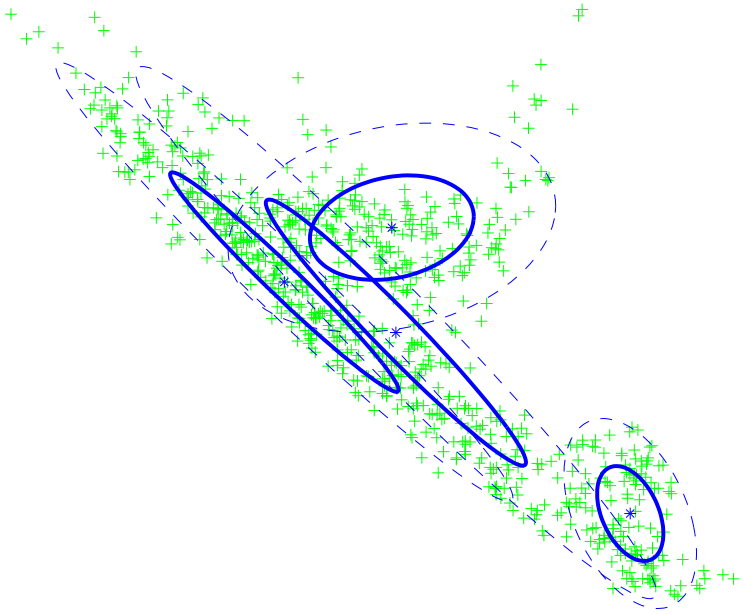
FOI-R--1526--SE
December 2004
1650-1942

Scientific report

Jörgen Ahlberg
Ingmar Renhorn

Multi- and Hyperspectral Target and Anomaly Detection



Division of Sensor Technology
Box 1165
SE-581 11 LINKÖPING
Sweden

Swedish Defence Research Agency
Division of Sensor Technology
Box 1165
SE-581 11 LINKÖPING
Sweden

FOI-R--1526--SE
December 2004
1650-1942

Scientific report

Jörgen Ahlberg
Ingmar Renhorn

Multi- and Hyperspectral Target and Anomaly Detection

Issuing organization	Report number, ISRN	Report type
Swedish Defence Research Agency Division of Sensor Technology Box 1165 SE-581 11 LINKÖPING Sweden	FOI-R--1526--SE	Scientific report
	Research area code C^4ISR	
	Month year December 2004	Project no. E3059
	Customers code Commissioned Research	
	Sub area code Reconnaissance and Surveillance	

Author/s (editor/s)	Project manager
Jörgen Ahlberg Ingmar Renhorn	Tomas Chevalier
	Approved by Lena Klasén
	Sponsoring agency Swedish Armed Forces
	Scientifically and technically responsible

Report title

Multi- and Hyperspectral Target and Anomaly Detection

Abstract

This report treats detection of targets and anomalies in multi- and hyperspectral imagery. An anomaly is in this context something that does not fit a (spectral) model of the background, for example man-made objects in a natural environment. *Anomaly detection* has the advantage that it does not require a priori knowledge on what is searched for, and/it has the disadvantage that also subjectively unintresing objects are detected. In contrast to anomaly detection an algorithm for or *(signature-based) target detection* searches for specific targets with known spectral signatures, for example from a data base. This report describes mathematical models for targets and backgrounds and how they are used for detection algorithms.

Keywords

multispectral, hyperspectral, anomaly detection, target detection, image analysis, remote sensing

Further bibliographic information	Language
	English
ISSN 1650-1942	Pages 41
Distribution By sendlist	Price Acc. to pricelist Security classification Unclassified

Utgivare	Rapportnummer, ISRN	Klassificering
Totalförsvarets forskningsinstitut Avdelningen för Sensorteknik Box 1165 SE-581 11 LINKÖPING Sweden	FOI-R--1526--SE	Vetenskaplig rapport
	Forskningsområde: Spaning och ledning	
	Månad, år: December 2004	Projektnummer: E3059
	Verksamhetsgren: Uppdragsfinansierad verksamhet	
	Delområde: Spaningssensorer	

Författare/redaktör	Projektledare
Jörgen Ahlberg Ingmar Renhorn	Tomas Chevalier
	Godkänd av: Lena Klasén
	Uppdragsgivare/kundbeteckning: Försvarsmakten
	Tekniskt och/eller vetenskapligt ansvarig

Rapportens titel

Multi- och hyperspektral mål- och anomalidetektion

Sammanfattning

Denna rapport behandlar detektion av mål och anomalier i multi- och hyperspektrala bilder. En anomali är i detta sammanhang någonting som inte passar in i en (spektral) modell av bakgrunden, till exempel människotillverkade föremål i en naturlig miljö. *Anomalidetektion* har den fördelen att den inte förutsätter kunskap om exakt vad man letar efter, men har nackdelen att man även detekterar subjektivt ointressanta saker. I kontrast till anomalidetektion söker en algoritm för *(signatur-baserad) måldetektion* efter specifika mål med kända spektrala signaturer, till exempel från en databas. Rapporten beskriver ett antal matematiska modeller för mål och bakgrunder samt hur de används till detektionsalgoritmer.

Nyckelord

multispektral, hyperspektral, anomalidetektion, måldetektion, bildanalys, fjärranalys

Övriga bibliografiska uppgifter	Språk
	Engelska

ISSN	Antal sidor
1650-1942	41

Distribution	
Enligt missiv	Pris: Enligt prislista Sekretess: Öppen

Contents

1. Introduction

Multi- and hyperspectral image exploitation is a growing field not only in remote sensing within the civilian community but also in defence applications such as reconnaissance and surveillance. Multi- and hyperspectral electro-optical sensors are sensors (cameras) that sample the incoming light at several (multispectral sensors) or many (hyperspectral sensors) different wavelength bands. Compared to a consumer camera that, typically, uses three wavelength bands, corresponding to the red, green and blue colours, hyperspectral sensors sample the scene in a large number of wavelength (or spectral) bands, often several hundred. Moreover, these spectral bands can be beyond the visible range, i.e, in the infrared domain. Each pixel thus forms a (spectral) vector of measurements in the different bands. This vector, the observed spectral signature, contains information on the material(s) present in the scene, and can be exploited for detection, classification and recognition. This report treats methods for detecting anomalies and targets in hyperspectral images, using the spectral information in each pixel.

1.1 Spectral detection

If an observed target spectrum deviates from observed background spectra, this deviation can serve as a measure of anomaly. An *anomaly detector* is thus a detector that detects pixels that "stick out" from the background, without any a priori knowledge about target or background.

In order to be able to perform a unique *classification* or *signature-based detection*, the spectral properties of the scene elements or targets must be known from laboratory measurements or from in situ measurements. Observed spectra are analysed both from a statistical point of view and compared with laboratory data, that is, a priori knowledge is used in order to enhance detection probabilities and classification capabilities. The impact of illumination, weather and atmospheric transmission must also be estimated in order to relate observed spectra to laboratory measurements correctly. Reference panels and other reference sources in the scene can help in calibrating the sensor systems and also in estimating the correlation in spectral scene properties from one trial to another.

1.2 Hyperspectral sensors

The sensor (and the scene) can be characterized with respect to spatial, spectral, radiometric and temporal resolution (and properties).

The spatial resolution and the distance from the sensor to the target determines whether a target can be spatially resolved or not. A spatially *resolved target* covers at least one pixel completely, which means that the target pixel(s) will be *pure*, in contrast the *mixed* pixel of a *sub-pixel target*. Generally, sub-pixel targets are very difficult to detect and must deviate substantially from the surroundings in order to be distinguishable. Spatial resolution will therefore be an important performance parameter.

The spectral resolution determines the number of spectral bands, while the radiometric resolution determines the number of bits per sample and is limited by the

signal-to-noise ratio. The temporal resolution determines how often a new pixel can be produced by the sensor.

In practical sensor design, trade-offs have to be made between spatial, spectral and temporal resolution. An important issue is therefore to try to establish optimal trade-offs with respect to scenarios and applications. In the final end, tactical sensors must be both inexpensive and perform well with respect to spatial, spectral and temporal information. Hyperspectral information can strongly support such an optimisation. New technologies might also open up the possibility to make these sensors adaptive to changes in the spectral content of the scene and the application requirement.

1.3 Outline of this report

This report describes algorithms and methods for target and anomaly detection in multi- and hyperspectral images. Spatial domain detectors are not considered at all, even if much of the underlying theory is identical. For such detectors, see the survey in [5].

No detection results are given in this report. Instead, experiments and experimental results will be reported in a separate document.

The outline of the report is as follows. In Chapter 2 some mathematical preliminaries are given and different detection principles are defined. Chapters 3–5 describe methods for spectral modelling and how they are used for detection. Chapter 6 describes the spatial modelling, basically defining how to select signature vectors for spectral modelling. Chapter 7 contains a summary and a final discussion.

The detectors treated in Chapters 3–5 are organized according to (primarily) what model for background clutter they use and (secondarily) what a priori knowledge about targets they require. The summary in Chapter 7 is organized primarily according to the task (anomaly or target detection) and secondarily according to a priori knowledge. The purpose is that the designer of a detector should be able to, given a task and some knowledge about the targets, look up the specific detector that suits his needs.

Table 1.1 summarizes the mathematical notation used in this report.

Table 1.1: Notation

Symbol	Meaning
a, x, ϕ	Scalars
$\mathbf{a}$, $\mathbf{x}$, $\boldsymbol{\phi}$	Vectors
$\mathbf{A}$, $\mathbf{X}$, $\boldsymbol{\Phi}$	Matrices
a, x	Random variables
$\mathbf{a}, \mathbf{x}$	Random vectors
$\mathcal{A}, \mathcal{X}$	Models/classes
$\sim$	"is distributed as"
$\hat{x}, \hat{\mathbf{x}}, \hat{\mathbf{X}}$	Approximation
$\tilde{\mathbf{x}}, \tilde{\mathbf{X}}, \tilde{\mathbf{x}}$	Whitening
Special symbols	
$\mathbf{0}$	The zero vector
$\mathbf{I}$	The identity matrix
$\boldsymbol{\Gamma}$	Covariance matrix
σ^2	Variance
$\boldsymbol{\mu}$	Mean or prototype vector
$\boldsymbol{\Phi}$	Basis matrix
$\mathcal{B}$, $\mathcal{T}$	Background and target models
$\Pr(A), P_A$	Probability of the event A
$p(x)$	Probability density function

2. Spectral Signal Processing and Modelling

This chapter treats some of the preliminaries needed for the rest of the report. The chapter can be skipped by the reader who is already familiar with signal processing.

The topics covered here, though somewhat superficially, are basic signal detection theory, methods for reducing the dimensionality of vector data, and different types of spectral modelling and detection.

2.1 Signal detection theory

Assume we have a scalar-valued observed quantity. On the basis of this observation, a decision of two hypotheses, H_0 and H_1, shall be made. For example, hypothesis H_0 could be "no target is present" and hypothesis H_1 should then be "a target *is* present". Since noise and other unknown factors influence the observation, it is regarded as a random variable x. x can then be characterized by its *probability density function* (pdf) $p(x)$ under H_0 and H_1 respectively, so that

$$\int_a^b p_k(x)dx = \Pr(a < \mathsf{x} \leq b \,|\, \mathsf{H}_k), \;\; k = 0, 1, \tag{2.1}$$

and our hypotheses are

$$\begin{aligned} \mathsf{H}_0 &: \; \mathsf{x} \sim p_0(x) \\ \mathsf{H}_1 &: \; \mathsf{x} \sim p_1(x). \end{aligned} \tag{2.2}$$

A threshold t can be defined, so that the hypothesis H_0 is accepted if $\mathsf{x} \leq t$, and H_1 is accepted if $\mathsf{x} > t$. The probability $Q_{10}(t)$ of choosing hypothesis H_1 when H_0 is actually true ("false alarm") is then

$$Q_{10}(t) = \int_t^\infty p_0(x)dx, \tag{2.3}$$

and the probability $Q_{01}(t)$ of choosing hypothesis H_0 when H_1 is true ("miss") is

$$Q_{01}(t) = \int_{-\infty}^t p_1(x)dx. \tag{2.4}$$

The value of t depends on the pdf:s and on how much these two different mistakes will cost. Assuming two values C_{10} and C_{01} to define the cost of the respective mistakes, the *risk* associated with hypothesis H_0 can be defined as $C_{10}Q_{10}(t)$ (and analogously for H_1). The *average risk* $C(t)$ is then dependent on the risks and the a priori probabilities of H_0 and H_1 (P_0 and $1 - P_0$ respectively). $C(t)$ is calculated as

$$\begin{aligned} C(t) &= P_0C_{10}Q_{10}(t) + (1 - P_0)C_{01}Q_{01}(t) \\ &= P_0C_{10}\int_t^\infty p_0(x)dx + (1 - P_0)C_{01}\int_{-\infty}^t p_1(x)dx \end{aligned} \tag{2.5}$$

Naturally, we want to choose the t that minimizes the risk. To find this level, (2.5) is differentiated with respect to t and set to zero. The result is

$$\frac{p_1(t)}{p_0(t)} = \frac{P_0 C_{10}}{(1-P_0)C_{01}}, \tag{2.6}$$

from which t can be calculated when assuming specific probability densities $p_k(x)$. The ratio

$$\Lambda(x) = \frac{p_1(x)}{p_0(x)} \tag{2.7}$$

is called the *likelihood ratio*, and $\Lambda^\star = \Lambda(t)$ is called the *decision level*. The decision regions R_0 and R_1 consists of the points where $\Lambda(x) < \Lambda^\star$ and $\Lambda(x) > \Lambda^\star$ respectively. This strategy is known as the *Bayes solution*, and the minimum value of $C(t)$ is called the *Bayes risk*.

In target detection, the probabilities

$$P_{\mathsf{D}}(t) = Q_{11}(t) = \int_t^\infty p_1(x)dx \qquad \text{(Probability of detection)} \tag{2.8}$$

$$P_{\mathsf{FA}}(t) = Q_{10}(t) = \int_t^\infty p_0(x)dx \qquad \text{(Probability of false alarm)} \tag{2.9}$$

are used as performance measure, as will be discussed in Section 2.5.

All the above can be generalized to multivariate distributions. The only changes needed concerns the dimensionality of x and $p_k(x)$, and consequently the integrals in (2.3)–(2.5). Alternatively, the input is tranformed to a scalar value, as is exemplified in Section 3.1.4.

In general, the probability distributions $p_k(x)$ are not known, and thus a model is assumed, as will be described in Chapter 3.

2.2 Dimensionality reduction and whitening

Given a set of N-dimensional data samples $\{\mathbf{x}_k\}_{k=1}^K$ to analyse, a problem is often the massive amount of data, especially if the samples $\mathbf{x}_k$ are high-dimensional. One solution is to reduce the dimensionality of the data, provided that this can be done without losing important information. A simple and popular way to reduce dimensionality and also to extract intresting features is *principal component analysis* (PCA). The procedure is as follows.

- Compute the mean and covariance of the training data:

$$\boldsymbol{\mu} = \frac{1}{K}\sum_{k=1}^K \mathbf{x}_k \tag{2.10a}$$

$$\boldsymbol{\Gamma} = \frac{1}{K-1}\sum_{k=1}^K (\mathbf{x}_k - \boldsymbol{\mu})(\mathbf{x}_k - \boldsymbol{\mu})^T \tag{2.10b}$$

  Perform a *singular value decomposition* (SVD) or an *eigenvalue decomposition* of the covariance matrix, i.e., find the matrices $\boldsymbol{\Gamma} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^T$, where the columns of $\mathbf{U}$ contain the subspace basis. $\boldsymbol{\Sigma}$ is a diagonal matrix where the elements σ_i^2 of the diagonal indicate the energy distribution of the training samples along the directions in the corresponding columns of $\mathbf{U}$. We assume in the following that the columns of $\mathbf{U}$ (and $\boldsymbol{\Sigma}$) are ordered so that $\sigma_1^2 > \sigma_2^2 > \ldots$.

- Alternatively, if we assume that the subspace includes the origin (the zero vector), we can perform an SVD directly on the data matrix $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_K]$. The mean vector $\boldsymbol{\mu}$ can then be omitted from equations (2.12)–(2.14) below.

- The M-dimensional subspace basis is spanned by $\boldsymbol{\Phi} = [\mathbf{u}_1 \cdots \mathbf{u}_M]$, where the vectors $\mathbf{u}_i$ are called the *principal components*. Typically, M is chosen so that a certain amount q, say 99 percent, of the signal energy is preserved, i.e.,

$$\frac{\sum_{i=1}^{M} \sigma_i^2}{\sum_{i=1}^{N} \sigma_i^2} \geq q. \tag{2.11}$$

- To *project* a sample $\mathbf{x}$ on the subspace, i.e., to reduce the dimensionality, compute

$$\mathbf{y} = \boldsymbol{\Phi}^T(\mathbf{x} - \boldsymbol{\mu}) \tag{2.12}$$

- To *whiten* data, i.e., to transform it so that its components are uncorrelated and have equal variance, compute

$$\tilde{\mathbf{x}} = \boldsymbol{\Gamma}^{-\frac{1}{2}}(\mathbf{x} - \boldsymbol{\mu}) = \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{U}^T(\mathbf{x} - \boldsymbol{\mu}). \tag{2.13}$$

  In the case of $\boldsymbol{\Gamma}$ not having full rank, which means that the training data is fully contained in a space of lower dimensionality, or if a dimensionality reduction should be performed for other reasons, use

$$\tilde{\mathbf{y}} = \boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\Phi}^T(\mathbf{x} - \boldsymbol{\mu}). \tag{2.14}$$

If we know, or can estimate, the statistics of the noise (for example, from the sensor characteristics), we can use the combined covariance matrix $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}_S\boldsymbol{\Gamma}_N^{-1}$, where $\boldsymbol{\Gamma}_S$ is the covariance of the signal and $\boldsymbol{\Gamma}_N$ is the covariance of the noise, for dimensionality reduction. This is called *minimum noise fractions* (MNF) [1, 4]. If the sensor noise is white, this is equivalent to PCA.

2.3 Spectral modelling

Assume that we have a source (for example an electro-optical sensor) outputting a sequence of samples (measurements). Having no knowledge of the inner workings of the source, we regard the samples as realisations of a random variable and use the samples to build a model of the source. The model also gives us a measure telling us how well each new sample is described by the model (or, the other way around, a distance from the new sample to the model). Note that when the sensor is multi- or hyperspectral, the samples are multidimensional, i.e., vector-valued and not scalar-valued.

A simple model would be to calculate the mean of the received samples so far, and for each new sample, the deviation from the mean is computed. A large deviation is to be regarded as an *anomaly* and the scheme is thus a simple *anomaly detector*. Below, we will refer to the samples used for calculating the model parameters as the *training samples* and the new samples as the *test sample(s)*.

Naturally, we might have some knowledge on the source that we can exploit when selecting and training the model. For example, we might assume that all samples are linear mixtures of a set of end-members.

Models are not necessarily trained from the actual sensor data, but might also originate from simulations and/or libraries with spectral signatures. If we, for example, are looking for certain materials in the scene and know their spectral signatures,

we can compare the test samples to those signatures and report similar samples as detections. We call this *signature-based target detection* or just *target detection*.

It is of fundamental interest if we are looking for *resolved* or *subpixel* targets. In the case of resolved targets, a sample originates either from the background or a target, whereas in the subpixel target case, a target sample might be a mixture of target and background spectra.

In the following chapters, we will describe different models and distance measure for detection. The treatment is general in the way that it is not limited to spectral measurements, but can be applied to any type of vector-valued samples.

2.4 Spectral detection algorithms

Target detection is, in this context, about finding pixels (samples, spectral vectors) in images that

- does *not* correspond to some model of the background spectral signature

and/or

- *does* correspond to a target model.

The case when a target model is available, we here call *signature-based target detection*, while the process of detecting an unknown target is called *anomaly detection*. Target detection is discussed briefly in Section 2.4.2 and anomaly detection in Section 2.4.1.

In our notation, the *detector* is a function

$$D\colon \mathcal{R}^N \to \{\text{true,false}\}, \tag{2.15}$$

telling if a (spectral) test vector is a target or not.

Related terms are *target classification*, i.e., to classify the (detected) target(s) as a specific type of target. This is not treated in this document. *Clustering* or *unsupervised classification* is the process of separating a set of vectors into different clusters or classes, and is discussed in Chapter 5.

2.4.1 Anomaly detection Anomaly detection is the case when we do not know the spectral signature of the target, and we try to find pixels that deviate from the background. We use a background model $\mathcal{B}$, a distance measure $d(\cdot)$, and a threshold t. We regard a pixel $\mathbf{x}$ as an anomaly if $d(\mathbf{x}, \mathcal{B}) > t$, and the detector is thus given by

$$D(\mathbf{x}|\mathcal{B}) = [\, d(\mathbf{x}, \mathcal{B}) > t \,]\,. \tag{2.16}$$

To exemplify, recall the example in Section 2.3, where we record the mean vector of the training samples. The model consists of the mean vector $\boldsymbol{\mu}$, and the distance measure is the Euclidean distance, i.e.,

$$D(\mathbf{x}|\mathcal{B}) = [\, \|\mathbf{x} - \boldsymbol{\mu}\| > t \,]\,. \tag{2.17}$$

Thus, a model for the background signature is needed, as well as a spatial model, i.e., from where to choose the spectral vectors to train the model. For example, we could use a local model (estimating the background signature from a local neighbourhood only) or a global model (using all available image data). Spatial models are disccused in Chapter 6.

Then, to measure the distance from each pixel signature to the background model, we need a distance measure. The choice of distance measure is restricted, or even determined, by the model used for the background and thus the assumptions about background spectral distribution.

Models and distance measures are discussed futher in Chapters 3–5.

Finally, we need to set the threshold t. A high threshold will give few detections, reducing the *detection rate* (DER), but also the *false-alarm rate* (FAR).

2.4.2 Signature-based target detection A signature-based algorithm for target detection searches for pixels that are similar to a *target probe*. The target probe is a model of a certain target signature $\mathcal{T}$ i.e., the spectral signature of the target or target class is known. In contrast, the anomaly detection discussed above assumes no such knowledge. Basically, we measure the distance from a pixel signature to the target model. That is, we can classify pixel $\mathbf{x}$ as a target pixel if $d(\mathbf{x}, \mathcal{T}) < t$ and the corresponding detector is thus

$$D(\mathbf{x}|\mathcal{T}) = [\, d(\mathbf{x}, \mathcal{T}) < t \,]\,. \tag{2.18}$$

Usually, we incorporate background suppression in our target detection scheme in order to enhance detection performance. There are basically three ways of doing this:

- **Separate thresholds**. First, run an anomaly detector

$$D_A(\mathbf{x}|\mathcal{B}) = [\, d(\mathbf{x}, \mathcal{B}) > t_A \,]\,. \tag{2.19}$$

  All pixels marked as anomalies are then investigated by the target detector

$$D_T(\mathbf{x}|\mathcal{T}) = [\, d(\mathbf{x}, \mathcal{T}) < t_T \,]\,. \tag{2.20}$$

  The advantage is that several different target detectors can be applied to only a small amount of the test samples.

- **Direct comparison**. Run the anomaly detector and the target detector on all test samples and use the compare the results:

$$D(\mathbf{x}|\mathcal{B}, \mathcal{T}) = \left[\, \frac{d(\mathbf{x}, \mathcal{T})}{d(\mathbf{x}, \mathcal{B})} > t \,\right]. \tag{2.21}$$

- **Combined detector**. For certain models, a combined detector $D(\mathbf{x}|\mathcal{B}, \mathcal{T})$ can be derived. That is, instead of measuring a distance to the target and a distance to the background, a joint measure is derived.

2.5 Performance measures

By changing the threshold t above, the detection rate (DER) and the false alarm rate (FAR) can be varied. FAR and DER correspond to the probabilites of detection ($P_{\mathsf{D}}(t)$) and false alarm ($P_{\mathsf{FA}}(t)$) respectively.

Unfortunately both are increased (decreased) simultaneously whereas the wish would be to increase the detection rate and still keep the false alarm rate low. Thus, FAR and DER must be related to be meaningful—it is easy to create a detector with 100% detection rate if no requirement is set on the false alarm rate.

There are a few common ways of presenting the performance of a detector:

- The *receiver operator characteristics* (ROC) is a graph with FAR and DER on the axes, and a curve showing DER as a function of FAR (found by varying the threshold), i.e., a parametric curve

$$\mathbf{x}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} \mathrm{FAR}(t) \\ \mathrm{DER}(t) \end{pmatrix}, \tag{2.22}$$

  where t is varied so that FAR and DER goes from zero to one.

- The *FAR at first detection* (FFR) is the false alarm rate when the first pixel of a certain target is detected, giving an indication of the minimum achievable FAR for that type of target, detector, and so on.

- The *area under curve* (AUC) is the integral of the ROC, giving one scalar value describing the performance of the detector. Since the performance of the detector at high FAR is less intresting, the integral is sometimes computed over an interval FAR $= [0, I]$, for example $\mathrm{AUC}_{0.01}$ is defined by $\int_{t=t_0}^{t_{0.01}} \mathrm{DER}(t)dt$ where the limits are defined by $\mathrm{FAR}(t_c) = c$.

3. Detectors using Unstructured Background Models

Using an unstructured model, we infer no specific structure on the data. We incorporate any additive noise in the model, and make no assumptions based on a priori knowledge. The unstructured models are also called probabilistic, statistical, and/or data-driven.

3.1 Probabilistic models

As mentioned earlier, the simplest conceivable model is to compute the mean of the training samples and use as a model for the source. In that case, our model of a (background or target) class $\mathcal{C}$ consists of a *prototype vector* $\boldsymbol{\mu}$ (the mean) and we use the squared Euclidian distance to the prototype as distance, i.e.,

$$d_{\mathsf{E}}(\mathbf{x}, \mathcal{C}) \triangleq \|\mathbf{x} - \boldsymbol{\mu}\|^2 = (\mathbf{x} - \boldsymbol{\mu})^T(\mathbf{x} - \boldsymbol{\mu}). \tag{3.1}$$

The only advantage of this detector is its simplicity and the fact that we need only one training sample.

Being just a little bit more sophisticated, we might use the within-class variance σ^2 for weighting the distance, i.e.,

$$d_{\mathsf{E}'}(\mathbf{x}, \mathcal{C}) \triangleq d_{\mathsf{E}}(\mathcal{C}, \mathbf{x})/\sigma^2. \tag{3.2}$$

However, the within-class variance might differ significantly for the different dimensions (spectral bands), so let us instead use different weights for different dimensions:

$$\begin{aligned} d_{\mathsf{E}''}(\mathbf{x}, \mathcal{C}) &\triangleq \sum_{n=1}^{N} \frac{(x_n - \mu_n)^2}{\sigma_n^2} \\ &= (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}), \end{aligned} \tag{3.3}$$

where $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_N^2)$.

In practice, the different samples are often correlated between the different dimensions, and we can take this into consideration by using the full covariance matrix for weighting, i.e.,

$$d_{\mathsf{M}}(\mathbf{x}, \mathcal{C}) \triangleq (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Gamma}^{-1} (\mathbf{x} - \boldsymbol{\mu}), \tag{3.4}$$

which is the *Mahalanobis distance*. This is proportional to the log-likelihood function for a Gaussian, or normal, distribution, i.e.,

$$d_{\mathsf{M}}(\mathbf{x}, \mathcal{C}) \propto -\log p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Gamma}), \tag{3.5}$$

where

$$p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Gamma}) \triangleq \frac{1}{(2\pi)^{\frac{N}{2}} |\boldsymbol{\Gamma}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Gamma}^{-1} (\mathbf{x}-\boldsymbol{\mu})} \tag{3.6}$$

Thus, if we decide to model the source as a multivariate Gaussian distribution, then the Mahalanobis distance is an adequate distance measure, which also means that we can directly relate it to the Bayes solution as will be shown below.

Note that measuring the Euclidean distance after whitening with respect to$\mathcal{C}$ (see Section 2.2) is identical to measuring the Mahalanobis distance.

The Euclidean and Mahalanobis distances are illustrated in Figure3.1.

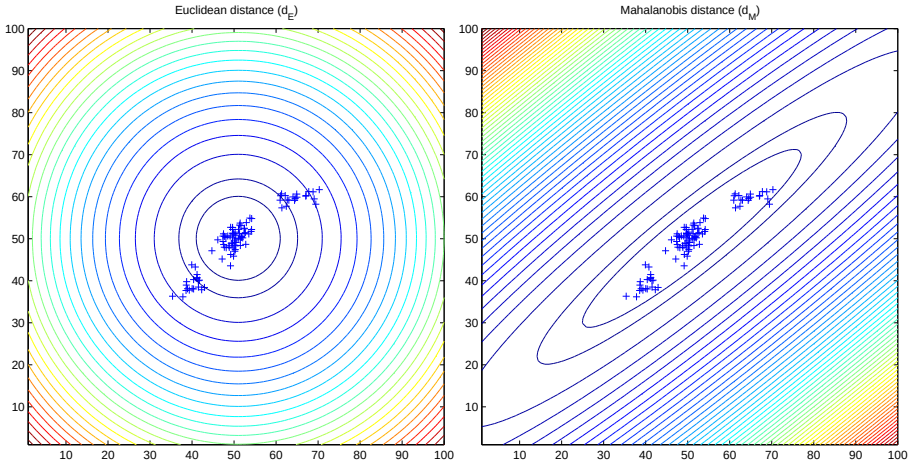


Figure 3.1: Isocurves of the Euclidean and Mahalanobis distances.

3.1.1 Estimating model parameters Given a set of N-dimensional training vectors $\{\mathbf{x}_k\}_{k=1}^K$, we can estimate the mean of the random variable as

$$\hat{\boldsymbol{\mu}} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k. \tag{3.7}$$

The variances σ^2 and $\{\sigma_n^2\}_{n=1}^N$ and the covariance $\boldsymbol{\Gamma}$ can be estimated as well:

$$\hat{\boldsymbol{\Gamma}} = \frac{1}{K-1}\sum_{k=1}^{K}(\mathbf{x}_k-\hat{\boldsymbol{\mu}})(\mathbf{x}_k-\hat{\boldsymbol{\mu}})^T \tag{3.8}$$

$$\hat{\sigma}_n^2 = \frac{1}{K-1}\sum_{k=1}^{K}((\mathbf{x}_k)_n-\hat{\mu}_n)^2 \tag{3.9}$$

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{n=1}^{N}\hat{\sigma^2}_n \tag{3.10}$$

Note that we need more training vectors the higher the dimensionality and for each level of complexity the model has. If we have only one training vector, we can only estimate the mean, and not very reliably—any textbook on statistics will tell you the variance of the estimate. In that case, we use the squared Euclidean distance measure (equivalent to modelling the class as a Gaussian with unit covariance).

To estimate a full covariance we need, as a rule of thumb, at least N^2 training vectors to get a reliabe estimate.

Since d_{E}, $d_{\mathsf{E}'}$, $d_{\mathsf{E}''}$ are special cases of d_{M} (Gaussians with covariances $\mathbf{I}$, $\sigma^2\mathbf{I}$, and $\mathrm{diag}(\sigma_1^2,\ldots,\sigma_N^2)$ respectively), they are not treated separately in the rest of this document.

3.1.2 Anomaly detection using a Gaussian background model If we model the background as a multivariate Gaussian distribution, the resulting anomaly detector is

$$\begin{aligned} D_{\mathsf{RX}}(\mathbf{x}|\mathcal{B}) &= [\, d_{\mathsf{M}}(\mathbf{x},\mathcal{B}) > t \,] \\ &= \left[\, (\mathbf{x}-\hat{\boldsymbol{\mu}})^T \hat{\boldsymbol{\Gamma}}^{-1} (\mathbf{x}-\hat{\boldsymbol{\mu}}) > t \,\right] \\ &= \left[\, \|\tilde{\mathbf{x}}\|^2 > t \,\right], \end{aligned} \tag{3.11}$$

commonly known as the RX-detector [8]. t is a parameter controlling the detection and false alarm rates. Since the Mahalanobis distance is χ^2-distributed under the null hypothesis (no target), t can be set to achieve a specific (constant) false alarm rate.

Since we make no assumptions whatsoever about the targets, this detector is valid for subpixel as well as for resolved targets. Naturally, resolved targets give significantly better detection performance.

3.1.3 Gaussian target model We recall the likelihood ratio (2.7) and model the background and target classes $\mathcal{B}$ and $\mathcal{T}$ as multivariate Gaussian distributions. For example, the two distributions $p_B(x)$ and $p_T(x)$ corresponding to H_0 and H_1 could be the distributions for background and target spectra respectively. The competing hypotheses are:

$$\begin{aligned} \mathsf{H}_0 &: \ \mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_B, \boldsymbol{\Gamma}_B) \\ \mathsf{H}_1 &: \ \mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_T, \boldsymbol{\Gamma}_T). \end{aligned} \tag{3.12}$$

Inserting (3.5) in (2.7) we see that

$$\begin{aligned} \log \Lambda(\mathbf{x}) &= \log p_T(\mathbf{x}) - \log p_B(\mathbf{x}) \\ &\propto d_{\mathsf{M}}(\mathbf{x},\mathcal{B}) - d_{\mathsf{M}}(\mathbf{x},\mathcal{T}), \end{aligned} \tag{3.13}$$

i.e., the vector that (simultaneously) maximizes the distance to the background model and minimizes the distance to the target model maximizes the likelihood of the vector being a target. Given costs and a priori probabilities, we classify a sample as a target if

$$d_{\mathsf{M}}(\mathbf{x},\mathcal{B}) - d_{\mathsf{M}}(\mathbf{x},\mathcal{T}) > \Lambda' = 2\log \frac{P_0 C_{10} |\boldsymbol{\Gamma}_B|^{\frac{1}{2}}}{(1-P_0) C_{01} |\boldsymbol{\Gamma}_T|^{\frac{1}{2}}}. \tag{3.14}$$

The resulting *generalized likelihood ratio* (GLRT) detector is thus

$$D_{\mathsf{B}}(x|\mathcal{B},\mathcal{T}) = [\, d_{\mathsf{M}}(\mathbf{x},\mathcal{B}) - d_{\mathsf{M}}(\mathbf{x},\mathcal{T}) > t \,]. \tag{3.15}$$

The decision line for two (two-dimensional) Gaussian models is illustrated in Figure 3.2.

The hypotheses (3.12) are stated for resolved targets, but since the models incorporate additive noise, the same detector is valid for subpixel targets.

3.1.4 Known target signature The situation where we try to detect a known signal disturbed by additive Gaussian noise is common in communication systems [7], for examle, a specific radio wave might be transmitted and the receiver receives the signal plus background noise. In remote sensing, the signal would be the spectral signature for a certain material, and the noise would be the background clutter, atmospheric effects, and sensor noise. The hypotheses are

$$\begin{aligned} \mathsf{H}_0 &: \ \mathbf{x} = \mathbf{b}, \qquad \mathbf{b} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma}) \\ \mathsf{H}_1 &: \ \mathbf{x} = \mathbf{t} + \mathbf{b}. \end{aligned} \tag{3.16}$$

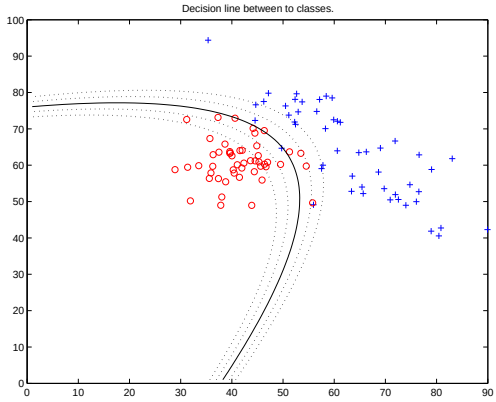


Figure 3.2: Samples from two Gaussian distributions (with different mean and covariance) and the decision line separating them for different decision levels.

If the noise $\mathbf{b}$ is coloured, i.e., $\mathbf{\Gamma} \neq \sigma^2\mathbf{I}$, the problem is simplified by whitening $\mathbf{x}$ with respect to $\mathbf{b}$,

$$\tilde{\mathbf{x}} = \mathbf{\Gamma}^{-\frac{1}{2}}(\mathbf{x} - \boldsymbol{\mu}), \tag{3.17}$$

giving us the simpler hypotheses

$$\begin{aligned} \mathsf{H}_0 &: \ \tilde{\mathbf{x}} = \tilde{\mathbf{b}}, \qquad \tilde{\mathbf{b}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \mathsf{H}_1 &: \ \tilde{\mathbf{x}} = \tilde{\mathbf{t}} + \tilde{\mathbf{b}}. \end{aligned} \tag{3.18}$$

This signal can then be correlated with the *matched filter* $\tilde{\mathbf{t}}$, giving us a scalar output $x = \tilde{\mathbf{t}}^T\tilde{\mathbf{x}}$ and the one-dimensional problem

$$\begin{aligned} \mathsf{H}_0 &: \ x = \mathsf{b}, \qquad \mathsf{b} \sim \mathcal{N}(0, 1) \\ \mathsf{H}_1 &: \ x = \|\tilde{\mathbf{t}}\|^2 + \mathsf{b}, \end{aligned} \tag{3.19}$$

which can be directly inserted in the GLRT framework above.

Creating $x' = \frac{x}{\|\tilde{\mathbf{t}}\|^2}$ to make the mean (under H_1) equal one, the resulting detector is

$$\begin{aligned} D_{\mathsf{AMF}}(\mathbf{x}|\mathcal{B}, \mathcal{T}) &= \left[\frac{\tilde{\mathbf{t}}^T\tilde{\mathbf{x}}}{\|\tilde{\mathbf{t}}\|^2} > t \right] \\ &= \left[\frac{(\mathbf{t} - \boldsymbol{\mu})^T\mathbf{\Gamma}^{-1}(\mathbf{x} - \boldsymbol{\mu})}{(\mathbf{t} - \boldsymbol{\mu})^T\mathbf{\Gamma}^{-1}(\mathbf{t} - \boldsymbol{\mu})} > t \right]. \end{aligned} \tag{3.20}$$

This is called the *adaptive matched filter* (AMF) detector. This detector is optimum only when the target and background follow the same (Gaussian) distribution, which in real applications is highly unlikely.

3.1.5 Known target signature with unknown norm If we know the target signature except for the norm, for example in the subpixel case or due to transmission

effects, we instead have the following hypotheses:

$$\begin{aligned} \mathsf{H}_0 &: \ \mathbf{x} = \mathbf{b}, \qquad \mathbf{b} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma}) \\ \mathsf{H}_1 &: \ \mathbf{x} = k\mathbf{t} + \mathbf{b}, \end{aligned} \tag{3.21}$$

where k is an unknown parameter.

An alternative formulation is that we know the *structure* but not the *level* of the noise.

The angle α between $\mathbf{x}$ and $\mathbf{t}$ is uniformly distibuted in the interval $[-\pi, \pi]$ under H_0 and centered around zero under H_1. The resulting detector is called the *spectral angle mapper* (SAM). Using the cosine instead of the angle, we get

$$\begin{aligned} D_{\mathsf{SAM}}(\mathbf{x}|\mathcal{T}) &= [\cos\alpha > t\ ,] \\ &= \left[\frac{\mathbf{t}\cdot\mathbf{x}}{\|\mathbf{t}\|\|\mathbf{x}\|} > t \right]. \end{aligned} \tag{3.22}$$

The disitribution of the correlation between two random variables have been studied extensively [3].

3.1.6 Subspace target model Modelling target variability as a linear combination $\boldsymbol{\Phi}_T\mathbf{a}_T$ of target exemplars or target subspace basis vectors, we get the following hypotheses

$$\begin{aligned} \mathsf{H}_0 &: \ \mathbf{x} = \mathbf{b} \\ \mathsf{H}_1 &: \ \mathbf{x} = \boldsymbol{\Phi}_T\mathbf{a}_T + k\mathbf{b}, \end{aligned} \tag{3.23}$$

where k is related to fill factor of the target, i.e., how large part of the pixel that is occupied by the target. The coefficient vector $\mathbf{a}$ incorporates the fill factor to simplify notation.

We use a Gaussian backgound model $\mathbf{b} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma})$ and whiten $\mathbf{x}$ as well as $\boldsymbol{\Phi}_T$ with respect to the background according to (2.13). Still following the generalized likelihood approach, the resulting detector is

$$\begin{aligned} D_{\mathsf{ACE}}(\mathbf{x}|\mathcal{B},\mathcal{T}) &= \left[\frac{\tilde{\mathbf{x}}^T\tilde{\mathbf{P}}_T\tilde{\mathbf{x}}}{\tilde{\mathbf{x}}^T\tilde{\mathbf{x}}} > t \right] \\ &= \left[cos^2\alpha > t \right], \end{aligned} \tag{3.24}$$

where $\tilde{\mathbf{P}}_T$ is the projection and reconstruction operator onto the whitened target subspace, i.e.,

$$\tilde{\mathbf{P}}_T = \tilde{\boldsymbol{\Phi}}_T(\tilde{\boldsymbol{\Phi}}_T^T\tilde{\boldsymbol{\Phi}}_T)^{-1}\tilde{\boldsymbol{\Phi}}_T^T, \tag{3.25}$$

and α is the angle between the whitened target subspace and the whitened test vector. The detector is called the *adaptive coherence/cosine detector* (ACE) [6].

Linear subspaces will be discussed further in the next chapter.

3.2 Nearest neighbour

Observing that the background signature vectors being (spectrally) closest to the test vector(s) probably are the most important, a natural approach would be to consider these vectors only. Instead of representing the background class with its mean we could use a weighted mean

$$\boldsymbol{\mu}' = \frac{1}{K}\sum_{k=1}^{K} w_k\mathbf{x}_k \tag{3.26}$$

where $\{\mathbf{x}_k\}$ are the background vectors and w_k is a weight depending on the distance between $\mathbf{x}_k$ and the test vector $\mathbf{x}$. To simplify notation, assume that the vectors are sorted according to their distance to $\mathbf{x}$ (so that $\mathbf{x}_1$ is the closest). The simplest weighting would be

$$w_k = \begin{cases} 1 & \text{if } k \leq C \\ 0 & \text{if } k > C \end{cases} \tag{3.27}$$

(*C nearest neighbours*). The simplest case, $C = 1$, i.e., the distance to the nearest neighbour, is illustrated in Figure 3.3. The detector is

$$D_{\mathsf{NN}}(\mathbf{x}|\mathcal{B}) = \left[\|\mathbf{x} - \frac{1}{K}\sum_{k=1}^{K} w_k \mathbf{x}_k\| > t \right] \tag{3.28}$$

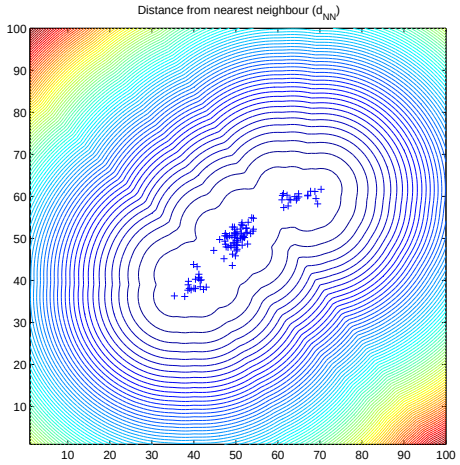


Figure 3.3: Isocurves of the nearest neighbour distances.

3.3 Summary of detectors using unstructured background models

In this chapter we have studied detectors using unstructured background models. Depending on which model we use for target signatues, we get different detectors. The background models and distances we have studied are the Gaussian distribution with the corresponding Mahalanobis distance (with (weighted) Euclidean distance as a special case), spectral angle mapper, and the nearest neighbour.

Table 3.1 summarizes the various detectors.

Table 3.1: Target models and corresponding detectors using an unstructured background model.

Target model		**Detector**	
None		RX	Eq. 3.11
None		NN	Eq. 3.28
Known	$\mathbf{t}$	AMF	Eq. 3.20
Known up to norm	$k\,\mathbf{t}$	SAM	Eq. 3.22
Gaussian	$\boldsymbol{\Gamma}_T, \boldsymbol{\mu}_T$	Bayesian	Eq. 3.15
Subspace	$\boldsymbol{\Phi}_T$	ACE	Eq. 3.24

4. Detectors using Structured Background Models

It is important to realize that all distance measures imply a model for the within-class distribution. In each case, the models should be checked for physical validity, i.e., is this model relevant regarding the physical reality the data is sampled from?

Using a *structured model* we infer some kind of structure attained from a priori knowledge on the data. In spectral imaging, the inferred structure should be related to the underlying physics of the observed source. Considering that an observed spectrum is a mixture of the spectra corresponding to the materials in the pixel's footprint, and assuming linearity in the mixing process, all observed spectra should lie in the subspace spanned by the spectra of the materials in the scene. Thus, a linear subspace model should be useful. The subspace could be computed from a spectral library or directly from data, as described below. Note that spectra from a library need to be modified according to the current sensing conditions, i.e., weather and light conditions.

4.1 Linear subspaces

Assume, for the sake of illustration, that our data samples are two-dimensional, and consider the toy example in in Figure 4.1 where our training samples are plotted. Apparently, the data is mainly distributed along a line, i.e., in a one-dimensional subspace of the two-dimensional data space. We call the one-dimensional subspace (the line) the *feature space* of the data, and we use the (squared) Euclidean distance to the feature space as distance measure.

A linear subspace is represented by a (1) set of basis vectors that spans the subspace, and (2) an offset vector. The offset vector is often omitted, thus assuming that the origin is included in the subspace.

Assume that a subspace is spanned by the vectors $\{\mathbf{a}_1, \ldots, \mathbf{a}_K\}$. Any vector $\mathbf{x}$ in the subspace can then be written as a linear cominbation

$$\mathbf{x} = \sum_{k=1}^{K} a_k \mathbf{a}_k = \mathbf{A}\mathbf{a}, \tag{4.1}$$

where $\mathbf{a}$ is the coefficient or weight vector.

We define the *projection matrix* for the the subspace $\mathbf{A}$ as

$$\mathbf{P}_A \triangleq \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T, \tag{4.2}$$

that is, the component of a vector $\mathbf{x}$ within the subspace $\mathbf{A}$ is

$$\hat{\mathbf{x}} = \mathbf{P}_A\mathbf{x}. \tag{4.3}$$

This is called the approximation or reconstruction of $\mathbf{x}$ by the subspace $\mathbf{A}$.

The component of $\mathbf{x}$ begin perpendicular to $\mathbf{A}$, i.e., the residual when approximating $\mathbf{x}$ using $\mathbf{A}$, is given by

$$\mathbf{r} = \mathbf{x} - \hat{\mathbf{x}} = \mathbf{P}_A^{\perp}\mathbf{x}, \tag{4.4}$$

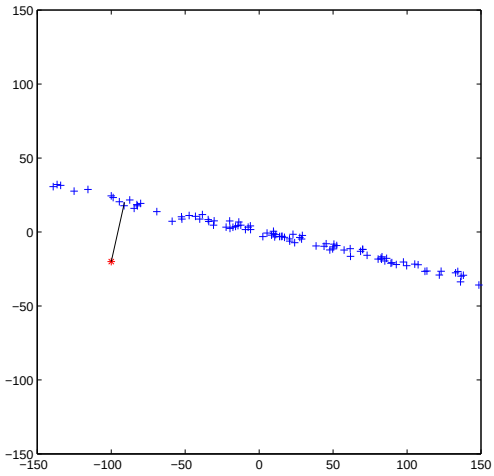


Figure 4.1: Measuring the distance to a one-dimensional feature space in a two dimensional space.

where

$$\begin{aligned}\mathbf{P}_A^{\perp} &\triangleq \mathbf{I} - \mathbf{P}_A \\ &= \mathbf{I} - \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\end{aligned} \tag{4.5}$$

The squared norm of the residual vector is given by $\|\mathbf{r}\|^2 = \mathbf{x}^T\mathbf{Q}_A\mathbf{x}$ where

$$\mathbf{Q}_A \triangleq (\mathbf{P}_A^{\perp})^T\mathbf{P}_A^{\perp}. \tag{4.6}$$

This is called the *distance from feature space* (DFFS)

$$d_{\mathsf{FFS}}(\mathbf{x}, \mathcal{A}) \triangleq \mathbf{x}^T\mathbf{Q}_A\mathbf{x}. \tag{4.7}$$

and is illustrated in Figure 4.2

If the origin is not included in the subspace, an offset vector $\boldsymbol{\mu}_A$ is needed as well. This can in fact be any vector in the subspace. The expression for reconstruction becomes

$$\hat{\mathbf{x}} = \boldsymbol{\mu}_A + \mathbf{P}_A(\mathbf{x} - \boldsymbol{\mu}_A). \tag{4.8}$$

4.1.1 Estimating model parameters If we have a few noise free examples of signatures (i.e., training vectors) that we want to use as model for a linear subspace, we can simply merge them to a subspace basis matrix as in (4.1). This is the typical case if we have a few target signatures in a data base.

However, if the training set is noisy, we need a larger set and a statistical method for eliminating the noise. Also, if our training set is large compared to the dimensionality M of the subspace (i.e., our training vectors are linearly dependent), we might simplify our computations by creating an orthonormal basis matrix $\boldsymbol{\Phi}_A$ with M columns instead of K. The common solution to both problems is to use PCA as described in Section 2.2.

Below, we will use $\boldsymbol{\Phi}_A$ for the basis matrix that represents the subspace model $\mathcal{A}$ regardless if it us computed using PCA or if $\boldsymbol{\Phi} = \mathbf{A}$.

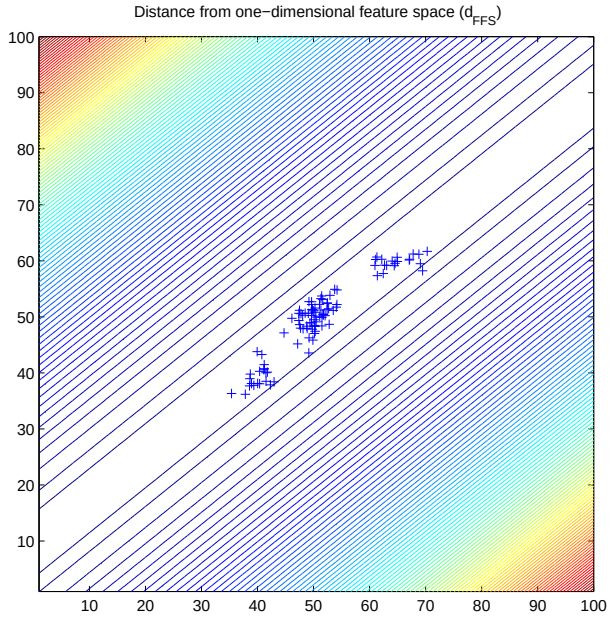


Figure 4.2: Isocurves of the distance to a one-dimensional feature space in a two-dimensional data space.

4.1.2 Anomaly detection using a subspace background model Modelling the background as an M-dimensional subspace plus noise, and assuming no model for the targets, the distance from feature space gives us the anomaly detector

$$\begin{aligned} D_{\mathsf{DFFS}}(\mathbf{x}|\mathcal{B}) &= \left[\, d_{\mathsf{FFS}}(\mathbf{x},\mathcal{B}) > t \,\right] \\ &= \left[\, \mathbf{x}^T\mathbf{Q}_B\mathbf{x} > t \,\right]. \end{aligned} \tag{4.9}$$

Note that this is basically the same as the RX detector if the variances are thresholded so that the the M largest eigenvalues are set to infinity and the others to one.

4.1.3 Known target signature A widely used detector is the *orthogonal subspace projector* (OSP) [2]. It uses a subspace model for the background and a single signature vector $\mathbf{t}$ as target model:

$$\begin{aligned} \mathsf{H}_0 &: \ \mathbf{x} = \mathbf{\Phi}_B\mathbf{a}_B + \mathbf{n}, \qquad \mathbf{n} = \mathcal{N}(\mathbf{0},\sigma_n^2\mathbf{I}) \\ \mathsf{H}_1 &: \ \mathbf{x} = \mathbf{\Phi}_B\mathbf{a}_B + \mathbf{t} + \mathbf{n} \end{aligned} \tag{4.10}$$

By removing the component of $\mathbf{x}$ within the background subspace and matching with the target signature we get the detector

$$D_{\mathsf{OSP}}(\mathbf{x}|\mathcal{T},\mathcal{B}) = \left[\, \mathbf{t}^T\mathbf{P}_B^{\perp}\mathbf{x} > t \,\right]. \tag{4.11}$$

This is basically the same as the AMF detector in the subspace complementary to the background subspace.

4.1.4 Subspace target model Modelling the background as well as target variablility with subspaces gives us two different detectors depending on if we assume full-pixel targets or not.

- In the subpixel case we have the following two hypotheses:

$$\begin{aligned} \mathsf{H}_0 &: \ \mathbf{x} = \mathbf{\Phi}_B \mathbf{a}_B + \mathbf{n}, \qquad\qquad \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma_{\mathbf{n}}^{\mathbf{2}}\mathbf{I}) \\ \mathsf{H}_1 &: \ \mathbf{x} = \mathbf{\Phi}_B \mathbf{a}_B + \mathbf{\Phi}_T \mathbf{a}_T + \mathbf{n} = \mathbf{\Phi}_A \mathbf{a}_A + \mathbf{n} \end{aligned} \tag{4.12}$$

  giving us the detector

$$\begin{aligned} D_{\mathsf{ASD}}(\mathbf{x}|\mathcal{B},\mathcal{T}) &= \left[\frac{d_{\mathsf{FFS}}(\mathbf{x},\mathcal{B}) - d_{\mathsf{FFS}}(\mathbf{x},\mathcal{A})}{d_{\mathsf{FFS}}(\mathbf{x},\mathcal{A})} > t \right] \\ &= \left[\frac{\mathbf{x}^T(\mathbf{Q}_B - \mathbf{Q}_A)\mathbf{x}}{\mathbf{x}^T\mathbf{Q}_A\mathbf{x}} > t \right], \end{aligned} \tag{4.13}$$

  where $\mathcal{A}$ is the combined target and background subspace model ($\mathbf{\Phi}_A = [\mathbf{\Phi}_B \ \mathbf{\Phi}_T]$ and $\mathbf{a}^T = [\mathbf{a}_B^T \ \mathbf{a}_T^T]^T$. This detector is known as the *adaptive subspace detector* (ASD). In statistics, it is known as the F-test, and the random variable is F-distributed. The false alarm rate is constant and specified by

$$P_{\mathsf{FA}}(t) = 1 - F_{M_t, N-M_t-M_b}(0,t), \tag{4.14}$$

  where M_t and M_b are the dimensionalities of the target and background spaces respectively.

- In the full-pixel case we have the following two hypotheses:

$$\begin{aligned} \mathsf{H}_0 &: \ \mathbf{x} = \mathbf{\Phi}_B \mathbf{a}_B + \mathbf{n}, \qquad \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2\mathbf{I}) \\ \mathsf{H}_1 &: \ \mathbf{x} = \mathbf{\Phi}_T \mathbf{a}_T + \mathbf{n} \end{aligned} \tag{4.15}$$

  giving us the detector

$$\begin{aligned} D_{\mathsf{ASD}'}(\mathbf{x}|\mathcal{B},\mathcal{T}) &= \left[\frac{d_{\mathsf{FFS}}(\mathbf{x},\mathcal{B})}{d_{\mathsf{FFS}}(\mathbf{x},\mathcal{T})} > t \right] \\ &= \left[\frac{\mathbf{x}^T\mathbf{Q}_B\mathbf{x}}{\mathbf{x}^T\mathbf{Q}_T\mathbf{x}} > t \right]. \end{aligned} \tag{4.16}$$

4.2 Linear mixing models

The reasoning around linear subspaces above is useful, but actually somewhat flawed. If the observed spectrum $\mathbf{x}$ is mixture of a set of M end-members $\mathbf{e}_m$, i.e.,

$$\mathbf{x} = \sum_{m=1}^{M} a_m \mathbf{e}_m = \mathbf{E}\mathbf{a}, \tag{4.17}$$

where the coefficient a_m describe the proportion of the mth end-member ($\mathbf{e}_m$) in the sample, then the possible values of $\mathbf{x}$ do not fill the entire subspace spanned by $\mathbf{E}$. Instead, only the convex hull of $\{\mathbf{e}_m\}$ should be considered as non-anomalous.

If a spectral library of materials is available, and we assume that M of these materials/spectra are present in the scene, we must relate M to the data dimensionality N. If $M << N$, then the difference between using a linear subspace model and a linear mixing model is probably insignificant. However, if the end-members span the spectral space (or a large part thereof), a subspace model is quite useless.

Note that the linear mixing model is an approximation of the physical reality. When estimating the coefficients a_m to investigate the proportions of various materials in a scene, a non-linear mixing model is more accurate. However, for the purposes in this report, the linear mixing model is satisfactory.

4.2.1 Estimating model parameters There are several approaches to end-member extraction from data. One fundamental difference is if they assume the presence of pure pixels in the training data, i.e., do they assume that the end-members are present, or are they extrapolated?

A basic approach is to search for a simplex, i.e., a polyhedron with $M = N + 1$ corners, where the corners (vertices) are the end-members. If $M < N+1$ end-members is searched for, the simplex search must be preceded by a dimensionality reduction step, searching for end-members in a $(M-1)$-dimensional subspace.

In the pure pixel case, a popular approach is the N-FINDR [9] algorithm, see Appendix A. N-FINDR traverses the samples in the training data and finds the $M = N + 1$ samples that forms the simplex with the maximum volume.

In the mixed pixel case, we instead search for the minimum simplex containing all the training data. The vertices are thus extrapolated from the simplex vertices. This is a somewhat more complicated procedure.

4.2.2 Anomaly detection using a linear mixture model The anomaly detector is given by the distance from a vector $\mathbf{x}$ to the convex hull of the end-members

$$\begin{aligned} D_{\mathsf{LMM}}(\mathbf{x}|\mathcal{B}) &= [\, d_{\mathsf{LMM}}(\mathbf{x}, \mathcal{B}) > t \,] \\ &= \left[\, \|\mathbf{x} - \mathbf{E}\mathbf{a}\|^2 > t \,\right], \end{aligned} \tag{4.18}$$

where $\mathbf{a}$ is the least-squares solution to $\mathbf{E}\mathbf{a} = \mathbf{x}$ constrained by

$$\begin{cases} 0 \leq a_m \leq 1 \\ \sum a_m = 1 \end{cases} \tag{4.19}$$

4.3 Summary of detectors using structured background models

In this chapter we have studied detectors when using structured background models. Depending on which model we use for target and background signatues, we get different detectors. The background models we have studied are linear subspaces and linear mixture models.

Table 4.1 summarizes the various detectors.

Table 4.1: Target models and corresponding detectors using a structured background model.

Target model		**Background model**		**Detector**	
None		Subspace	$\mathbf{\Phi}_B$	DFFS	Eq. 4.9
None		Linear mixture	$\mathbf{E}$	LMM	Eq. 4.18
Known	$\mathbf{t}$	Subspace	$\mathbf{\Phi}_B$	OSP	Eq. 4.11
Subspace	$\mathbf{\Phi}_T$	Subspace	$\mathbf{\Phi}_B$		
- subpixel				ASD	Eq. 4.13
- resolved				ASD	Eq. 4.16

5. Detectors using Cluster and Mixture Models

Consider the data set illustrated in Figure 5.1. The crosses are samples from a background distribution, and we estimate their mean and covariance in order to use a Gaussian model. When we receive a test signature vector (is this vector an anomaly or not?), we measure the Mahalanobis distance d_{M} to the background model. Using the solid isocurve as decision level, we see that the point P1 is classified as a non-anomaly when it is quite clear that it does not match the background samples. Using the same decision level, we also see that the point P2 will be classified as an anomaly, which it clearly should not.

To overcome this problem, we suggest two alternative solutions:

1. *Use a more complex model.* If the Gaussian model does not model the sampled data well, then use another one! The model that will be discussed here is a simple but very general extension of the Gaussian distributions – the *Gaussian mixture model* (GMM).

2. *Cluster the data.* The sample data in Figure 5.1 seems to come from two different sources, each having a less complex distribution. This is typically the case when we sample spectral vectors from the real world. We could then try to separate the samples that (probably) originate from each source, and use a different model for each. Thus, a test vector that is an anomaly with respect to all clusters is regarded as an anomaly.

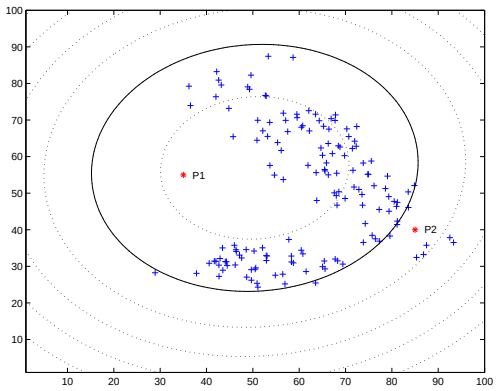


Figure 5.1: Using a Gaussian model for non-Gaussian data.

5.1 Clustering

Given a set of K training vectors (pixel spectral signatures) $\mathbf{x}_k$, a clustering algorithm organizes the vectors into clusters or classes, giving each vector a class label or a class membership value. The resulting clusters can also be used as a *classifier* that assigns class labels or class membership values to new vectors.

Classification can be supervised or unsupervised. An unsupervised classifier automatically clusters the training samples during training, while a supervised classifier also needs class labels for the training data as input. The term clustering is commonly used as synonymous to unsupervised classification.

5.1.1 Hard clustering A hard clustering assign integer class labels to the vectors, so that a vector belongs to one cluster only. The output is:

- Clustering: A *class label* $c_k \in \{1, 2, \ldots, L\}$ for each vector, telling to which class it belongs.
- Classifier: A *discrimination function* $c = f_d(\mathbf{x})$ producing a class label for a new vector $\mathbf{x}$.

5.1.2 Soft clustering A soft or fuzzy clustering assign membership values to the vectors, so a vector belongs to several clusters to a varying degree. The output is:

- Clustering: A *membership value* $m_{kl} \in [0, 1]$ for each vector in each class.
- Classifier: A *membership function* $m_l = f_m(\mathbf{x}, \mathcal{C}_l)$.

5.2 A class of clustering methods

There is a number of algorithms for clustering, of which many comply to the following scheme:

1. Assume a set of training vectors and initialize the classes randomly.
2. For each training vector, (re)compute the distance to each class.
3. Recompute the classes using the distances.
4. Repeat steps 2 and 3 until convergence.

By defining different distance functions, membership functions, and class representations we get different algorithms. The *expectation-maximization* (EM) algorithm uses the Mahalanobis distance, the *Linde-Buzo-Gray* (LBG) or *K-means* algorithm uses squared Euclidian distance, and *fuzzy C-means* uses the Euclidian distance to the power of a design variable c. The LBG algorithms assigns a discrete membership value to each vector, while the EM and Fuzzy C-means combines the distances to all classes from each vector. Variations of EM include *classification EM* and *stochastic EM*, as described in Appendix B.

5.3 Anomaly detection using a cluster model

After performing the clustering of the available background data, i.e., adapting a cluster model $\mathcal{B}$ to a set of training vectors, we can for a test vector measure $\mathbf{x}$ measure how well it fits to $\mathcal{B}$ by computing the distance

$$d_{\mathsf{C}}(\mathbf{x}, \mathcal{B}) = \min_l d(\mathbf{x}, \mathcal{B}_l). \tag{5.1}$$

Figure 5.2 illustrates this distance measure using the same data set as in Figure5.1. Both spherical clusters (LBG) and Gaussian clusters (EM) solves the example problem

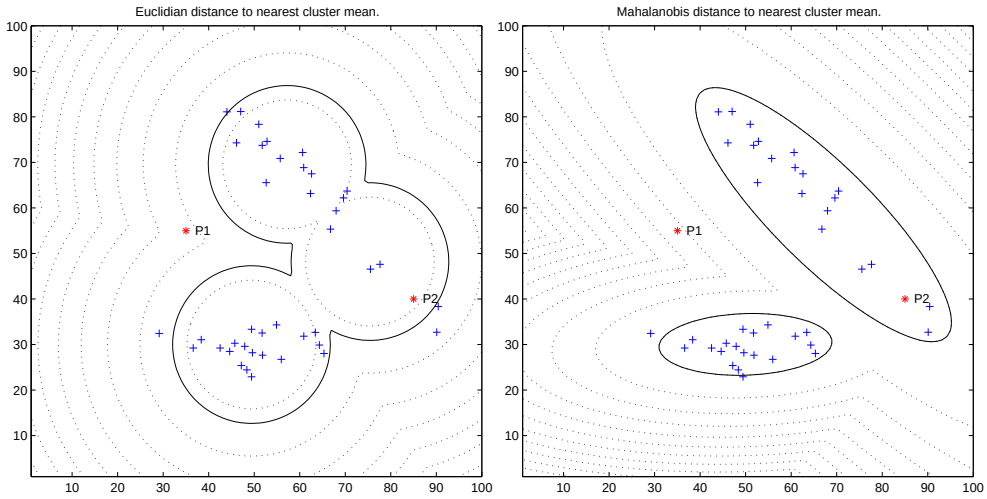


Figure 5.2: Isocurves of the distance to the nearest cluster center using the spherical clusters (left) or Gaussian clusters (right).

of classifying P1 as an anomaly and P2 as a background, however, three spherical clusters are needed compared to two Gaussian clusters.

Note that when using Gaussian clusters, this is similar to the RX detector with the spatial model defined by the cluster. Thus, the detector is sometimes calles *class conditional RX* or *Gaussian mixture RX*:

$$\begin{aligned} D_{\mathsf{GMRX}}(\mathbf{x},\mathcal{B}) &= \left[\min_l (d_{\mathsf{M}}(\mathbf{x},\mathcal{B}_l)) > t_l \right] \\ &= \left[\min_l (\,(\mathbf{x}-\boldsymbol{\mu}_l)^T \boldsymbol{\Gamma}_l^{-1} (\mathbf{x}-\boldsymbol{\mu}_l)\,) > t_l \right]. \end{aligned} \tag{5.2}$$

A threshold can be computed for a specific CFAR for each cluster.

5.4 Anomaly detection using a Gaussian mixture model

A Gaussian mixture model (GMM, a.k.a a *mixture of Gaussians* or a *multimodal Gaussian*) can be used for modelling complex mixtures. The pdf of a GMM is a weighted sum of L Gaussian distributions

$$p_{\mathsf{GMM}}(\mathbf{x}|\mathcal{C}) = \sum_{k,l} w_l \; p_{\mathcal{N}}(\mathbf{x}_k|\boldsymbol{\mu}_l,\boldsymbol{\Gamma}_l), \tag{5.3}$$

where $\mathcal{C} = \{w_l,\boldsymbol{\mu}_l,\boldsymbol{\Gamma}_l\}_{l=1}^L$.

A Gaussian mixture model (GMM) can be used for anomaly detection in the same way as a Gaussian model. While estimating the parameters of a single Gaussian is straightforward, estimating the parameters of a GMM is done using the EM algorithm (or a variation thereof). Note that a GMM can approximate any pdf arbitrarily well, given a large enough number of components (Gaussians).

An anomaly is, as earlier, a vector that does not fit well to the model of background data, so we can thus use the pdf as an (negative) anomaly score, i.e.,

$$\begin{aligned} D_{\mathsf{GMM}}(\mathbf{x}|\mathcal{B}) &= [\,p_{\mathsf{GMM}}(\mathbf{x}|C) < t\,] \\ &= \left[\sum_l w_l \; p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\mu}_l,\boldsymbol{\Gamma}_l) < t \right], \end{aligned} \tag{5.4}$$

where the parameters $\{w_l, \boldsymbol{\mu}_l, \boldsymbol{\Gamma}_l\}_{l=1}^{L}$ are estimated using the EM algorithm. The distance measure is illustrated in Figure 5.3; the same data set as in Figure 5.1 is used and the points P1 and P2 are classified correctly.

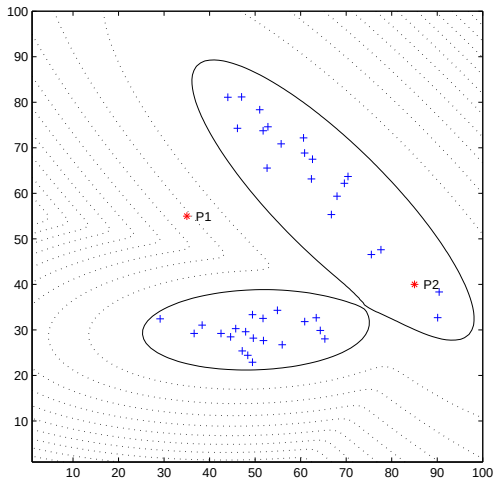


Figure 5.3: Isocurves of the pdf of a Gaussian mixture model estimated using the EM-algorithm. Note the similarity to Figure 5.2 (right).

5.5 Reducing the computation time

Estimating a cluster set or a multimodal pdf from high-dimensional data is a time-consuming process. However, we can subsample the data spectrally and/or spatially prior to the clustering without sacrificing detection performance. The scheme would be as follows:

1. Reduce the size of the data set, e.g., by spatial subsampling.
2. Reduce the spectral dimensionality of the data set, e.g, by PCA or spectral binning.
3. Cluster the reduced data set and return the class labels.
4. Restore the dimensionality of the data set and use the class labels to re-compute the parameters.

5.6 Summary of detectors using cluster and mixture models

Clustering the available image data using LBG or K-means we get spherical clusters. In practice, however, Gaussian clusters and the corresponding GMRX detector have proven more useful. By assuming a Gaussian mixture model, quite similar to the Gaussian cluster model, we get the GMM detector. The detectors are summarized in Table 5.1.

Table 5.1: Detectors using a cluster or mixture background models.

Target model	**Background model**		**Detector**	
None	Spherical clusters	$\{\boldsymbol{\mu}_l\}$		
None	Gaussian clusters	$\{\boldsymbol{\mu}_l, \boldsymbol{\Gamma}_l\}$	GMRX	Eq. 5.2
None	Gaussian mixture	$\{w_l, \boldsymbol{\mu}_l, \boldsymbol{\Gamma}_l\}$	GMM	Eq. 5.4

6. Spatial Modelling

The detection methods discussed above require spatial and spectral models for targets and background. Here, we disregard spatial patterns, and thus the spatial models basically only tell us which pixels to consider as background and potential target, i.e., where to collect data to train our spectral model(s).

To measure a distance from a pixel signature to, for example, the background model, we need to define the spatial area that represent the background, i.e., what pixel signature to chose as training vectors for the model. We define the following areas (illustrated in Figure 6.1):

- The *center pixel* is the pixel we are currently examining.
- The *global background* is the entire available image.
- The *local background* contains all pixels within a distance of n_L pixels from the center pixel. Typically the L1 distance is used, making the neighbourhood square. Pixels within a distance n_G pixels, the *guard distance*, from the center pixel might be excluded from the local background.
- The *Target area* contains the pixels within a certain distance from the center pixel. Each pixel is weighted as to reflect the likelihood of the target stretching to the pixel. Commonly, a Gaussian distribution is used. The resulting value should be normalized so that it sums to 1 over all target area pixels.

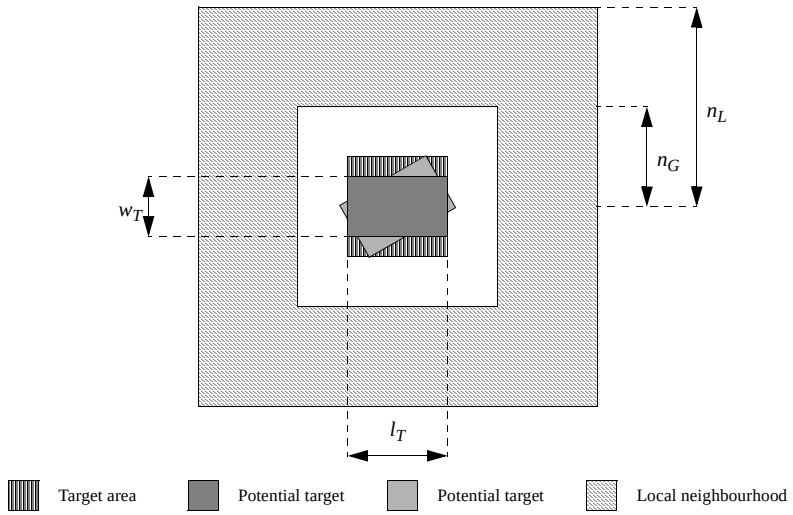


Figure 6.1: Target area, guard distance, and local neighbourhood.

From the global and/or local background we can build the background model $\mathcal{B}$ or even several background models if we perform a clustering of the background. The

target signature $\mathbf{x}$ is estimated as the weighted average of the target area pixels. Given a target probe $\mathcal{T}$ that can be a single signature vector or a representation of a class, we then measure $d(\mathbf{x}, \mathcal{B})$ and $d(\mathbf{x}, \mathcal{T})$, as mentioned above.

A global model is useful when statistics on the entire scene is necessary (for example, end-member extraction) or when the model is advanced enough to handle a complex scene (for example, Gaussian mixture models). Global models also have the advantage that they are not re-trained for each pixel.

Yet another spatial model is to segment the image and use different background models for different parts of the image. This corresponds to the class conditional RX detector described in Section 5.3.

7. Summary and Discussion

In this report, several methods for target detection in multi/hyperspectral imagery have been discussed. Experimental results are accounted for in a separate report. For reference, the use of many of the algorithms can be found in the literature.

7.1 Anomaly detectors

Various anomaly detectors have been described, differing in what kind of model they use for background data. The detectors are summarized below and in Table7.1.

- The most common detector is the **RX** detector modelling the background as a multivariate Gaussian distribution. The model can be applied locally as well as globally, however a global Gaussian model is typically not very accurate and give poor results. If the dimensionality (number of spectral bands) is high, the number of samples needed to estimate the model parameters get very high, prohibiting the use of a small local neighbourhood.
- The problems of RX are solved by using a **Gaussian mixture model** (GMM) as background model. Since the model can handle complex distributions, it can be applied globally without deterioration. In fact, it can be trained quite quickly, and although the traversal of the test pixels is slower than for global RX, it is faster than for local RX. It tends to outperform both.
- The **nearest neighbour** detector tends to be very slow if the spatial model is large since it includes a sorting step. It outperforms RX and is sometimes on par with Gaussian mixture models.
- The **distance from (background) feature space** (DFFS) detector is basically a simplification of RX.
- The **linear mixture model** (LMM) relies on the extraction of end-members, and treat all test pixels not being a linear mixture of these end-members as anomalies.
- The **class conditional RX** or **Gaussian mixture RX** (GMRX) assumes clustering by LBG or EM (or a variation). However, a GMM detector (see above) performs better and is only slightly slower in the training phase. If the dimensionality is high, the peformance is nearly identical since the pdf of a Gaussian mixture model is then often dominated by the closest component anyway.

7.2 Target detectors

Depending on what knowledge we have about the background and target, we select different detectors. Regarding the background, we can model it as a Gaussian distribution or a linear subspace. The target can be modelled as a certain signature vector, a combination of signature vectors, or a Gaussian distribution.

Table 7.1: Summary of anomaly detectors.

Assumption/model for background	**Detector**		**Spatial model**
Gaussian distribution	RX	Eq. 3.11	Global or local
Complex distribution	GMM	Eq. 5.4	Global
Only the closest background samples are worth considering	NN	Eq. 3.28	Local
Linear subspace	DFFS	Eq. 4.9	Global or local
Background samples are mixture of end-members	LMM	Eq. 4.18	Global
Background samples are clustered in spectral space	GMRX	Eq. 5.2	Global

The detectors using Gaussian or subspace background models, all target detectors and two anomaly detectors, are summarized in Table 7.2. The table tells what parameters need to be known about background and targets for using the respective models, and the corresponding detectors are given with name and reference to the defining equation. Thus, for example, if you know (or can estimate) the mean $\boldsymbol{\mu}_B$ and covariance $\boldsymbol{\Gamma}_B$ of the background clutter, and you know the signature $\mathbf{t}$ of the target you want to detect, then Table 7.2 tells that the AMF detector defined in (3.20) should be used.

- If the target signature is known, that is, we search for a specific target signature, and the background/noise is modelled as a Gaussian distribution, we use the **adaptive mathced filter** (AMF). The background model is typically trained from image data. The AMF whitens both test vector and target signature with respect to the background/noise, and computes their scalar product.
- If the target signature is not known, but not the norm of the signature vector (as would be the case for a subpixel target or when the noise level is unknown) we instead use the **spectral angle mapper** (SAM). SAM measures the angle between the target signature and test vector.
- In case the background is modelled as a linear subspace, the **orthogonal subspace projector** (OSP) can be used. OSP matches the component of the test vector that is orthogonal to the background subspace with the target signature.
- In the rare case that so many target training vectors are available that a Gaussian model can be trained for the target, then the **Bayesian classifier** can be used directly.
- Using a subspace model for targets, which is a common way to do when several target signatures are available, we can use the **adaptive cosine detector** (ACE) when modelling the background as Gaussian and the **adaptive subspace detector** (ASD) when modelling the background as a subspace. ACE relates to the angle between the whitened test vector and the whitened target subspace. ASD has different formulations for resolved or subpixel targets.

Table 7.2: Summary of target detectors.

		Background model			
Target model		Gaussian $\boldsymbol{\mu}_B, \boldsymbol{\Gamma}_B$		Subspace $\boldsymbol{\Phi}_B$	
None		RX	Eq. 3.11	DFFS	Eq. 4.9
Known	$\mathbf{t}$	AMF	Eq. 3.20	OSP	Eq. 4.11
Known up to norm	$k\,\mathbf{t}$	SAM	Eq. 3.22	ASD	Eq. 4.11
Gaussian	$\boldsymbol{\mu}_T, \boldsymbol{\Gamma}_T$	Bayes	Eq. 3.15	Not treated, use ACE	
Subspace	$\boldsymbol{\Phi}_T$	ACE	Eq. 3.24		
- subpixel				ASD	Eq. 4.13
- resolved				ASD	Eq. 4.16

A. End-member Extration Using N-FINDR

The N-FINDR algorithm [9] selects $M = N + 1$ pixels in a scene (samples in training data) as end-members, where N is the number of spectral bands. N-FINDR traverses the samples and finds the ones that form the simplex with maximum volume.

The algorithm is as follows:

1. Randomly pick M samples as end-member candidates $\mathbf{e}_1 \dots \mathbf{e}_M$ and create the matrix
$$\mathbf{E}_0 = \begin{bmatrix} 1 & \cdots & 1 \\ \mathbf{e}_1 & \cdots & \mathbf{e}_M \end{bmatrix}. \tag{A.1}$$

2. Compute the simplex volume
$$V(\mathbf{E}_0) = \frac{1}{(N-1)!}|\mathbf{E}_0|. \tag{A.2}$$

3. Let i go from 1 to K (the number of training vectors).

4. Replace, one by one, each of the end-member candidates with $\mathbf{x}_i$, i.e., create the matrices
$$\mathbf{E}_m = \begin{bmatrix} 1 & \cdots & 1 & 1 & 1 & \cdots & 1 \\ \mathbf{e}_1 & \cdots & \mathbf{e}_{m-1} & \mathbf{x}_i & \mathbf{e}_{m+1} & \cdots & \mathbf{e}_M \end{bmatrix}. \tag{A.3}$$
for $m = 1, \dots, M$. Compute the corresponding volumes $V(\mathbf{E}_m)$ and select the simplex with the largest volume, i.e., let
$$\mathbf{E}_0 = \mathbf{E}_{m^\star}, \tag{A.4}$$
where
$$m^\star = \arg \max_{m=0,\dots,M} V(\mathbf{E}_m). \tag{A.5}$$

B. Clustering Methods

There is a number of related algorithms for clustering. Many variants comply to the following scheme (unsupervised clustering):

1. Assume K training vectors $\mathbf{x}_k$. Initialize L classes $\mathbf{C}_l$.
2. For each vector $\mathbf{x}_k$, measure the distance

$$d_{kl} = d(\mathbf{x}_k, \mathcal{C}_l) \tag{B.1}$$

   and compute the membership value

$$m_{kl} = f_m(d_{kl}). \tag{B.2}$$

3. Recompute the classes

$$\mathcal{C}_l = f_c(\{m_{kl}\}, \{\mathbf{x}_k\}) \tag{B.3}$$

4. Repeat steps 2 and 3 until convergence.

By defining different distance functions $d(\cdot)$, membership functions $f_m(\cdot)$, and class representations/functions $f_c(\cdot)$ we get different algorithms. A few examples are given here; *Linde-Buzo-Gray* (LBG) or K-means and three variations of *Expectation-Maximization* (EM, CEM, and SEM). The LBG algorithms uses the squared Euclidean distance and assigns a discrete membership value to each vector. The EM algorithm uses the Mahalanobis distance and adapts a Gaussian mixture model to the training data. CEM is similar to LBG, but uses the Mahalanobis distance. CEM is also similar to EM, but assigns discrete class labels to the training samples.

The formal definitions follow.

B.1 Linde-Buzo-Gray

For the LBG algorithm, each class is represented by the mean vector, i.e.,

$$\mathcal{C}_l =< \boldsymbol{\mu}_l > . \tag{B.4}$$

The distance is the squared Euclidian distance

$$d(\mathbf{x}, \mathcal{C}_l) = \|\mathbf{x} - \boldsymbol{\mu}_l\|^2, \tag{B.5}$$

and the membership function is

$$f_m(d_{kl}) = \begin{cases} 1 & \text{if } d_{kl} = \min_l d_{kl} \\ 0 & \text{otherwise.} \end{cases} \tag{B.6}$$

The class update function is simply

$$\boldsymbol{\mu}_l = \frac{1}{|\mathcal{C}_l|} \sum_k m_{kl} \mathbf{x}_k, \tag{B.7}$$

and the final discrimination/classification function is

$$f_d(\mathbf{x}) = \arg\min_l d(\mathbf{x}, \mathcal{C}_l). \tag{B.8}$$

To measure the quality of the final clustering, the function

$$f_q(\{\mathbf{x}_k\}, \{\mathcal{C}_l\}) = \sum_k d_{kl} m_{kl}) \tag{B.9}$$

can be evaluated.

B.2 Expectation-Maximization

The EM-algorithm is an algorithm for estimating the parameters for a Gaussian mixture model with a pdf

$$p_{\mathsf{GMM}}(\mathbf{x}|\mathcal{M}) = \sum_l w_l \; p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\mu}_l, \boldsymbol{\Gamma}_l), \tag{B.10}$$

where the mixture model $\mathcal{M} = \{\mathcal{C}_l\}_{l=1}^L$ is a set of L components, where each component $\mathcal{C}_l$ is represented by a prototype vector, a covariance matrix, and a weight,

$$\mathcal{C}_l =< w_l, \boldsymbol{\mu}_l, \boldsymbol{\Gamma}_l > . \tag{B.11}$$

There are several variations, of which a few are mentioned below. They all aim to optimise the summed log-likelihood function of the pdf over the training data, i.e.,

$$f_q(\{\mathbf{x}_k\}, \{\mathcal{C}_l\}) = \sum_k -\log p_{\mathsf{GMM}}(\mathbf{x}_k)) \tag{B.12}$$

- **Expectation**. Compute the weighted pdf and the membership function for each training sample with respet to each component

$$p_{kl} = w_l \; p_{\mathcal{N}}(\mathbf{x}_k|\boldsymbol{\mu}_l, \boldsymbol{\Gamma}_l) \tag{B.13}$$

$$m_{kl} = \frac{p_{kl}}{\sum_l p_{kl}}. \tag{B.14}$$

- **Maximization**. Compute the new model parameters, i.e., the class function

$$w_l = \frac{\sum_k p_{kl}}{K} \tag{B.15a}$$

$$\boldsymbol{\mu}_l = \frac{1}{K} \sum_k m_{kl} \mathbf{x}_k \tag{B.15b}$$

$$\boldsymbol{\Gamma}_l = \frac{1}{K-1} \sum_k m_{kl} (\mathbf{x}_k - \boldsymbol{\mu}_l)(\mathbf{x}_k - \boldsymbol{\mu}_l)^T. \tag{B.15c}$$

B.3 Classification Expectation-Maximization

A somewhat simplified variation is Classification Expectation-Maximization (CEM). It is basically the same as LBG, but with Gaussian clusters.

- **Expectation**. As above.

- **Classification**. Assign a label to each training sample, i.e.,

$$c_k = \arg\min_l p_{kl} \tag{B.16}$$

$$m_{kl} = \begin{cases} 1 \text{ if } c_k = l \\ 0 \text{ otherwise} \end{cases} \tag{B.17}$$

- **Maximization**. Compute the new model parameters

$$w_l = \frac{\sum_k p_{kl}}{K} \tag{B.18a}$$

$$\boldsymbol{\mu}_l = \frac{1}{|\mathcal{C}_l|} \sum_k m_{kl} \mathbf{x}_k \tag{B.18b}$$

$$\boldsymbol{\Gamma}_l = \frac{1}{|\mathcal{C}_l| - 1} \sum_k m_{kl} (\mathbf{x}_k - \boldsymbol{\mu}_l)(\mathbf{x}_k - \boldsymbol{\mu}_l)^T, \tag{B.18c}$$

B.4 Stochastic Expectation-Maximization

Stochastic Expectation-Maximization (SEM) is very similar to EM, but uses the computed probabilities for a stochastic classification of the training samples. SEM is less likely to get stuck in local minima than EM, and tends to converge somewhat faster.

- **Excpectation**. As above.
- **Stochastic classification**. Stochastically assign a label to each training sample

$$c_k = l \text{ with the probability } m_{kl}, \tag{B.19}$$

  and recompute the membership function as

$$m_{kl} = \begin{cases} 1 & \text{if } c_k = l \\ 0 & \text{otherwise} \end{cases} \tag{B.20}$$

- **Maximization**. As in CEM.

Bibliography

[1] A. A. Green, M. Berman, P. Swixter, and M. D. Craig. A transformation for ordering multispectral data in terms of image quality with implication for noise removeal. *IEEE Transactions on Geoscience and Remote Sensing*, 26(1):65–74, 1988.

[2] J. C. Harsanyi and C. I. Chang. Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach. *IEEE Transactions on Geoscience and Remote Sensing*, 32(4):779–785, 1994.

[3] H. Hotelling. New light on the correltation coefficient and its transforms. with discussion. *Journal of the Royal Statistical Society*, 15:193–232, 1953.

[4] J.B.Lee, A.S.Woodyatt, and M.Berman. Enhancement of high spectral resolution remote-sensing data by a noise-adjusted principal components transform. *IEEE Transactions on Geoscience and Remote Sensing*, 28(3):295–304, 1990.

[5] J. Karlholm, M. Ulvklo, S. Nyberg, A. Lauberts, and A. Linderhed. A survey of methods for detection of extended ground targets in EO/IR imagery. Technical Report FOI-R–0892–SE, Swedish Defence Research Agency, 2003.

[6] S. Kraut, L. L. Scharf, and L. T. McWorther. Adaptive subspace detectors. *IEEE Transactions on Signal Processing*, 49(1):1–16, 2001.

[7] J. G. Proakis. *Digital Communications*. McGraw-Hill, Singapore, 1995.

[8] I. S. Reed and X. Yu. Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38:1760–1770, 1990.

[9] M. E. Winter. N-FINDR: an algorithm for fast autonomous spectral end-member determination in hyperspectral data. In *SPIE*, volume 3753 (Imaging Spectrometry V), 1999.