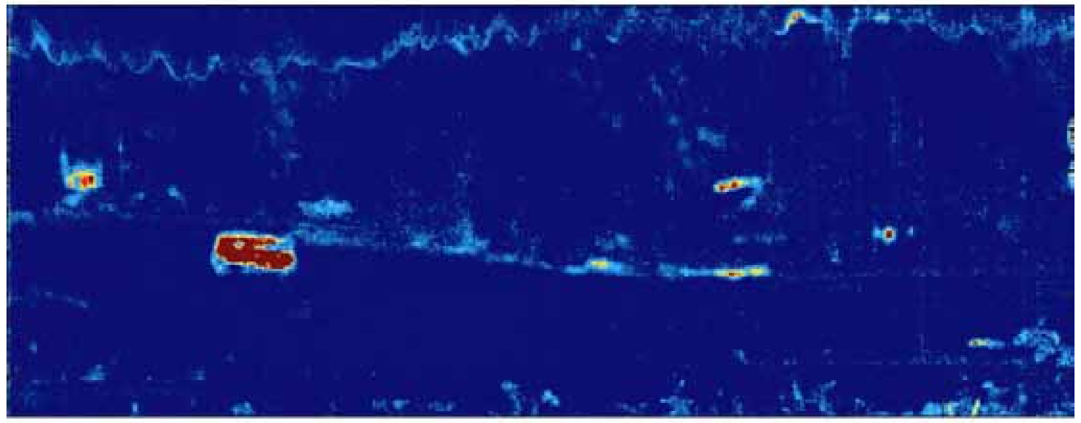
FOI
Defence Research Agency
Sensor Technology
P.O. Box 1165
SE-581 11 Linköping

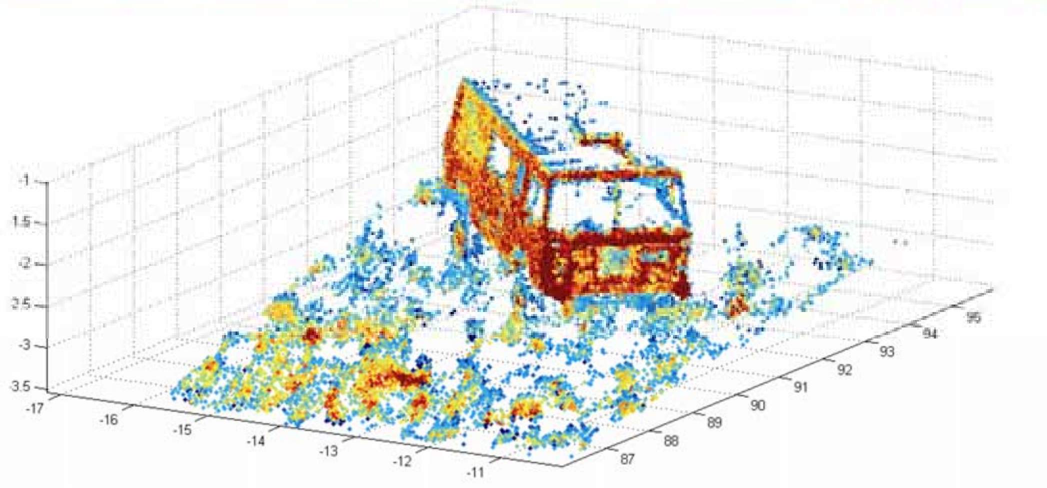
Phone: +46 13 37 80 00
Fax: +46 13 37 81 00

www.foi.se

Visualisation and detection using 3-D laser radar and hyperspectral sensor

CHRISTINA FREYHULT





FOI-R--2125 --SE Technical report
ISSN 1650-1942 November 2006

Sensor Technology

Christina Freyhult

Visualisation and detection using 3-D laser radar and hyperspectral sensor

FOI-R--2125--SE Technical report Sensor Technology
ISSN 1650-1942 November 2006

Issuing organization
FOI – Swedish Defence Research Agency

Sensor Technology
P.O. Box 1165
SE-581 11 Linköping

Report number, ISRN
FOI-R--2125--SE

Report type
Technical report

Research area code
4. C4ISTAR

Month year
November 2006

Project no.
E3080, E3082

Sub area code
42 Above water Surveillance, Target acquistion and Reconnaissance

Sub area code 2

Author/s (editor/s)
Christina Freyhult

Project manager
Tomas Chevalier, Jörgen Ahlberg

Approved by
Mattias Severin

Sponsoring agency
Swedish Armed Forces

Scientifically and technically responsible
Jörgen Ahlberg

Report title
Visualisation and detection using 3-D laser radar and hyperspectral sensor

Abstract

The main goal of this thesis is to show the strength of combining datasets from two different types of sensors. The sensors used in this thesis are a hyperspectral camera and a scanning 3-D laser. The hyperspectral camera is used for detection and the laser for recognition/visualisation.

The project can be divided into two main parts. The first part discusses the properties of one of the datasets and how these are used to isolate anomalies. An issue to deal with here is not only what properties to look at, but how to make the process automatic. The information retained from the first dataset is then used to make intelligent choices in the second dataset. Again, one of the challenges is to make this process automatic and accurate. The second part of the project consists of presenting the results in a way that gives the most information to the user. This is done with a graphical user interface that allows the user to manipulate the way the result is presented.

The conclusion of this project is that the information from the combined sensor datasets gives better results than the sum of the information from the individual datasets.

Keywords
laser radar, hyperspectral, automatic target detection, sensor fusion, visualization

Further bibliographic information

Language English

ISSN 1650-1942

Pages 35 p.

Price acc. to pricelist

Utgivare FOI - Totalförsvarets forskningsinstitut Sensorteknik Box 1165 581 11 Linköping	**Rapportnummer, ISRN** FOI-R--2125--SE	**Klassificering** Teknisk rapport
	Forskningsområde 4. Ledning, informationsteknik och sensorer	
	Månad, år November 2006	**Projektnummer** E3080, E3082
	Delområde 42 Spaningssensorer	
	Delområde 2	

Författare/redaktör Christina Freyhult	**Projektledare** Tomas Chevalier, Jörgen Ahlberg
	Godkänd av Mattias Severin
	Uppdragsgivare/kundbeteckning Försvarsmakten
	Tekniskt och/eller vetenskapligt ansvarig Jörgen Ahlberg

Rapportens titel

Visualisering och detektion genom användning av 3D-laserradar och hyperspektral sensor

Sammanfattning

Målet med detta examensarbete var att visa på styrkan av att kombinera datamängder från två olika typer av sensorer. Sensorerna som används i detta examensarbete är en hyperspektral kamera och en skannande 3D-laser. Den hyperspektrala kameran användes för detektion och lasern för igenkänning/visualisering.

Projektet kan delas upp i två huvuddelar. Den första delen diskuterar egenskaper av endera datamängderna och hur dessa kan användas för att isolera anomalier. Informationen som tas från första datamängden används för att göra ett smart val i den andra datamängden. Ett ämne som tas upp i båda de första stegen är inte bara vilka egenskaper som ska analyseras, men även hur processen kan göras automatisk. Den andra halvan av projektet består av att presentera resultatet från första delen på ett sätt som ger mest information till användaren. Detta görs med hjälp av ett grafiskt gränssnitt som tillåter användaren att styra över sättet som resultatet presenteras.

Slutsatsen av detta projekt är att informationen som man får från samverkande sensorers datamängder ger bättre resultat än summan av information som får från individuella sensorers datamängder.

Nyckelord

laserradar, hyperspektral, automatisk måldetektion, sensorfusion, visualisering

Övriga bibliografiska uppgifter	**Språk** Engelska
ISSN 1650-1942	**Antal sidor:** 35 s.
Distribution enligt missiv	**Pris: Enligt prislista**

Contents

Table of figures

1 Introduction

1.1 Background

The Swedish Armed Forces use a variety of sensors for surveillance and reconnaissance purposes. Many times, the use of a certain sensor will give an abundance of information concerning a particular property of the scene, but will fail to give satisfying information about other properties. These other properties can be captured by additional sensors. Hence, to capture all the desired properties of a scene, a collection of different sensors is needed.

1.2 Problem

While several different sensors may give a wide range of data for a certain scene, the combination of all that data will be hard to understand or to use, because even if the information is interrelated, we do not know how. This can be resolved by *registering* the data from the sensors i.e., to establish a sample-to-sample correspondence between the two data sets. When the data is registered it still takes time, effort, and knowledge to make any sense of all the data. This is what the thesis will try to resolve by creating a program that, using automatic registration, uses a property of one dataset to automatically extract further information from the remaining datasets. The thesis will also resolve the problem of automatically selecting parts of the data that are of interest of the user, without supervision.

1.3 Purpose

The goal of this thesis is to present a program that illustrates the benefits of using interacting sensors. It will simulate that the sensors are directing and directed, so that one sensor records and processes data, and, using the information from that first step, directs the second sensor to record only in specific areas. This way, the final result will be data that has a known relation. The anomalies looked for are vehicles hidden in a forest scenario. Using the data from two sensors recording the same scene, the program will also demonstrate the result in a straightforward manner that does not demand any calculations from the user, only basic understanding of the concepts involved.

2 Sensors and data acquisition

To allow this project to proceed beyond theoretical matter, a field measurement campaign was conducted in the early stages of the project. The collected data would be a basis on which to test theories and the program. In our work, the vehicles placed in the scene would constitute the anomalies to find in the datasets.

2.1 Field measurments

2.1.1 Location

FOI has access to an area near Ströplahult called *P4/Kvarn*, at the Army Combat School (MSS). The sensors were placed facing a small hill composed of an open space and forest. This setting allowed for acquiring data over varying terrain and thereby giving the data more versatility.

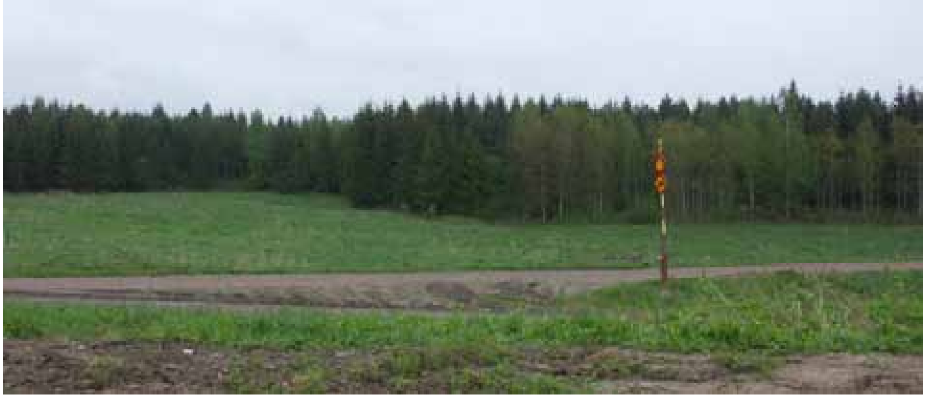


Figure 2.1 The location of the data acquisition at Kvarn.

2.1.2 Targets

For the purpose of the study several vehicles were lent to FOI by MSS, to be placed in different positions throughout our chosen location. The data collection was conducted in such a manner that different scenarios were recorded. Examples of the variations are the different placements of the vehicles in the scene, as well as the number of targets present at one time. A written record of the various scenes was made, containing the positions of the targets for later reference [5]. Aside from the realistic scenes to be used for the program testing, data was collected of some chequered boards in the scene for sensor registration purposes.

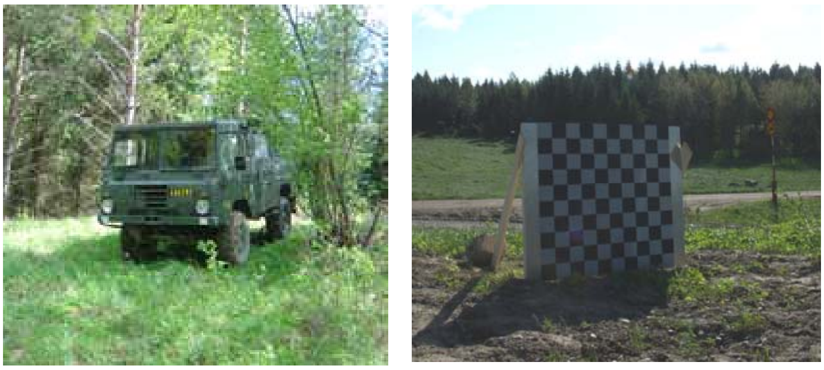


Figure 2.2 Left: One of the vehicles used as targets in the scene.
Right: The chequered board used for registration.

2.1.3 The sensors

The two sensors used in this data collection were a hyperspectral camera and a range sensing 3-D laser to measure point’s positions in 3-D. The sensors were placed as close to each other as possible, at the same height and facing the same direction, so that the data would have maximum overlap. They were fixed in this position and held the same settings through the measurements. More extensive data about the sensors can be found in [5] and [6].

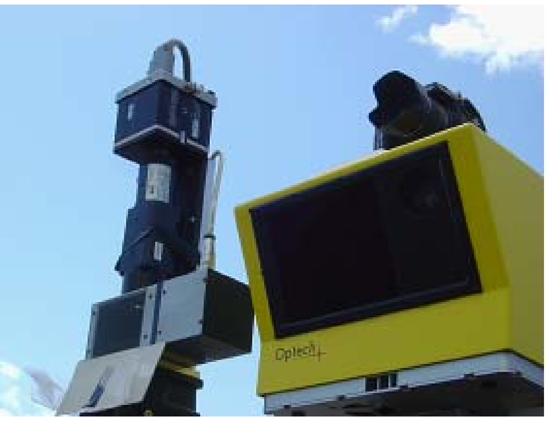


Figure 2.3 The hyperspectral (left) and 3-D laser sensor (right).

2.2 Hyperspectral sensor

2.2.1 The sensor

Imspec, from the Finnish company Specim, is a sensor that is used to aquire hyperspectral images of the incoming light, spanning from the visual light to the near infrared area (NIR). It is called hyperspectral since it aquires hundreds of wavelengths of light, spanning from 396 to 961 nm in the electromagnetic spectrum. This can be compared to a consumer camera that aquires only three bands of wavelengths that correspond to the red, green, and blue colours.

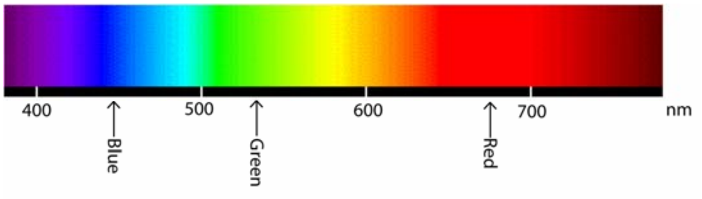


Figure 2.4 The wavelengths recorded by a consumer camera.

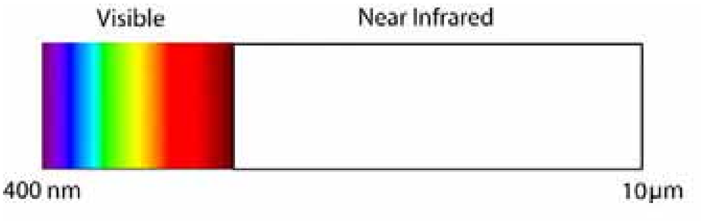


Figure 2.5 The range of wavelengths recorded by a hyperspectral sensor.

The components of the ImSpec sensor can be generalised into three parts. At the back of the sensor a Charged Coupled Device (CCD) array is placed for the registration of incoming data. In front of the CCD array is a crystal that divides the incoming light into different wavelengths. At the front is a scanning mirror which is necessary for aquiring the entire image. Without the mirror, only a single line would be aquired. In summary, the CCD array saves the incoming light over all the spectral components, divided up by the crystal, and the mirror scans the entire area to make an entire image.

2.2.2 The data

The data collected from the ImSpec sensor contains information about the spectral signature of the image. This spectral signature, as explained above, covers both the visible light as well as the NIR area. This allows the sensor to aquire spectral signatures that are not visible to the human eye.

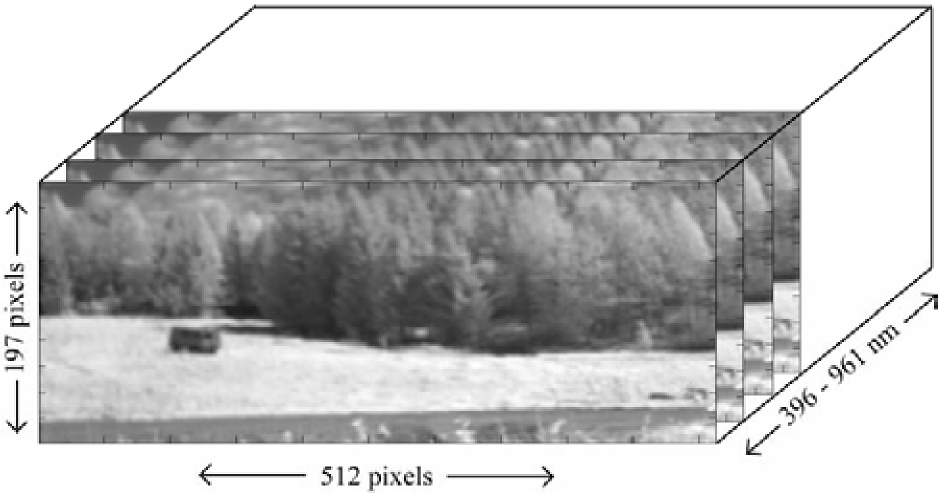


Figure 2.6 The hyperspectral data viewed as “slices”.

The data that is collected is arranged as a 3-D matrix. One “slice” is comprised of 512 x 197 pixels and can be seen as an image of the entire scene in a particular wavelength interval. The entire dataset spans 240 wavelength bands and can be compared to 240 images. Because the frequency information ranges from 396 to 961 nm, each band represents approximately 2,4 nm. If we abandon the notion of the data as slices and instead look at it as a stack of thin rods, we can see that each single “rod” contains the information of all the wavelengths recorded, for the corresponding pixel. The field of view of the sensor is 29x29º, so a pixel aquired with standard settings represents 0,056º.

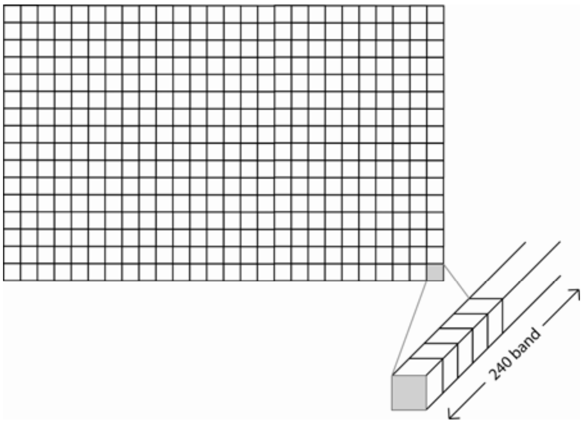


Figure 2.7 The hyperspectral data viewed as “rods”.

2.3 3-D laser

2.3.1 The sensor

To collect the 3-D data, the ILRIS-3D sensor from Optech was used. This sensor is a scanning 3-D laser radar and is composed of a laser, a detector and an advanced mechanical deflector. The laser can control the speed of the impulse as well as its frequency. The detector receives the returning laser shot and measures the time difference from that of exit to return as well as the intensity of the returning shot. The mechanical deflector controls the angle of the shot with great precision. It is crucial to the sensor, as it allows the sensor to fire entire grids of shots instead of just a single shot, creating an entire point cloud instead of a single reading.

To calculate the distance from the sensor to the target hit by the laser shot, the time of flight is used together with the known speed of the laser pulse. With precision control over the horizontal and vertical positions of the laser shot a 3-D point cloud is created. The position coordinates are given as X, Y, and Z with an additional variable I that represents the intensity. The intensity will vary in response to the target hit. A target with highly reflective properties will yield a high intensity value. Also, the angle of the target will affect the intensity value, with angles perpendicular to the laser shot returning high intensities.

Two general methods are used by the system, *first* and *last echo*. This means that the position recorded in the sensor is that of the first, or the last, thing the laser hits, exceeding a certain amplitude limit. To be able to penetrate the forest the equipment needed to use *last echo*.

2.3.2 The data

The data collected from the ILRIS sensor is called a point cloud and contains the positions and intensities of the surfaces hit by the laser. The positions are arranged in a 3-D coordinate system, given by either [X, Y, Z] or [Azimuth, Elevation, Range]. The intensity is given as a scalar ranging from 0 to 255, independent of the choice of coordinate system. In this thesis, the XYZ-system is used.

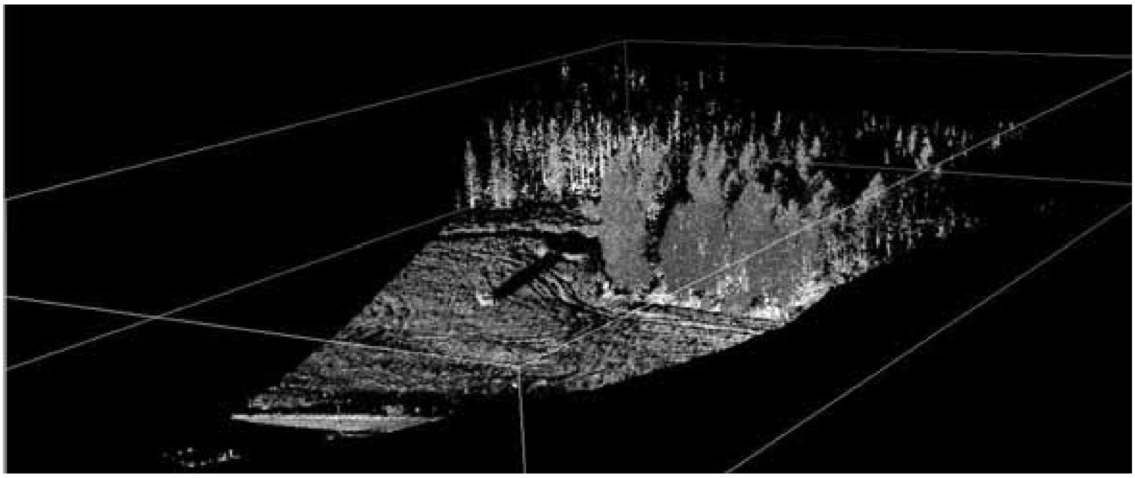


Figure 2.8 The point cloud data, slightly rotated.

Due to the nature of the collection of the data, the resulting point cloud is somewhat cone-shaped, as can be seen on the above image. Also, many areas will have "information shadows" where solid objects block the view of objects laying behind them and so there will not be any information in this area. The field of view of the sensor is about 40x40° and the divergence between the shots is 170 µrad. The mechanical deflector gives the laser shots a precision of approximately ±8 mm in the X-Y directions and ±7 mm in Z direction, all at 100 m. The sensor can fire 2000 shots/second. As an example it can be mentioned that one of our average scenes contained 2 178 882 shots, taking about 18 minutes to record. This requires a relatively static scene.

2.4 Registration

The registration of the two sensors used is a vital part of the process and allows the two datasets to be coherent. It ensures that points found in one dataset can be related to points in the other.

The relation between the 2-D hyperspectral image described by the variables (u,v) and the 3-D point cloud described by the variables (X,Y,Z) can be explained by the equation:

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

where

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{W} \begin{bmatrix} U \\ V \end{bmatrix}$$

The goal of his thesis was to calculate the value of P that would allow transition between the two datasets. To do this, both intrinsic and extrinsic parameters of the sensors had to be taken into account. To his help, he had data from both sensors containing the chequered boards that could be used to find common points.

The registration of the sensors was not a part of this thesis, but was done as a separate thesis by Kevin Chan at the University of Lund. Both projects were completed alongside each other, and used the same datasets. Further descriptions of his work can be found in [6].

3 Theory

This chapter will cover some general theoretical matter concerning methods and algorithms used further in the report. The use of them is not treated here, but in the respective parts of the report. This chapter can be read as part of the report or used as a separate reference.

The following illustrations will only show samples of 2-D data for practical reasons, but the theories holds true for higher dimensions as well. The data is only example data and is not related with the data collected for the thesis.

3.1 Definition of detection

A cornerstone in this thesis is the detection of anomalies in data. It is therefore useful to look at two different types of detection.

The most intuitive is *signature-based detection*, which means that we have knowledge of what we are looking for. In the context of spectral data, it would mean that we have a library of known spectral signatures or measured spectral signatures to use for comparison with the test signal.

The other definition of detection assumes no prior knowledge of the target or general scene. This is called an *anomaly detector*, and works on the principle of singling out an observed target spectrum if it deviates from the observed background spectra.

It is this latter form of detection that is used in this thesis.

3.2 Mahalanobis distance

Measuring the distance from a test point (x) to the mean (µ) of a collection of points(C) can easily be done by using the *Euclidian distance*.

$$d(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\| = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}.$$

However, this method does not into account the correlations of the dataset and does therefore not give a just measurement. To accomplish this, we use the *Mahalanobis distance*

$$D_M(x) = \sqrt{(x-\mu)^T\Sigma^{-1}(x-\mu)}.$$

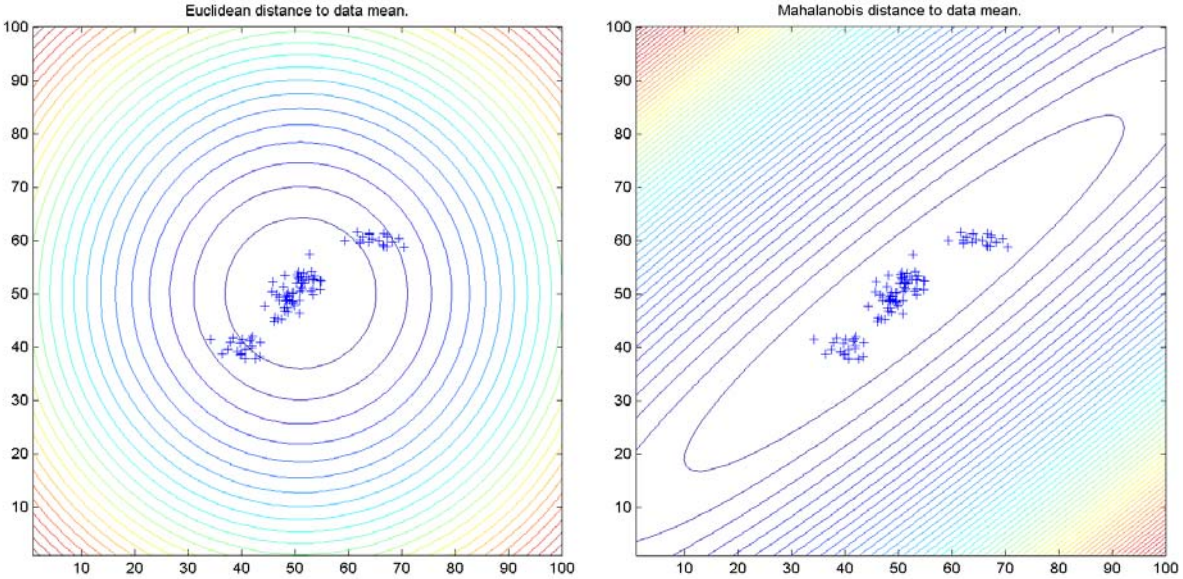
composed of a group of values with mean

$$\mu = (\mu_1, \mu_2, \mu_3, \dots, \mu_p)^T$$

and a covariance matrix Σ for a multivariate vector

$$x = (x_1, x_2, x_3, \dots, x_p)^T.$$

The difference between the two methods is clearly illustrated in the images below, where an isocurve represents a certain distance from the mean of the points.



.

Figure 3.1 Left: Euclidian isocurves representing the distance to the mean of the data. Right: Mahalanobian isocurves representing the distance to the mean of the data.

3.3 Clusters

As stated previously, the use of an anomaly detector requires no prior knowledge of the target. Therefore, the information must be derived from the scene by using samples from it. Given a collection of test data, the wish is to identify the points that seem to have little belonging to the model. The simplest way to do this is to measure the *Mahalanobis* distance, explained in 3.2, from the test point to the mean of the model. An isocurve is used as a decision level, so that all points lying outside are classified as anomalies. This simple method has flaws, as seen in the classification of the points below.

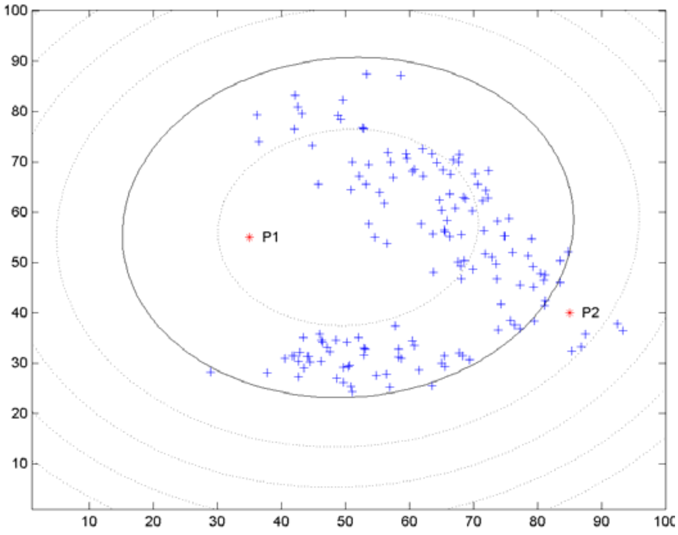


Figure 3.2 Points belonging to the cluster are left outside, and stray points are wrongly included.

The above method fails because the data appears to come from separate sources, and so a single mean is not sufficient. This is often the case with real world data because it comes from complex sources. To resolve this problem the data is first clustered, and then a model is created for each cluster. An anomaly is then defined only when it does not belong to any cluster.

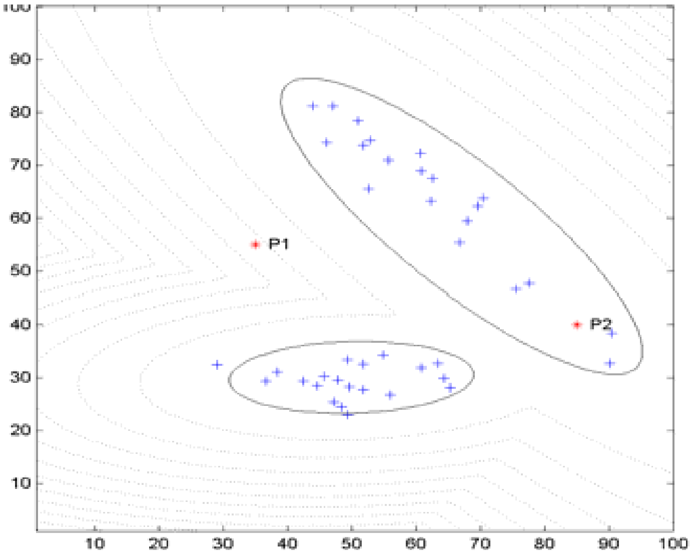


Figure 3.3 The points are now more naturally sorted, by clusters.

It can also be mentioned here that *hard clustering* is used in this thesis, meaning that a point can only belong to one cluster, as opposed to *soft* or *fussy clustering* where a point can belong to several clusters to various degrees.

3.4 Principal component analysis

Principal Component Analysis (PCA) is a statistical technique useful for finding patterns in data and data compression. Our main interest is in finding patterns. This technique relies on the basis of covariance in data and eigenvectors. It is assumed here that the reader has basic knowledge of statistics and linear algebra. If not, good sources can be found in [7] and [12].

Using PCA, it is possible to find similarities and dependences between different dimensions in data. An example can be the number of hours studied for an exam compared to the received mark.

The technique can be described by these steps:

1. Subtract the mean from each data sample.
2. Using the data from step 1, calculate its covariance matrix.
3. Calculate the eigenvectors and eigenvalues of the covariance matrix.
4. Of the eigenvectors, you can either choose the ones that represent the greatest covariances as main components, or use all the eigenvectors, and use these to form an ON basis.
5. Derive the new dataset using the new ON basis to describe the original dataset.

The steps described above may seem to have little to do with finding patterns, but does so by describing the data by its most important components. When the data is described in this manner, its most important features are “highlighted” by being the principal (first) components. To relate this to our topic, when analyzing 3D data, flat surfaces can be located. Such surfaces will vary greatly in two dimensions but not at all in a third, making the variance in this third dimension, or component, very small. Looking for data with such properties will point to flat areas. The image below shows a sample of 2-D data and the two components that describe it. It can easily be seen that the component P2 is much more important than P1, as this is where the data has the greatest variance. If we were to describe the data using only one component (data compression), P2 would give a better description than P1.

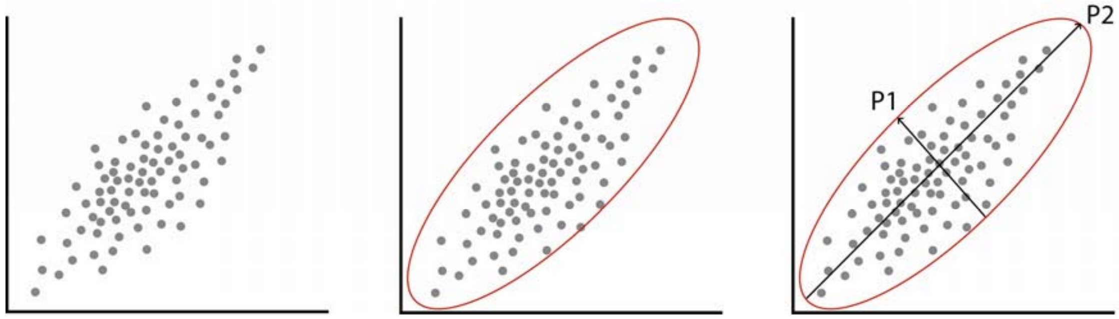


Figure 3.4 The two components describing the data collection.

3.5 Classification Expectation-Maximization

Now that the theories of distances and clusters have been explained, an algorithm used to perform all of this together can be analysed. The *Classification Expectation-Maximization* (CEM) algorithm was used to create the model of our image, based on samples taken from the image. There are a number of other algorithms that perform essentially the same task and return similar results using closely related methods. The reason why CEM was chosen was that it did the work in a reasonable amount of time and with satisfying results. More information about the remaining algorithms can be found in [1] and [2].

CEM takes samples from an image, called training vectors here, and classifies them into a number of clusters, giving each cluster an identity. The number of clusters is set in advance by the user. The process is done by these steps:

1. Initialize the given number of clusters randomly.
2. For each training vector, (re)compute the *Mahalanobis distance* to each class and classify into a cluster.
3. Recompute the classes using the method described in 3.4, where patterns indicate that data belong to the same class.
4. Repeat steps 2 and 3 until convergence.

In the first step, the initialization of the clusters means that the supposed clusters are being placed in the dataset at random positions. The fact that their placement is random results in the same training set yielding somewhat different results each time the algorithm is run. In the second and third step, the computation of the classes is done by analysing the eigenvectors to look for similarities between the points, and thereby determining if they belong to the same class. This is the same method used for PCA, and is explained in 5.4.

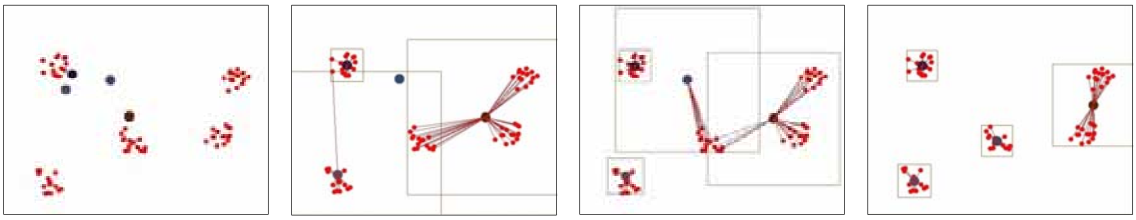


Figure 3.5 An example of CEM iterating with four clusters, their means represented by the darker points. The rectangles show to what class the points belong to in that particular step.

In summary, CEM is an algorithm that takes a collection of points and, given a number of wanted clusters, iterates until it has found that particular number of clusters amongst the points. It also gives a set identity to each cluster, so they can be distinguished from each other.

4 Software

The software used for all calculations and visualisations is *Matlab R2006a*. Many of the built-in toolboxes were used, in particular the *Image Processing Toolbox* and *GUI Toolbox*. Also, two custom toolboxes built at FOI were of great importance as they handled the hyper spectral and point cloud data we were dealing with. They are presented here below.

4.1 Hyper spectral data – HSI toolbox

This toolbox was created by Jörgen Ahlberg at FOI and a full technical report can be found in [2]. The purpose of the toolbox is to handle hyper spectral data and one of the many functions allows the detection of anomalies in such data. The goal, in our case, is to detect vehicles in a forest scenario.

4.1.1 Definition of anomaly

To classify a pixel as different compared to the surroundings, it must first be established what the image is composed of. In our case, various kinds of vegetation are the main components of the image. To make the comparison, a model of the image is built. Everything in the image that differs from this model can then be classified as an anomaly.

4.1.2 How the toolbox operates

As was specified in the previous chapter on hyper spectral data, this data is composed of vectors of several dimensions. Each vector represents one pixel in the image and each dimension of the vector represents a certain interval of wavelength. Because the data in our case contains 20 bands, each vector will be 20 dimensional.

To classify a pixel as an anomaly, its vector must be compared to the model. To do this comparison, the CEM algorithm mentioned in 5.4 is used to create a model of the image. Every vector in the image is then compared to this model. The *Mahalanobis distance* from the vector to the nearest cluster is then the indication of how common or unusual this particular vector is. A new image can be created from this information with anomaly values. This way a new image can be generated that alerts us to what areas may contain anomalies by being of a distinct colour, in this case a reddish hue.

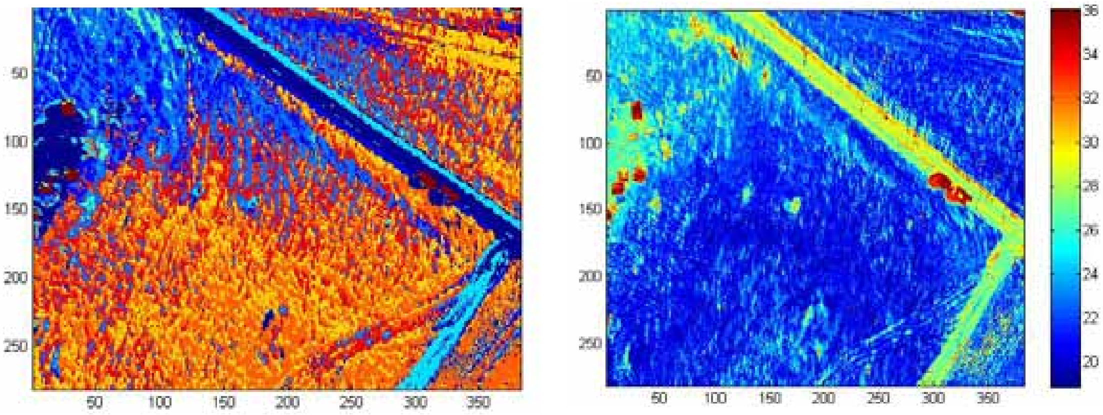


Figure 4.1 Left: Shows to what cluster every pixel belongs to, after the algorithm has been applied. Right The anomaly image where a redder colour indicates stronger deviation. Both images are generated from data taken from [13].

4.2 Point cloud data – 3-D Signal Processing toolbox

This toolbox was created by Tomas Chevalier at FOI and a related report can be found in [3] and [4]. The purpose of this toolbox is to facilitate the handling and modification of 3-D data sets.

4.2.1 How the toolbox operates

The main class is `PointCloud` which handles the 3-D points along with their intensities. The class has methods to transform the data as well as methods to gate, grid, plot and analyse the points.

The transformations possible are the basic tools used in data manipulation such as rotating, scaling, adding and subtracting point clouds. Explanations of these terms and how they are defined for 3-D data can be found in [11].

The method of gating allows us to select a section of the data for viewing and manipulating, without actually changing the original point cloud. To grid the data is to make an orthogonal projection of the 3-D data onto a 2-D plane, similar to taking a picture of the dataset. Plotting the data gives us information about its contents such as the distribution of points along a certain axis.

These functions are crucial to the combination of information from both the datasets. They allow us to make selections and modifications in the point cloud based on information gained from a previous or the current dataset.

5 Detection of anomalies

In this section, the mention of ”data” will refer to the hyperspectral data, unless otherwise mentioned.

5.1 Pre-processing

The data, as mentioned earlier, is comprised of 512x197 pixels and spans over 240 bands. Each band contains the information about a small number of wavelengths, covering approximately 2,4 nm in the electromagnetic spectrum. The small difference between each band in nm means that many of the bands are similar. To lighten the load for the program and to speed up the calculations, our data is pre-processed. The number of bands is reduced from 240 to 12. This is done by taking the mean of 20 consecutive bands and creating a new band. The pre-processing is done to all the available data and the new result is stored for future use.

This step is not necessary for the functionality of the program, but a simplification done to make the testing easier. Also, simply taking the mean of the bands is not the ideal compression of data, as some valuable information may be lost. A better way would be to group the data in such a way that bands containing wavelengths of greater importance (e.g. high reflection of metal) are given a wider range in the new dataset.

5.2 Detection

Once this pre-processing of the data is done, the HSI toolbox can be used to detect anomalies. A model is created by sampling every hundredth pixel of the image and using these together with the CEM algorithm. The sampling rate is, like many other variables, subject to the balance between fast data processing and accurate data. A model is created with 8 clusters into which all the vectors are clustered. Again, the number of clusters only reflects the current dataset and can be subject to change for other datasets. The method of choosing the ideal number of clusters is not a topic covered in this thesis. Finally, an iteration is done over the image where each pixel is compared to the model and it can then be decided if the pixel is an anomaly or not, and if so, to what extent. This creates the anomaly image.

5.3 Morphological operations

After the creation of the anomaly image, the HSI-toolbox is no longer used. The interest now lies in the isolation of interesting areas, the anomalies. The anomalies are found by a series of morphological operations performed on the anomaly image. These are operations that are done on the contents of the image, sometimes with a structured element (a mask), to alter the contents. Each operation will be briefly explained as it is encountered. For further explanations on this topic consult [10].

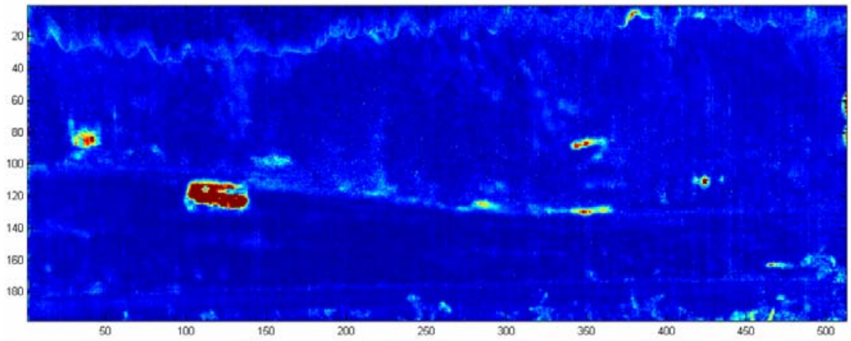


Figure 5.1 The floating point anomaly image, as created from the comparison.

5.3.1 Thresholding

As seen above, our anomaly image is in floating point. The first operation is creating a binary version of the anomaly image by using a threshold. This is simply a limit that decides if a pixel being analysed should be set to black or white. The value of the threshold is set automatically by a Matlab function (`graythresh`) based on the content of the image. It uses a method by Otsu, which chooses the threshold to minimize the intraclass variance of the black and white pixels [14].

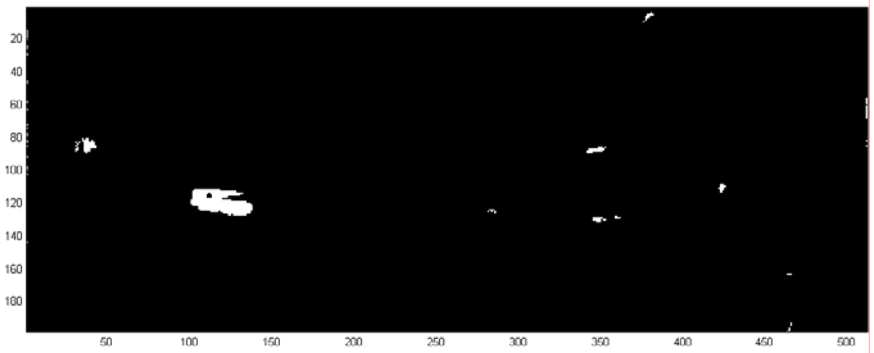


Figure 5.2 The binary anomaly image.

5.3.2 Opening and closing

The result is an image with an abundance of white areas in many different sizes. There is no interest for lonely single white pixels, but more so in larger continuous areas of white or dense gatherings of white pixels. This is because they are more likely to be a real object as opposed to noisy data. To eliminate the unwanted small points the methods of *opening* and *closing* are used. Opening and closing are the (different) combinations of two smaller morphological operations called *dilation* and *erosion*. As their names suggest, they dilate and erode objects.
They operate as follows:

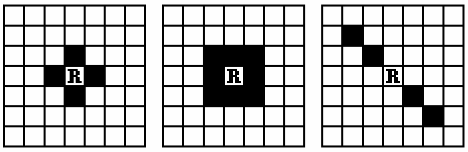


Figure 5.3 Three structured elements.

A structured element is, as seen in Fig.5.3, a grid containing zeroes (white) or ones (black). When iterated over an image, the reference point of the structured element is passed over every pixel of the object in the image. The eroded image of an object is the set of all reference points for which the structured element is completely contained in the object. The dilated image of an object is the set of all reference points for which the object and the structured element have at least one common point.

Opening is defined as an erosion, followed by a dilation. Closing is defined as a dilation, followed by an erosion. Opening and closing may seem like opposed methods, but applying them after each other to an image will not cause the original image to reappear, but a similar one. This fact is used to eliminate the unwanted points. Both opening and closing are applied once to the image. By first closing the objects in the image with a circular structured element larger than the unwanted points, the points will disappear. Next, the closed image is opened by the same element. The objects still present in the scene will almost return to their original look and smaller holes in large structures are filled. The fact that the resulting objects are not perfect copies of the original does not affect our work. The main interests are the location and approximate size of the objects, not their smaller details.

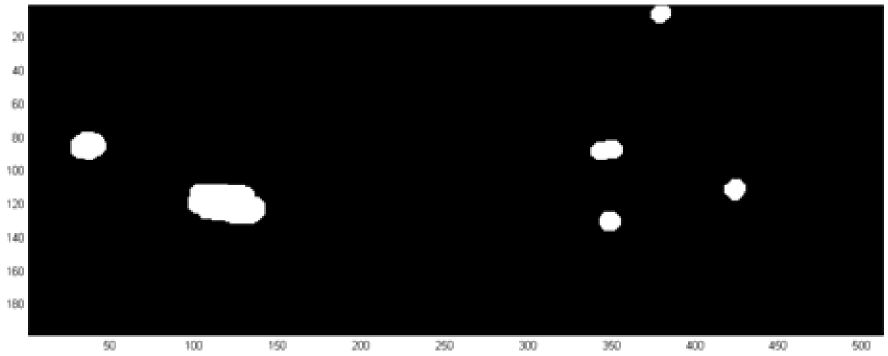


Figure 5.4 The anomaly image after removal of smaller points.

5.3.3 Labelling

Left in the image are the larger and more interesting areas i.e., the detected objects. Each white area is a separate anomaly that will be further analysed. It is therefore important that each anomaly is given an identity. This task is very simple for the human eye, but less so for a program. The answer is using morphological labelling. This process involves iterations of the image with a single point. For every white pixel the point encounters, it gives it a number based on its neighbours and the previous number given. When this is done a table is created, as large as the total number of objects. For each object found, the pixels belonging to it are saved in this table. It is now possible to access any anomaly separate from the others.

5.3.4 Bounding boxes

From the table of anomalies created in the labelling process, it is possible to extract the pixels that belong to each anomaly. These pixels take up a specific area in the image. It is this same area that will later be analysed in the 3D dataset. To make future selection of data easier, the smallest box possible surrounding each area, a bounding box, is calculated and the coordinates of these are saved. These coordinates are then used to extract the corresponding information from the 3D dataset. This can be done thanks to the registration of the cameras that gives us the corresponding points in the 3D coordinate system.

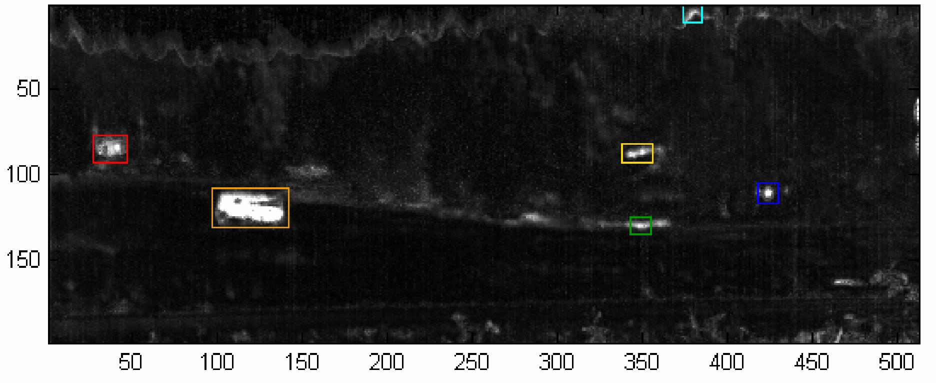


Figure 5.5 The original anomaly image in grey level with bounding boxes added.

6 Extraction of 3-D data

In this section, the mention of ”data” will refer to the point cloud data, unless otherwise mentioned.

As mentioned before, the purpose of this thesis is to show the strength of using interacting sensors. The previous chapters explained how information is collected from the hyper spectral sensor and then transformed in several steps. The new information retrieved from the hyper spectral data will now be used to better manage the 3-D data.

The data is, as mentioned before, comprised of a point cloud in 3-D space. Each point has an *X*, *Y* and *Z* value that represents its position in space and a value *I* that represents the intensity recorded. Using certain programs, like Matlab, it is possible to look at this data by zooming, translating and rotating it. This allows a user to make more sense of the data by looking at it from different views and thereby gaining better understanding of the contents. The large amount of data is at the same time positive and negative for the user. While a large amount of data gives much and detailed information, it can at the same time be overwhelming to search and important parts can be lost in the flood of information. The ideal solution would be to have detailed data on certain parts only. How is the data chosen that is to be removed?

This is when the combination of information from both sensors is used to make an intelligent choice.

The anomaly image is used to isolate interesting areas that should be explored further. As seen in the previous chapter, the anomaly image does not provide much information about the shape of the anomaly or its surroundings, but does indicate where to look more closely. The corresponding area in the data is extracted by cutting out a “tunnel” from the 3-D point cloud, possible thanks to the registration between the sensors. The amount of information to analyse is now greatly reduced, from the entire point cloud to a tunnel. However, the object of interest could be anywhere in this tunnel as the entire depth of the volume is taken.

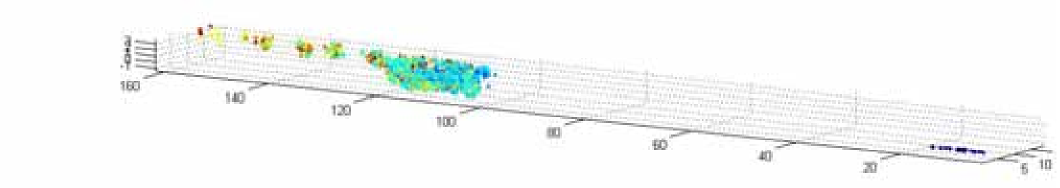


Figure 6.1 We do not know where in the point cloud tunnel the anomaly is located.

It is desirable to again reduce the amount of data by cutting the point cloud along the depth. Here, the information provided to us by the hyperspectral image cannot be of use. Other methods must be used to deduce what parts are objects of interest.

In order to automatically reduce the information depth-wise, certain features of the data must be analysed so that the object is isolated. What features are we looking for and what tools do we have at our hands?

Statistics of the data and the analysis of them are the most powerful tools available. As each dataset is different from the other, the method used must apply to various scenes.

The statistics that can be extracted from the data are the positions of the points and their intensities. Both these statistics are taken along the Y-axis, as it is the depth that should be reduced. Our hope is that the objects of interest will have some features that can be isolated in these statistics.

6.1 Special case: Vehicles

In this thesis, the search of anomalies has been limited to vehicles. Searching for vehicles means dealing with many flat surfaces and materials such as metal, rubber, and glass. Because of the way the 3D-data was collected (from one direction) there will only be a “profile” of the vehicle. The profile naturally depends on the angle at which the data was collected compared to the ground angle, but will normally incorporate one to three or four surfaces of the vehicle. This profile can be seen as a dense collection of points in space. Also, the materials on the vehicle have different reflective properties than the forest surrounding it, which means that the intensity of points that hit the vehicle will differ from the points of the surroundings.

6.1.1 Using the reflected intensity

Concluding from the previous statements, if there is a vehicle in the area, there should be a cluster of intensities quite different from the environment. An obvious way to find the vehicle is thus to look for such a cluster in a histogram of the data. The problem is that the reflective properties of materials are not the only factor to affect the intensity of a point. The angle of the surface will also highly influence the intensity value, with surfaces perpendicular to the laser shot giving a high intensity even if their reflective properties indicate they would not. So, when searching for vehicles in a forest area, the vehicles’ reflective properties will give high intensity values, but so will also trees, in particular tree trunks because of their angle. Another factor adding to the problem is that the vehicles, for obvious reasons, are often hidden at a depth into the forest. This reduces the amount of hits we can get on the vehicle from the laser scanner because of information shadows.

6.1.2 Proposed solution

The possible solution mentioned above would work only in a limited number of cases where the vehicles are well exposed. To locate the vehicles, even the more hidden ones, a different approach than intensities is used. The positions of the points are looked at and also their interrelations, more specifically their flat surfaces. This method will be more efficient as the tree trunks do not have the same large flat surfaces. The method is divided into several steps.

Removal of environment data

Firstly, our chances of locating the flat surface of the vehicle will improve by eliminating flat surfaces known to not be of interest, mainly the ground. The ground is defined as the lowest points in the point cloud, relative to the known direction of the sensor. Because terrain may vary in height, the lowest points will only be defined relative to small areas. Hence, the point cloud is divided into cells, spanning the entire height of the point cloud, and in each cell the ground is found and eliminated from the point cloud. In our case, all points less than 20cm from the estimated ground were eliminated, as were the points above 4m from the ground, assuming no vehicle would surpass that height.

Finding flat structures

The next step involves looking for flat areas. This property may be easy to recognise for a human observer, but much harder to define for a program. Again, the remaining point cloud is divided into cells. The size of the cells is relative to the nature of the object searched. Meter-wide cells were chosen in this case and a PCA is performed on the data in each cell. After the PCA, the result is analysed. Groups of points that are spread widely across two dimensions but have no variance at all in the third fulfil our requirements of “flatness”. It is therefore possible to grade how flat a surface is by looking at the PCA of it. A scalar score can then be given to all the points of this grouping, based on their flatness. The score is given by the proportions between the two larger eigenvalues and the smaller eigenvalue of the corresponding eigenvectors.

Histogram analysis

The final step in isolating the vehicle in the tunnel of points is done by looking at the new dataset created in the previous step. Vehicles with large flat surfaces will have a higher score than natural surroundings like trees who are linear rather than planar. By looking at a histogram of intensities of this new dataset, along the Y axis, the location of the vehicle is marked by a significant peak in the histogram. It is then easy to cut out the corresponding area directly surrounding this peak. In our case an area 5m before and after the peak was selected to assure that the entire vehicle would be included.

The first part of the thesis is now completed, the vehicle has been isolated in the second dataset, by using information from the first dataset.

7 Visualising the result – Graphical User Interface

Now that the object has been isolated in the point cloud data, the result should be presented in a manner that gives the most information to the user. Using a Graphical User Interface (GUI) to present the results means letting the user interact with the data by means of buttons and menus instead of written commands. The GUI in this thesis was created in Matlab, which is perhaps not the ideal program for aesthetically pleasing results, but good for the underlying calculations. The GUI is composed of two individual windows that present the results for the hyper spectral analysis and the point cloud analysis respectively.

In Matlab, there are two ways of generating GUIs. One is by using *GUIDE*, Matlab's own development environment for GUIs, which is mainly a "drag-and-drop" environment for easy creation. The second method, and the one chosen for this project, is manually coding the GUI. This is more time consuming but gives a lot more control over the features of the GUI. The hyperspectral data, and the following point cloud data, are both presented in standard Matlab windows. These can be altered to include menus, buttons, images, grids, texts, and so on.

7.1 Hyperspectral

7.1.1 The look

The hyperspectral data is presented by two images. The first contains a single band from the hyperspectral data which shows the view as seen by the "naked eye". It may seem unnecessary to include, but it is always good to relate back to the origins of the data. The second image is the anomaly image which shows the areas of interest as whiter then the surroundings. The anomaly image is the one created directly by comparison of the vectors to the model and has not been modified. Both images are presented in greyscale. This is mainly to minimise confusion with the bounding boxes added to the images. The boxes, which appear on both images, are the bounding boxes that surround the found anomalies. They are there to alert the user of the interesting areas and their different colours are also an identity marker, each anomaly having its own colour. The same colour is also used in the point cloud GUI.

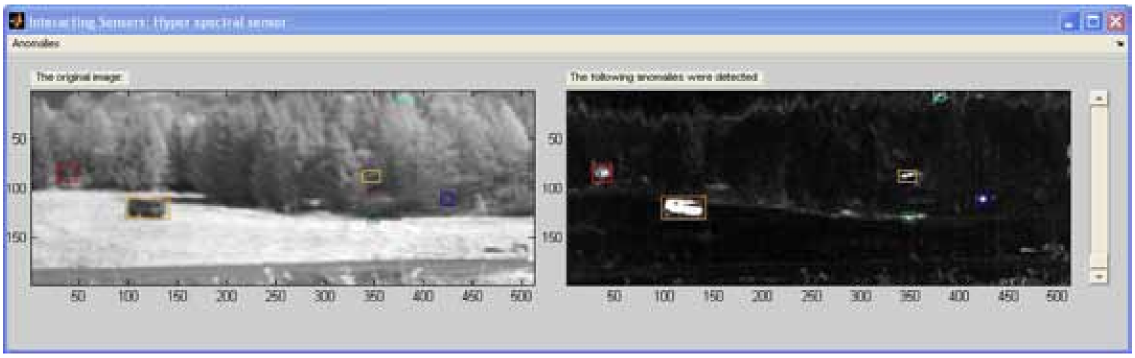


Figure 7.1 The hyperspectral GUI. Left image shows one wavelength band with rectangles marking anomalies, detected through the anomaly image to the right.

7.1.2 Functions

When presented the anomaly image, the user may not be satisfied with the anomalies chosen by the program. The user then has the option of manually adding further anomalies. This is done by choosing *Anomalies > Add anomaly* in the window menu. Using the pointer, the user is asked to click first on the upper left and then the lower right points enclosing the area they wish to select as anomaly. The result of this action is that both windows of the GUI's are updated with the latest anomaly.

Another function that the user can control is the contrast level of both images. By scrolling the bar to the right of the anomaly image the contrast is adjusted in both images. This can be a desirable function when manually looking for anomalies.

7.2 Point cloud

7.2.1 The look

In the second window the point cloud dataset is presented, cut down to the volumes that contain the anomalies. A projection of each anomaly point cloud is used as a preview. The colours in the projection image are the original intensities of the points, as aquired by the sensor. If some projections may seem distorted, this is because the anomaly point cloud does not contain many points and Matlab does an interpolation between them for the projection. The identity colour used in the hyper spectral GUI is used in the text of each anomaly to associate the two different datasets. Because of restrictions in screen space, and for clarity, the full point cloud content of each anomaly is presented only if the user presses on the *Analyse* button belonging to the anomaly.

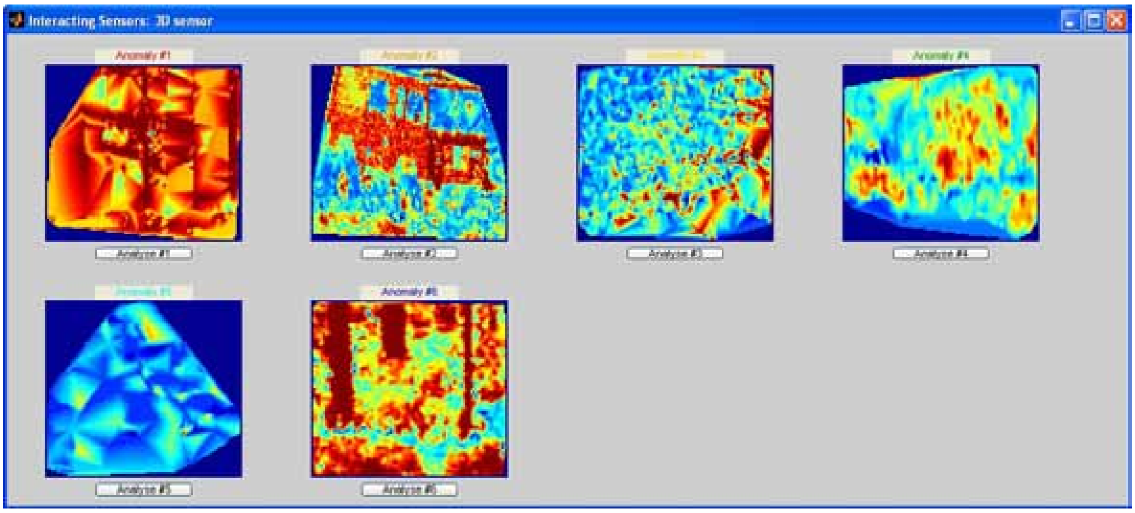


Figure 7.2 The point cloud GUI. Each figure is an anomaly point cloud projected onto 2-D.

7.2.2 Functions

Pressing the *Analyse* button in the main point cloud window generates a separate standard Matlab window in which that particular anomaly is presented. Added to this window are two extra menu choices, *View* and *Extract*. The point cloud is shown in a grid that can be rotated, zoomed in/out and moved around. The user has the option to view the data in different modes, selected in the *View* menu, presented in this list:

Colour:
Shows the anomaly point cloud with intensities as recorded from the sensor.
Greyscale:
Shows the anomaly point cloud with intensities in a greyscale
Flatness:
Shows the point cloud generated when calculating the flatness, with the flatness scores as intensities.
Entire tunnel:
Shows the entire point cloud tunnel, uncut in depth, with intensities as recorded from the sensor.
Textured:
Shows the anomaly point cloud with the intensities replaced by values taken from the anomaly image.

Images of these modes can be found in the Appendix. The user can also save each one of these viewing modes by selecting *Extract > Save PointCloud* from the menu in the separate window.

8 Conclusion

This final chapter is an analysis of the results.

8.1 Result

The main goal of this thesis was to show that using interacting sensors and combining their data would yield informative results. An aspect of the goal was to present this result in a straightforward manner that did not demand any specific knowledge from the user. The thesis has resulted in a program that takes the data from a hyper spectral sensor and a scanning 3D laser. By analysing and combining the information retrieved from respective data sets, anomalies could be isolated and presented, that would otherwise have had to be found manually. The user can then look closer at this result by using the GUI and may also add more anomalies.

8.2 Analysis of result

8.2.1 Advantages

- The use of this program does not require any specific knowledge from the user, apart from a basic understanding of the concepts involved. This means that it can be used by individuals that do not know the algorithms or techniques involved but still have use of the results.

- The presentation is simple and straightforward, but there are at the same time easy ways to further analyse the results. The individual windows that display anomalies are also simple in use and at the same time rich in viewing modes, giving much information to the user.

- Even though most of the process is automatic, the user still has the possibility to add areas of interest for further inspection. This means that areas of interest can be found based on the knowledge of the user, which can not always be transmitted to the program.

8.2.2 Drawbacks

- Just as it was an advantage that the program does not require any input other than the data to compute the results, it is also a drawback for the advanced user. The possibilities to make adjustments in the settings are limited to those who have access to the code.

- The presentation is perhaps too simple. It is quite limited and could be made more intuitive. There is always room for improvement concerning the features that could be included in the program.

- The program uses pre-processed data to perform the calculations, but is still not very fast. It takes between one and four minutes for the program to process all the calculations and graphics. This time would be even longer had it used the raw data from the sensors. This is clearly a disadvantage if the program were to be used in occasions that demand fast response.

8.3 Further development

The results of this thesis can be further developed by correcting flaws discussed in the Drawback section and also by new implementations:

- Adding the possibility to adjust settings before running the program, such as the number of clusters, thresholds and so on. This requires a standard setting, for the users who do not wish to manually adjust settings.

- Investigating the possibility to create an application that automatically chooses the number of clusters for the scene presented.

- If using the same sensors as in this thesis, one main development would include other manners of isolating the anomalies. This detection could include new ways to isolate the same targets, which does not rely on flatness. Another possibility would be the detection of other possible targets, such as humans or man-built structures (weapons, tents and so on).

- Development of the program to allow for the use of other types of sensors. This naturally means handling completely different types of data and also using this data in an entirely different manner.

- Better compression of the hyper spectral data. As mentioned before, the data was compressed from 240 bands to 12 using only the mean of 20 consecutive bands. A better way to compress data would be one that gives greater importance to bands that contain wavelengths with strong signatures from the materials searched for, and less importance to the other bands.

- Including the possibility to apply *signature-based detection* to the data, making it possible to search for a particular feature.

- Creating a better graphical solution. As previously mentioned, the use of Matlab is not ideal in a graphical point of view. Perhaps the use of another program could allow the presentation of the result to be better.

- Including a learning system in the program. The program could then draw knowledge from when the user identifies that the program has discovered a false anomaly and likewise when the user manually adds an anomaly missed by the program.

Acknowledgement

This thesis will finalise a M.Sc. in Media Technology at the Institute of Technology, Linköping University. The thesis was done at the Swedish Defence Research Agency (FOI), division of Sensor Technology. Supervisors at FOI have been Jörgen Ahlberg and Tomas Chevalier.

A special notice should go out to Kevin Chan, whose work made this thesis possible, and to other thesis students in the department, Linda Nordin and Erik Thorin, for their support.

I would like to thank my examiner Stefan Gustavson at Linköping University and my supervisors at FOI for all their help and advice. Their support has made this work possible and I am very happy to have been inspired by their great ideas and views.

Finally, I would like to thank Mats Janné for all his support throughout my entire thesis and for taking much of his time to review and comment on my work and report, and to my opponent David Hertz for reviewing my work.

References

[1] Jörgen Ahlberg, Ingmar Renhorn, *Multi- and Hyperspectral Target and Anomaly Detection*, Scientific report FOI-R--1526--SE, Swedish Defence Research Agency, 2004

[2] Jörgen Ahlberg, *A Matlab Toolbox for Analysis of Mulit/Hyperspectral Imagery*, Technical report FOI-R--1962--SE, Swedish Defence Research Agency, 2006

[3] T. Chevalier, P. Andersson, C. Grönwall, F. Gustafsson, J. Landgård, H. Larsson, D. Letalick, A. Linderhed, O. Steinvall, G. Tolt, *Årsrapport 3D-laser 2005*, User report FOI-R--1807--SE, Swedish Defence Research Agency, 2005

[4] O. Steinvall, L. Klasén, T. Chevalier, P. Andersson, H. Larsson, M. Elmqvist, M. Henriksson, *Grindad Avbildning - fördjupad studie*, Scientific Report, FOI-R--0991--SE, Swedish Defence Reseach Agency, 2003.

[5] H. Larsson, P. Nilsson, T. Chevalier, R. Persson, *Mätrapport Samverkande sensorer Ströplahult Norra maj 2006*, FOI-D--0621--SE, Swedish Defence Research Agency, 2006

[6] Kevin Chan, *Registration of 3-D laser radar data with hyperspectral imagery for target detection*, Technical Report FOI-R--2101--SE, Swedish Defence Research Agency, 2006

[7] Lindsay I. Smith, *A tutorial on principal component analysis*, Cornell University, USA, 2002

[8] Duane Hanselman & Bruce Littlefield, *Mastering Matlab 7*, Prentice Hall, 2005

[9] Robert A Schowengredt, *Remote Sensing*, Academic Press, 1997

[10] Rafael Gonzalez, Richard Woods, *Digital Image Processing, Second Edition*, Prentice Hall, 2003

[11] Alan H Watt, *3D Computer Graphics, Third Edition*, Addison-Wesley, 1999

[12] Gunnar Sparr, *Linjär Algebra, Andra upplagan*, Studentlitteratur, 1998

[13] Oskar Brattberg, *Analys av multispektrala spaningsdata för måligenkänning*, Technical report, FOI - Swedish Defence Research Agency, to be published.

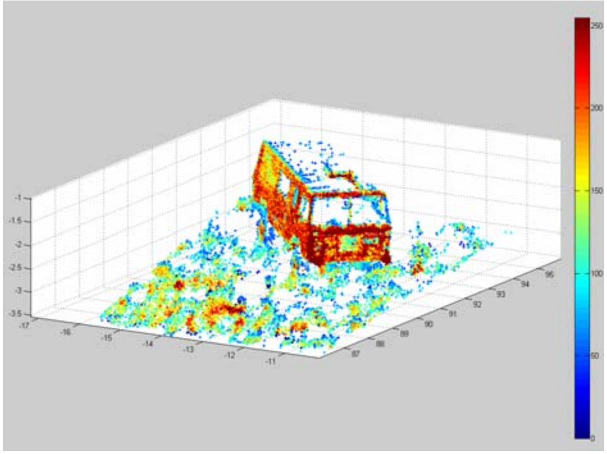
[14] N Otsu, *A Threshold Selection Method from Gray-Level Histograms, IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1, 1979, pp. 62-66.

Matlab support homepage
http://www.mathworks.com

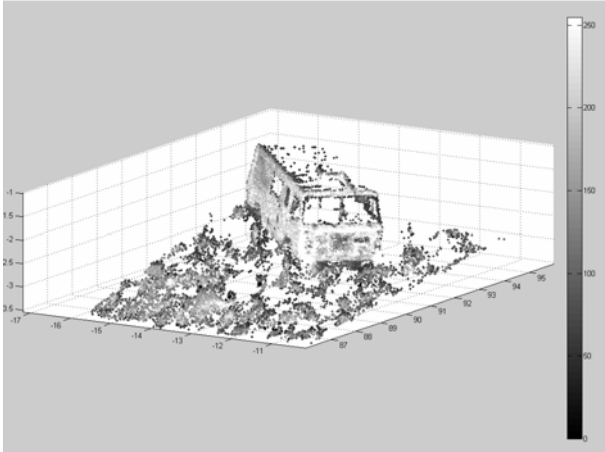
Online encyclopaedia
http://www.wikipedia.org/

Appendix - Images

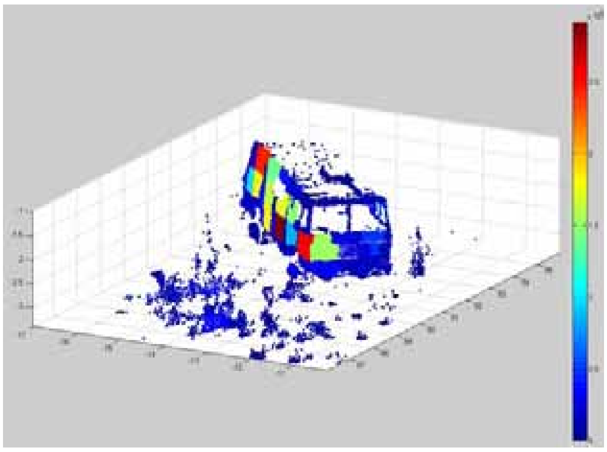
The five different viewing modes of an individual anomaly:



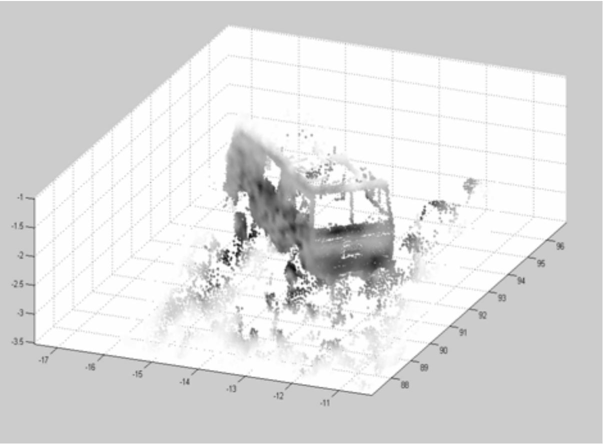
Intensity from laser (Colour)



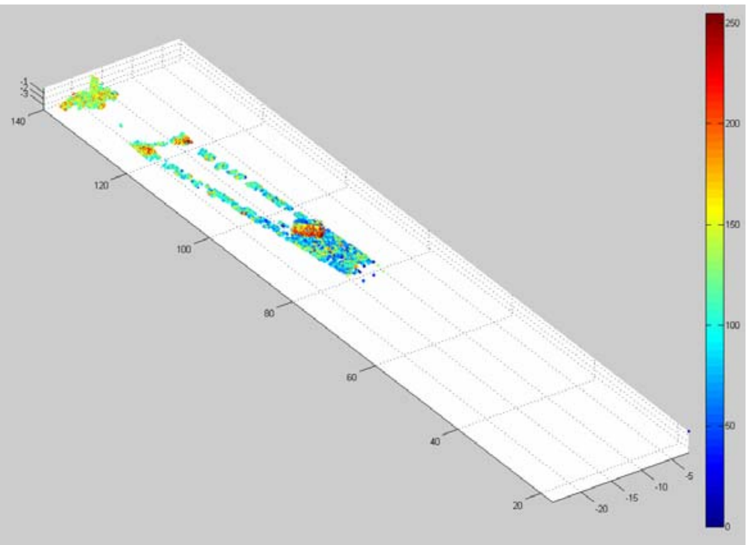
Intensity from laser (Grayscale)



Flatness score



Textured with anomaly values



Entire tunnel with intensity from laser