENVI Image File Reader

The following listing reads the ENVI data from the given filename into a hyperspectral image structure, called HSI in the code.

Listing D.7: ENVI Image File Reader Code

```
function [ X ] = hsiread( filename )
```

```
%HSIREAD This function will read a hyperspectral image of the ENVI format
%given a valid path to the flat binary file, not the header. The header
%will be found and parsed first

% Initial data setup
X.filepath = filename;
X.isDataRead = 0;
X.isHeaderRead = 0;

% Control of validity for variable ‘filepath‘
if ~ischar(filename)
    error('Error 400: Filename is not a char array.');
else
    %disp(['Loading file ',filename]);
end

% Existance of input file
X.headerpath = strcat(filename,'.hdr');
rfilehandle = fopen(X.headerpath,'r');
reduced = filename(1:(max(size(filename))-4));
if rfilehandle == -1
    X.headerpath = strcat(reduced,'.hdr');
    rfilehandle = fopen(X.headerpath,'r');
    if rfilehandle == -1
        error('Error 404: Header file not found.');
    end;
end;

% Read header
X.hdr = hsiheaderread(X.headerpath,rfilehandle);
X.isHeaderRead = 1;
fclose(rfilehandle);
% Allocate memory
X.volume = prod(X.hdr.dims);
dims = X.hdr.dims;
X.data = zeros(dims(2),dims(1),dims(3),X.hdr.dtype);
% Read data
X.columnized = false;

filehandle = fopen(filename);
switch X.hdr.interleave
    case 'bsq'
        for alpha=1:1:dims(3)
            temp = fread(filehandle, (dims(1)*dims(2)), X.hdr.dtype);
            teller = 1;
            for x=1:1:dims(2)
```

```
                for y=1:1:dims(1)
                    X.data(x,y,alpha) = temp(teller);
                    teller = teller + 1;
                end
            end
        end
    case 'bip'
        for x=1:1:dims(2)
            temp = fread(filehandle, (dims(1)*dims(3)), X.hdr.dtype);
            teller = 1;
            for y=1:1:dims(1)
                for alpha=1:1:dims(3)
                    X.data(x,y,alpha) = temp(teller);
                    teller = teller + 1;
                end
            end
        end
    case 'bil'
        for x=1:1:dims(2)
            temp = fread(filehandle, (dims(1)*dims(3)), X.hdr.dtype);
            teller = 1;
            for alpha=1:1:dims(3)
                for y=1:1:dims(1)
                    X.data(x,y,alpha) = temp(teller);
                    teller = teller + 1;
                end
            end
        end
    otherwise
        error(['error: type was', interleave]);
end
X.isDataRead = 1;
end

%% INTERNAL FUNCTIONS
function [hdr] = hsiheaderread( headerpath, fh )
%HSIHEADERREAD This internal function will read the header of the
%hyperspectral image and return a struct containing the read data.
%tags = {'samples', 'lines','bands','header offset','file type', 'data type', 'interleave', 'byte order', 'x start', 'y start', 'default bands'};
datatypes = {'bit8' 'int16' 'int32' 'float32' 'float64' 'uint16' 'uint32' 'int64' 'uint64'};
hdr.dims = [0 0 0];
while 1
    line = fgetl(fh);
    if ~ischar(line)
```

```
        break
    end
    [head,tail]=strtok(line,'=');

    %remove tailing whitespace
    head = strtrim(head);

    switch head
        case 'samples'
            [f,s] = strtok(tail);
            hdr.dims(1) = str2double(s);
            hdr.samples = str2double(s);
        case 'lines'
            [f,s] = strtok(tail);
            hdr.dims(2) = str2double(s);
            hdr.lines = str2double(s);
        case 'bands'
            [f,s] = strtok(tail);
            hdr.dims(3) = str2double(s);
            hdr.bands = str2double(s);
        case 'header offset'
            [f,s] = strtok(tail);
            hdr.offset = str2double(s);
        case 'file type'
            [f,s] = strtok(tail);
            hdr.ftype = strtrim(s);
        case 'interleave'
            [f,s] = strtok(tail);
            hdr.interleave = strtrim(s);
        case 'byte order'
            [f,s] = strtok(tail);
            hdr.byteorder = strtrim(s);
        case 'x start'
            [f,s] = strtok(tail);
            hdr.xstart = str2double(s);
        case 'y start'
            [f,s] = strtok(tail);
            hdr.ystart = str2double(s);
        case 'data type'
            [f,s] = strtok(tail);
            hdr.dtype = str2double(s);
            hdr.dtypenumber = hdr.dtype;
            switch hdr.dtype
                case 1
                    hdr.dtype = datatypes(1); %8-bit byte
                case 2
                    hdr.dtype = datatypes(2); %16-bit signed integer
                case 3
```

```
                    hdr.dtype = datatypes(3); %32-bit signed long integer
                case 4
                    hdr.dtype = datatypes(4); %32-bit floating point
                case 5
                    hdr.dtype = datatypes(5); %64- bit double precision floating point
                case 6
                    error('2x32-bit complex not supported');
                    %2x32-bit complex, real-imaginary pair of double precision
                case 9
                    error('2x64-bit double precision complex not supported');
                    %2x64-bit double precision complex, real-imaginary pair of double precision
                case 12
                    hdr.dtype = datatypes(6); %16-bit unsigned integer
                case 13
                    hdr.dtype = datatypes(7); %32-bit unsigned long integer
                case 14
                    hdr.dtype = datatypes(8); %64-bit unsigned integer
                case 15
                    hdr.dtype = datatypes(9); %64-bit unsigned long integer
                otherwise
                    error('unknown data type');
            end
            hdr.dtype = hdr.dtype{1,1}; %make dtype into char from cell (like casting)
        case 'description'
            hdr.description = readInsideBracket(tail,fh);
        case 'wavelength'
            hdr.wavelength = readInsideBracket(tail,fh);
            [s1 s2 s3 s4 s5] = regexp(hdr.wavelength,'[{]{1}\s*([0-9]+[.]{1}[0-9]+)');
            hdr.minfreq = str2double( s5{1}{1} );
            [s1 s2 s3 s4 s5] = regexp(hdr.wavelength,'([0-9]+[.]{1}[0-9]+)\s*[}]{1}');
            hdr.maxfreq = str2double( s5{1}{1} );
            clear s1 s2 s3 s4 s5;
        case 'default bands'
            hdr.defbands = readInsideBracket(tail,fh);
        otherwise
    end
```

```
end
end


function s = readInsideBracket(li,fileh)
dOpen = 1;
s = li(2:end);
if ~isempty(strfind(s,'}'))
    dOpen = 0;
end
while dOpen
    line = fgetl(fileh);
    if ~ischar(line)
        error('D: met end of file inside open bracket. misformed
            header file.');
    end
    %check if end of description section
    s = strcat(s,line);
    if ~isempty(strfind(line,'}'))
        dOpen = 0;
    end
end
end
```

D.4.2 ENVI Image File Writer

The following listing writes the given ENVI data to the filename.

Listing D.8: ENVI Image File Writer Code

```
function hsiwrite( hsi, outfilename )
%HSIWRITE Will write the given @hsi to the path and filename given
     by
%@outfilename.

%%%%%%%%%%%%%%%%%% VENI - i came (aka prepare to write)
temp1 = fix(clock);
temp2 = [date,'-',num2str(temp1(4)),'-',num2str(temp1(5)),'-',
    num2str(temp1(6))];

if nargin==1
    outfilename = [hsi.filepath,'-matlab-',temp2];
    %disp(['Outfilename set to: ',outfilename]);
elseif nargin==2
    %disp(['Outfilename input: ',outfilename]);
else
    error('Error-ArgumentNumberIncorrect');
end

```

```
dataFilename = outfilename;
headerFilename = [outfilename,'.hdr'];

clear outfilename temp1 temp2; %cleaning :)

%%%%%%%%%%%%%%%%%%%% VIDI - i saw (assemble the header)
s1 = hsiwriteheader(hsi.hdr,headerFilename);
if ~s1
    error('Error-WritingHyperspectralHeader');
end

%%%%%%%%%%%%%%%%%%%% VICI - i conquered (write the data)
try
    multibandwrite(uint16(hsi.data),dataFilename,'bil');
    %     fhb = fopen(dataFilename,'w'); %empty the file for writing or create new file
    %     dims = hsi.hdr.dims;
    %     switch hsi.hdr.interleave
    %         case 'bsq'
    %             for alpha=1:1:dims(3)
    %                 for x=1:1:dims(2)
    %                     for y=1:1:dims(1)
    %                         fwrite(fhb,hsi.data(x,y,alpha),'uint16');
    %                     end
    %                 end
    %             end
    %         case 'bip'
    %             for x=1:1:dims(2)
    %                 for y=1:1:dims(1)
    %                     for alpha=1:1:dims(3)
    %                         fwrite(fhb,hsi.data(x,y,alpha),'uint16');
    %                     end
    %                 end
    %             end
    %         case 'bil'
    %             for x=1:1:dims(2)
    %                 for alpha=1:1:dims(3)
    %                     for y=1:1:dims(1)
    %                         fwrite(fhb,hsi.data(x,y,alpha),'uint16');
    %                     end
    %                 end
    %             end
    %         otherwise
    %             error(['error 3 (not known interleave): type was ', interleave]);
    %     end
    %      fclose(fhb);
    fclose('all');
catch
    error('Error-WritingHyperspectralData');
end
end
%%%%%%%%%%%%%%%%%%%% interal functions
function success = hsiwriteheader( hdr, hdrfile )
try
    fh = fopen(hdrfile,'w'); %empty the file for writing or create new file

    fprintf(fh,'%s\r\n','ENVI');
    if isfield(hdr,'description')
        fprintf(fh,'description = %s\r\n',  hdr.description);
    end
    if isfield(hdr,'dims')
        fprintf(fh,'samples = %d\r\n',      hdr.dims(1));
        fprintf(fh,'lines = %d\r\n',        hdr.dims(2));
        fprintf(fh,'bands = %d\r\n',        hdr.dims(3));
    end
    if isfield(hdr,'offset')
        fprintf(fh,'header offset = %d\r\n',hdr.offset);
    end
    if isfield(hdr,'ftype')
        fprintf(fh,'file type = %s\r\n',    hdr.ftype);
    end
    if isfield(hdr,'interleave')
        fprintf(fh,'interleave = %s\r\n',   hdr.interleave);
    end
    if isfield(hdr,'byteorder')
        fprintf(fh,'byte order = %s\r\n',   hdr.byteorder);
    end
    if isfield(hdr,'xstart')
        fprintf(fh,'x start = %d\r\n',      hdr.xstart);
    end
    if isfield(hdr,'ystart')
        fprintf(fh,'y start = %d\r\n',      hdr.ystart);
    end
    if isfield(hdr,'dtypenumber')
        fprintf(fh,'data type = %d\r\n',    hdr.dtypenumber);
    end
    if isfield(hdr,'defbands')
        fprintf(fh,'default bands = %s\r\n',hdr.defbands);
    end
    if isfield(hdr,'wavelength')
        fprintf(fh,'wavelength = %s\r\n',   hdr.wavelength);
    end
    fclose(fh); %close it
    success = true; %all good so far, then complete
catch
    success = false;
end
end
```

D.4.3 Image Columnizer and Decolumnizer

The fist following listing shows the code for the reorganisation of the data from a matrix of pixel vectors into a single row vector of pixel vectors. The second listing shows the code for converting the row vector back into the inital spatial image span.

Listing D.9: Image Reshaping Code - Columnize

```
function [ hsi ] = columnize( hsi )
% DECOLUMNIZE will decolumnize any 3d image
% i.e. a 300000x160 will become 500x600x160
dims = size(hsi.data);
if ~hsi.columnized
    t0 = 1;
    if isa(hsi.data,'uint16')
        ot = zeros(dims(1)*dims(2),dims(3),'uint16');
    elseif isa(hsi.data,'single')
        ot = zeros(dims(1)*dims(2),dims(3),'single');
    else
        ot = zeros(dims(1)*dims(2),dims(3),'double');
    end;
    for i=1:dims(1)
        for j=1:dims(2)
            ot(t0,:) = hsi.data(i,j,:);
            t0 = t0+1;
        end;
    end;
    hsi.data = ot;
    clear ot;
    hsi.columnized = true;
end
```

Listing D.10: Image Reshaping Code - Decolumnize

```
function [ hsi ] = decolumnize( hsi )
% DECOLUMNIZE will decolumnize any 3d image
% i.e. a 300000x160 will become 500x600x160
t0 = 1;
if hsi.columnized
    dims = [hsi.hdr.dims(2) hsi.hdr.dims(1) hsi.hdr.dims(3)] ;
    output = zeros(dims(1),dims(2),dims(3),'single');
    for i=1:dims(1)
        for j=1:dims(2)
```

```
            output(i,j,:) = hsi.data(t0,:);
            t0 = t0+1;
        end;
    end;
    hsi.data = output;
    clear output;
    hsi.columnized = false;
end
```

D.4.4 ENVI Image File Crop

The following listing cropes the loaded hyperspectral image.

Listing D.11: Image Crop Code

```
function [ hsi ] = hsicrop( hsi, yrange, xrange, zrange )
%HSICROP
% Crops the hyperspectral structure and updates the fields
hsi.data = hsi.data(yrange,xrange,zrange);
hsi.hdr.dims = [numel(xrange) numel(yrange) numel(zrange)];
hsi.hdr.samples = hsi.hdr.dims(2);
hsi.hdr.lines = hsi.hdr.dims(1);
hsi.hdr.bands = hsi.hdr.dims(3);
hsi.volume = prod(hsi.hdr.dims);
```

D.4.5 ENVI Image File Viewer

The following listing shows the code for an implemented function for viewing bands of the hyperspectral image in MATLAB.

Listing D.12: Image Viewer Code

```
function hsishow( hsi , layer, showAlsoInverted )
%HSISHOW

switch nargin
    case 2
        if ~isstruct(hsi) || ~isnumeric(layer)
            error('TwoInputsFormatError');
        end
        showAlsoInverted = false;
    case 3
        if ~isstruct(hsi) || ~isnumeric(layer) || ~islogical(showAlsoInverted)
            error('ThreeInputsFormatError');
        end
    otherwise
        error('ArgumentNumberError');
end
```

```

picture = normim(hsi.data(:,:,layer));
picture_adjusted = imadjust(picture,stretchlim(picture),[]);
picture_inverted = 1.0 - picture;
picture_adjusted_inverted = 1.0 - picture_adjusted;
%picture_cropped = normim(hsi.data(94:end,:,layer));
%picture_cropped_inverted = 1.0 - picture_cropped;

r0 = [min(picture(:)) max(picture(:))];
r1 = [min(picture_inverted(:)) max(picture_inverted(:))];

r2 = [min(picture_adjusted(:)) max(picture_adjusted(:))];
r3 = [min(picture_adjusted_inverted(:)) max(picture_adjusted_inverted(:))];

fi1 = figure;
imshow(picture, r0);
title(['Picture (normal) band ',num2str(layer)]);
fi2 = figure;
imshow(picture_adjusted, r2);
title(['Picture (normal adjusted 2%) band ',num2str(layer)]);

if showAlsoInverted
    fi3=figure;
    imshow(picture_inverted, r1);
    title(['Picture (inverted) band ',num2str(layer)]);
    fi4=figure;
    imshow(picture_adjusted_inverted, r3);
    title(['Picture (inverted adjusted 2%) band ',num2str(layer)]);
    waitfor(fi3);
    waitfor(fi4);
end
waitfor(fi2);
waitfor(fi1);
```