Institutionen för systemteknik

Department of Electrical Engineering

Examensarbete

Masters Thesis

Detection of Man-made Objects in Satellite Images

Per-Erik Forssén

LiTH-ISY-EX-1852

17 December 1997

Abstract

In this report, the principles of man-made object detection in satellite images is investigated. An overview of terminology and of how the detection problem is usually solved today is given. A three level system to solve the detection problem is proposed. The main branches of this system handle road, and city detection respectively. To achieve data source flexibility, the *Logical Sensor* notion is used to model the low level system components. Three *Logical Sensors* have been implemented and tested on Landsat TM and SPOT XS scenes. These are: BDT (Background Discriminant Transformation) to construct a man-made object property field; Local-orientation for texture estimation and road tracking; Texture estimation using *local variance* and *variance of local orientation*. A gradient magnitude measure for road seed generation has also been tested.

Contents

Acknowledgements

The problem addressed in this thesis was far from a recipe of how the work should proceed. Instead the problem definition consisted of a lot of unanswered questions and *some* direct instructions. This has meant that the work has progressed with quite big fluctuations. Sometimes everything has worked, and sometimes I have been totally stuck. However, I have a feeling that I have had much more fun doing it this way. For assistance and creative input during this thesis work, especially when I was stuck, I would like to thank the following persons:

My supervisor at SSC Satellitbild, Sören Molander for providing me with the opportunity to conduct this work, and for lots of suggestions and help. Not least for help in finding reference literature and existing algorithm implementations.

The people at the Computer Vision lab at ISY for letting me use their equipment and for showing me how to use it. Thank you Johan Wiklund and Mats Andersson.

My examiner Klas Nordberg for sorting out various theoretical and practical problems.

My opponent Raoul Dahlin for his comments on the contents and for suggestions concerning the layout.

And last but not least my friends Michail Ilias, Martin Eneling, and Håkan Olsson for discussing Computer Vision with me, and for taking the time to read this report and find some of the errors.

Linköping, December 1997

Per-Erik Forssén

Chapter 1

Introduction

This chapter contains information that is meant to simplify the digestion of this final report. After an initial document overview the reader is introduced to the topics *Computer Vision*, *Remote Sensing*, and *Logical sensors*. This is followed by a description of sensors operating within the electromagnetic spectrum. At the end of this chapter the actual problem that has been addressed during this thesis work is presented.

1.1 Overview

The essence of each section will now be described in a few words. If you are only interested in a small part of this work, this is where you find out where to look.

prerequisites

The level of detail in this thesis is adapted for readers with a Master of Science background. For this reason introductions to the topics *Computer Vision*, *Remote Sensing*, and *Logical Sensors* are included, as these are central in this work, but not “common knowledge” among the target group of readers. Familiarity with matrix algebra, calculus and elementary image processing is assumed.

presenting the context

After the introductory sections the possibilities and limitations associated with the sensors used in remote sensing are described in the section *Electromagnetic Sensors*. We are now ready to address the actual problem. The problem studied in this thesis is defined in the section *Problem Definition*.

solving the problem

The next chapter, *Detection Systems*, serve as an introduction to the detection problem. It describes the different ways this problem is usually solved.

system design

In the chapter *An Attempt at Man-made Object Detection* an overall design of a detection system is presented. This chapter contains the main results of this thesis work.

conclusions

The thesis ends with a general discussion of the results in the chapter *Discussion*. Finally the results; those things that actually originate from this work are summarized in the chapter *Summary*.

appendices

The appendices contain various pieces of information that was collected during this thesis work. This information was not considered crucial to the understanding of the results, and extracted to make the presentation more concise. The appendices are included in the report anyway in the hope that someone will find them useful.

The appendix *Preprocessing of satellite images* explains the prerequisites of the detection system. The appendix *Classification* describes the principles of classification to those who are unfamiliar with them.The structure of the AI systems known as Blackboard systems is explained in the appendix *Blackboard Systems*. The appendix *Earth monitoring satellites* lists data on the main Earth monitoring satellites used for cartography today. It is data from these satellites that is meant to be used by the detection system.

1.2 Computer Vision

The field of Computer Vision deals with extraction and interpretation of features and objects in images, image sequences, and similar data-sets.

scenes

In the following all Computer Vision data-sources are termed *scenes*. The name Computer Vision stems from the striking similarities in the tasks that computer-vision systems carry out and those that vision-systems in humans and other animals handle. In the field of Computer Vision, scene interpretation is usually categorized in three levels:

low-level analysis

1. Low-level analysis. This incorporates detection of edges, orientation, motion, and colour as well as methods for noise attenuation.

mid-level analysis

2. Mid-level analysis. Extraction of line-segments, segmentation of a scene, clustering etc.

high-level analysis

3. High-level analysis. This level of analysis is closely related to the field of *Artificial Intelligence* (AI). High-level analysis usually consists of models of the domain, often represented as *frames* (a kind of form with fields that are filled in as information becomes available), probabilistic networks, and rule based systems. The models in high-level analysis are categorised as either declarative (D) or procedural (P). D-models describe passive properties, such as numerical thresholds, while the P-models handle active decisions and strategies of interpretation (such as scheduling; what should be done and when). Tagging of objects (here, objects are things in a scene that we want to interpret) in the scene usually means an intensive flow of bottom-up (pixels and upwards) and top-down information (from interpretation models to pixels).

data flow

To achieve an effective interpretation of a scene it is important to accomplish models in the high level that as far as possible is free from details from the lower levels. This is desired in order to avoid the bottlenecks in performance that the increased flow of data will cause. One obstacle here is that most low-level algorithms depend on some sort of parameters, usually empirically derived thresholds. The model thus has to be able to choose “good” parameters, to avoid undesired iteration between the levels.

interpretation profiles

Preferably the model (as well as the operator of a Computer Vision system) should not need to know about numerical thresholds, but instead be able to provide *interpretation profiles* in a more abstract, high-level manner. This is important if we want to incorporate a human in the decision-chain, without having this person knowing detail information about all levels in the system.

1.3 Remote sensing

Remote sensing is a broad research field with a wide range of applications. Technically the term “remote sensing” means *acquisition of information without being in direct contact with the object that is studied* [20]. This will typically imply detection of some kind of radiation. The detected radiation is either emanating from the object itself or is reflected by it.

detection

Most people have performed the first variant of remote sensing (detection of radiation) themselves, for instance when looking at an object or when taking a photograph. Technical applications (this is where the term is actually used) are for example aerial photography and satellite monitoring. This thesis will concern the latter.

emission and detection

The second variant occurs not quite so often in every-day life, but for example bats and dolphins use it when they emit soundwaves and detect the echoes. More recent satellites also use this variant in SAR (Synthetic Aperture Radar) monitoring. Here pulses of radio wavelength are emitted by the satellite, and the emitted waveform is correlated with the echo to extract information about target motion. (Details on SAR can be found in the book “Remote Sensing and Image Interpretation” [20].)

remote sensing today

Remote sensing via satellites is today a field in rapid development with a wide spectrum of applications. New sensors with new wavelength ranges and improved resolutions constantly alter the demands on interpretation and analysis software. At present, the methods for extraction of information from remotely sensed images are relatively old-fashioned: Most of the work with classification and image-interpretation is performed either in a semi-automated fashion on digital images or manually on photographic reproductions. One exception is extraction of 3-D information, mainly from aerial photographs. This is at present a thriving research-field where some progress has been made. With the increasing supply of new sensors and applications it is obvious that an increased level of automation in image-interpretation is required.

1.4 Logical sensors

A logical sensor is by definition *either a physical sensor or a process which is only constrained by its I/O* [3]. This rather abstract description depicts a logical sensor as a black box (left part of Figure 1 below) with some kind of input (either physical sensory input or input from another logical sensor) invisible to the user of the sensor, and an output which is a synthesis of sensory data, descriptions of this data and descriptions of what kind of information is sought.

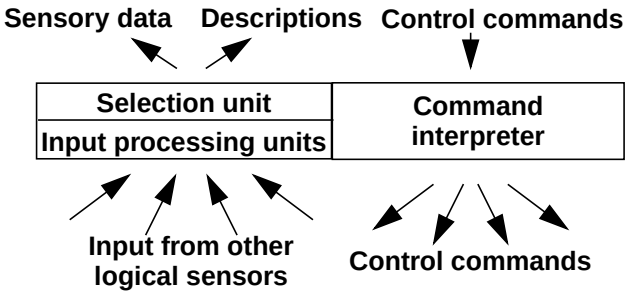

FIGURE 1. Logical sensor scheme

Adapted from [13].

purpose

The purpose of logical sensors is to serve as a means of achieving *data abstraction* and *modularity* [12]. As much of the low-level information as possible is hidden from the higher vision processes. Originally logical sensors were meant to enable multisensor-systems to cope with the loss of one or more sensors when the system contained other sensors that were functionally ***equivalent with respect to the data produced***.

descriptions of sensory data

Logical sensors are entities that integrate sensory data with descriptions of the data-source upon interpretation. The description of the source can speak of things like:

- what kind of data is being provided (wavelengths, resolutions etc.)
- shortcomings of the providing sensor
- the biasing introduced by the sensor.

In fact there is no real difference between these three categories, they can all be referred to as *descriptions of sensory data*. By providing an input that is a combination of data and descriptions that the sensor can understand we have provided the sensor with *information* (this is not to be confused with the term information used in information theory) instead of just data. As logical sensors produce output to other logical sensors, this type of information is both supplied to the sensor and produced by it.

The descriptions of sensory data can also be seen as a means of adapting the system's *a priori* (beforehand assumptions) information about the scenes.

command interpreter

To improve the adaptivity of a logical sensor a *command interpreter* (right part of Figure 1 on page 8) is added [13]. If the higher levels notice changes in the input domain they may send control commands to the lower levels notifying a change in the interpretation approach.

The logical sensor notion accomplishes three things:

- It separates the sensor functionality from the actual implementation. Functionality is described in a more high-level manner.
- It provides a standardised means of communication between the components of an interpretation-system.
- It focuses on increased "understanding" of the input within the data-interpreting system. This is a necessity when ***integrating data from different sources***. It should be noted that this is a mere side effect and not an aspect of the design of the scheme.

logical sensors for remote sensing

One of the big problems in the field of remote sensing today is integration and generalisation of the body of knowledge on geographical scene interpretation. Applications of almost identical methods for problem solving (classification, segmentation, tracking etc.) can derive completely different conclusions depending on prerequisites and background of the authors of the algorithms. Thus there is a very real need of interpretation models that consider both the actual data *and* the purpose of the analysis.

sensor and algorithm selection

The applicability of logical sensors in remote sensing is partly due to the abstraction and modularity aspects of the scheme. However, their use is also motivated by the ability to make selections in a flora of algorithms and available sensory data. In other words, we can provide our system with “too much” information, and it will still be able to work reasonably fast. Véronique Clément et al. [3] have successfully made use of logical sensors in a Computer Vision cartography system this way.

reduction of information flow

Yet another reason for investigating the logical sensor approach is that the scheme promises a much needed reduction of the information flow between different conceptual levels in a remote sensing system. The scheme suggests that some, very specific kinds of reasoning, should be embedded in the lower-levels, as they need to consider descriptions of sensory data.

1.5 Electromagnetic sensors

All remote-sensing cartography performed by Earth monitoring satellites is based on sensors that detect electromagnetic radiation. Theoretically there is an infinite range of different wavelengths of electromagnetic radiation to detect, but satellites monitoring the surface are limited to those wavelengths that are relatively free of atmospheric absorption (Figure 2 below).

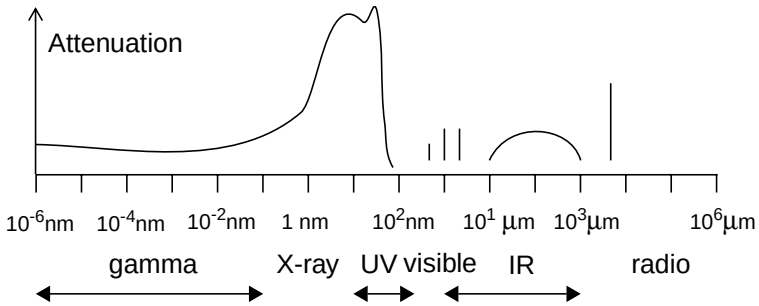

FIGURE 2. Atmospheric absorption

This is just an outline, it is adapted from [18].

The main atmospheric windows are the visible and the radio windows [18]. Satellites commonly also exploit the near infrared (the leftmost part of the IR-band) range as many objects show distinct features in these wavelengths. These bands are termed the *optical spectrum*, and the *radio spectrum* in remote sensing.

the optical spectrum The optical spectrum (0.3 to 14 µm) includes UV and IR wavelengths, the name stems from the usage of ordinary mirrors and lenses to reflect and refract the radiation.

the radio spectrum The radio spectrum (2 mm to 60 m) is the range in which radar-equipment operate.

In practice, all radiation that electromagnetic sensors detect originally stems from the sun [23]. (The sun is our dominant source of energy, and radiation is a form of energy.) When sensors look at the ground, they see a combination of radiation emitted by objects on the ground and reflected radiation from the sun.

1.5.1 Spectral signatures

One important way of discriminating objects in a remotely sensed scene is by means of examining their *spectral signatures*. We shall now describe this important concept.

emittance

All objects (with a temperature above 0 K) emit radiation [18]. This radiation is termed *emittance* and is distributed across the electromagnetic spectrum according to the temperature of the emitting object.

reflectance

The *reflectance* of an object on the other hand is radiation (normally originating from the sun) that has been directly reflected by the object. The sun emits most of its radiation in the range 0.2 to 3.4 μm. This further limits the range of the optical spectrum that is of interest.

spectral signature

The sensors on Earth monitoring satellites see a combination of these two categories. Each material has its own *spectral signature* constituting of the spectral distributions of its emittance and reflectance (see Figure 3 on page 13). The combination of these two spectra gives an exhaustive description of the radiation that we are able to detect from an object.

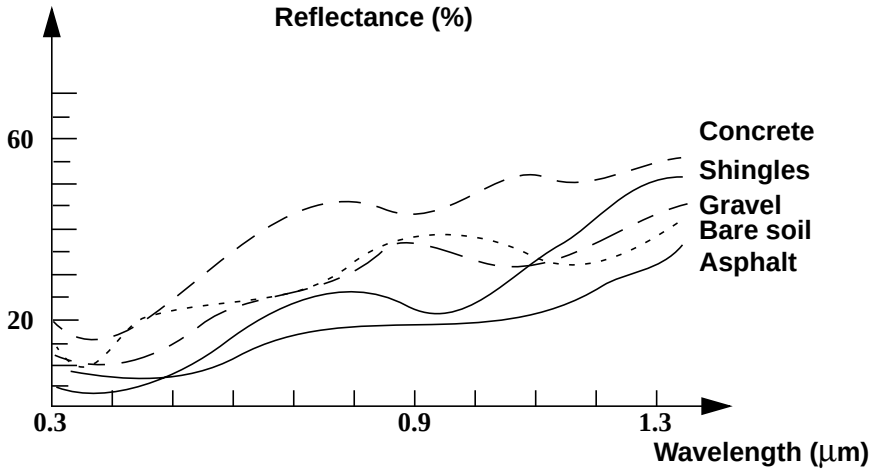

FIGURE 3. Spectral signatures

Spectral signatures of some artificial materials. Adapted from [23].

In most parts of the spectrum the reflectance is dominating and thus most remote sensing systems are designed to monitor reflected radiation. An exception is the thermal IR bands, where the emittance is stronger.

1.5.2 Complications

Unfortunately there are several obstacles that complicate the discrimination process (that is, the process of distinguishing an object in a scene). The emittance spectra for example depend on object temperatures, and the reflectance spectra depend on the amount of solar radiation that strike the objects. Another problem is that the absorption spectrum of the atmosphere (Figure 2 on page 11) is far from constant. For example when the sky is clouded, or foggy, or when it rains, the characteristics change radically. Discrimination of objects near high trees and mountains are further complicated due to shadows cast upon them, causing them to look different at different solar angles.

1.5.3 Sensor fusion

Earth monitoring satellites act as sensors when they detect electromagnetic radiation. Sensors detect *features of objects*. (To use this broad definition we must note that one "feature" of an object is the mere presence of it!)

wavelength bands

Electromagnetic sensors are designed to detect only radiation in a limited wavelength-range (a *band*). The reason for this is that the emitted energy within a narrow band tell us more about the reflectance of an object than an average over a wide band. When the satellite image is received and processed on the ground, bands from several sensors may be combined. This will generally simplify the interpretation of a satellite-scene, as some object features stand out in one band while other features are spotted in another band. The combination of information from several sensors is usually termed *sensor fusion*. (In general, sensor fusion need not only concern electromagnetic sensors.)

human vision system comparison

The combination (or fusion) of several bands is similar to the approach used by the *human vision system* to create colours [10]. The main purpose of colour perception is believed to be increased ability to discriminate objects by means of observing how they reflect light of different wavelengths. The concept of colour is, to be strict, not *one* feature of an object, it is more accurately described as *three* features. There are three kinds of cones (a kind of light sensitive cells) in our eyes, sensitive to three different wavelength-bands in the visible range, labelled red, green, and blue. The input from these bands are combined by the brain and we perceive them as a colour. In other words, the colour of an object is the combined perception of the light an object reflects in these three bands.

generalization

Satellite systems differ from the human vision system in that each satellite has its own set of sensors, constituting a unique set of bands and will thus require interpreting software specially adapted for it. One of the goals of this thesis is to generalize feature detection in satellite images so that *one* detection algorithm in *one* application can handle *all* satellites. To accomplish this the principle of logical sensors is investigated.

1.6 Problem definition

Below is the original definition of the problem addressed in this thesis work.

> The goal for this thesis is to develop models of logical sensors that use information from sensors (radar or optical data), and *a priori* data such as vector-fields in geographical databases in order to detect *man-made objects* (that is artificial objects in a scene, primarily cities, buildings, and roads). The work will comprise studies of papers and articles on related logical sensor systems and on road and city detection algorithms. Implementation of one or more algorithms adapted for SPOT-images (or preferably more generic), to be used as reference in system modelling. The work also includes studies of literature on existing systems for automated image-interpretation in the field of remote sensing and image-interpretation dealing with detection of roads and cities.
>
> A systematic line-up of the logical flows of information that is required in the detection process should be given. For example: Is it possible to make the sensors cooperate, or will one sensor suffice? If the sensors can indeed cooperate, how should the results be merged? Which parameters are crucial to detection of for example roads (threshold values)?
>
> For a given scene and the goal to find cities and roads, make a “simulated” interpretation of a number of images, in order to theoretically test different models for algorithm and method selection in the logical sensors. Tests on the actually implemented algorithm will also be important here. Give suggestions to how a human operator should be able to aid the interpretation process at a suitable level.
>
> If there is time left, suggest methods and algorithms for scheduling and high-level (object oriented expert-systems etc.) systems for strategy selection if one or more sensors are out of order. For example: If cities are most easily detected in SAR-images, how should the strategy be altered if only SPOT XS or Landsat (resampled to equivalent resolution, i.e. SPOT XS 20 m) data were available?

Chapter 2

Detection Systems

This chapter describes the principles that govern the design of detection systems in remote sensing. It starts with a general discussion of what we *can* detect from Earth orbit, and what we *want* to detect. It also contains an overview of detection system components, and a section dedicated to the road extraction problem.

2.1 Properties of objects

It is now time to discuss which properties of objects on the ground that can actually be sensed by satellite. There are two main ways to organize properties, either by focusing on the objects we want to detect or by focusing on the available sensors. The object property approach is useful for describing our knowledge about the objects we want to detect, while the sensor property approach is useful when studying the input to a detection system. The aim is to convert sensor properties into object properties.

2.1.1 The object property approach

Most people know what roads and cities are, but describing these entities in terms that still stand valid in satellite images and in terms that can be understood by a logical sensor is not all that trivial. If we want to construct a high-level description of the objects we want to detect (which is what we want, see “Computer Vision” on page 4) we must first delve into the fundamentals of sensoring principles in order to determine which properties are useful at all on the high level.

a priori knowledge

Véronique Clément et al. [3] have suggested three sets of descriptors for objects; *geometric*, *radiometric*, and *spatial context*. These constitute the *a priori* knowledge (i.e. the inherent knowledge of the system before it has “learned” anything by itself) of the detection system.

geometric descriptors

Geometric descriptors describe properties that relate to the shape of objects i.e. square, rectangular, circular, elongated, compact etc. and to their physical size.

radiometric descriptors

Radiometric descriptors give coarse descriptions of the emission-spectrum of an object. This will tell the logical sensor in which intensity-level range a feature may be found in the available bands. This set also includes textural properties such as raggedness and smoothness (or *homogeneity*).

spatial context descriptors

Spatial context descriptors concern spatial relationships between objects. For example spatial context can suggest joining of detected road-segments that have adjacent end-points, or suggest that a strike of “noise” in a detected river might be a bridge, if there are detected road-segments ending on both sides.

Of these the geometric and the radiometric properties are those that pertain to one logical sensor only, while the spatial context concern how logical sensors may cooperate in the scene interpretation.

2.1.2 The sensor property approach

On the lowest level of a logical sensor system the input must be described in terms of the properties of the sensors onboard the satellites. Satellite sensors have three main properties of interest, namely *spatial resolution*, *frequency range*, and *intensity resolution*.

spatial resolution

Earth monitoring satellites have a wide range of spatial resolutions ranging from 200 square meters to sub-meter resolutions (in espionage-satellites). A first thought on this might be that the higher resolution, the better, but this is not altogether true. In a number of applications, such as mapping of entire nations, it is actually more favourable to use low resolution satellite images, as a higher resolution implies that the image will cover a smaller area (due to limited resolution in the detector-elements and limitations on the rates at which the data can be transmitted to Earth), and the nation map would have to be created as a mosaic of a large number of small images. The higher resolution is

not of any use here either, as you cannot possibly see meter-sized objects on for example a map of Sweden. This, and the fact that you usually pay per image, not per square meter covered makes low-resolution satellites interesting.

The spatial resolution of the sensor is crucial information for the detection process. An image is of little use if the feature you wish to detect is smaller than the resolution of the sensor.

wavelength range

The wavelength range of an image tells the analysing component which kinds of objects it might expect to discriminate in the image. For example a blue band can discriminate water, a green band vegetation and so on. (For wavelengths outside the visual spectrum the applications are not quite as obvious.) In the field of manual image-interpretation an extensive body of knowledge on these matters has been gathered. It would indeed be nice if we could somehow incorporate this knowledge into a logical sensor model.

A crucial issue in sensor selection is whether the wavelength band of the sensor you wish to employ actually discriminates the feature you wish to detect.

intensity resolution

The intensity resolution of a sensor tells us how crude the detection is. A high intensity resolution will ease the discrimination process. However, most sensors today are 8 bit sensors (yielding 256 intensity levels).

A too low intensity resolution will complicate the feature extraction process, as two similar, yet different features might yield the same intensity value.

2.2 Principles of man-made object detection

The two sets of features of interest in man-made object detection on the logical sensor level are *radiometric* and *geometrical* properties (See “The object property approach” on page 18.). This section will describe the principles of how these features are normally used to discriminate two classes of objects, namely roads and cities from the rest of a remotely sensed scene.

2.2.1 Radiometric properties

Each material has a unique reflectance spectrum (under the constraint of a white light source). Therefore an object is more easily detected if we know what kinds of materials it may consist of. Unfortunately most radiometric (or *spectral*) properties of objects are not invariant to the time of detection. Different materials have different radiation profiles at different times of the day, in different seasons and during different weather conditions. There are two extreme solutions to this anomaly:

- The first alternative is to feed the sensor with *all* the conditions at the time of detection that are *equivariant* to (that is, varies with) the features we want to detect. This alternative would by far yield the best interpretation possibilities. However it is difficult to implement due to difficulties in obtaining the required descriptions of the sensory data.
- The second alternative is to discard all features that are not invariant to weather, solar angle, temperature etc. and concentrate on those that stay constant. This will leave us with a small amount of descriptions of the sensory data, and thus less information to extract from the data. If this approach works however, it is much easier to implement.

2.2.2 Handling of uncertain information

Most approaches to the detection problem fall somewhere in between the approaches mentioned above; they use a few of the equivariant features, and supply the sensor with rough information such as:

- exposure season
- time of day
- solar angle.

fuzzy logic

With this approach, the radiation profiles of the objects may be used in rough ways. For example asphalt-roads may be described as *hot in summer when the sun shines*. These kinds of descriptions are however generally hard to handle as they are *vague* or *fuzzy*. Systems using this kind of feature descriptions often use *fuzzy logic* theory to model them. (See Bart Kosko's book [16] for an excellent introduction to Fuzzy logic.) A few approaches to the detection problem have been made with this kind of descriptions, and they have been reportedly successful [3].

fuzzy sets

The real problem here is to translate the descriptions into *fuzzy sets* (curves determining to what degree the sensory information is accepted as fitting the descriptions). To build the fuzzy sets you usually collect a large amount of data from previously classified satellite images, or measure the features in the field. Another alternative that is becoming increasingly popular is to incorporate an *artificial neural net* in the system and give guidelines on how the system should learn the sets by itself.

2.2.3 Elimination of solar influence

Another way of obtaining measures that are invariant to a given disturbance is to use several properties which are equivariant to this disturbance in the same way. For example we could use this approach to construct a property that describes the presence of chlorophyll: All green plants and trees share a rough pattern in the reflectance function

which artificial objects lack [28]. If we define our chlorophyll presence property as the angular distance between a given property vector and a chlorophyll prototype, we will eliminate the equivariance from the solar light source (assuming the sun always has the same colour). See Figure 4 below.

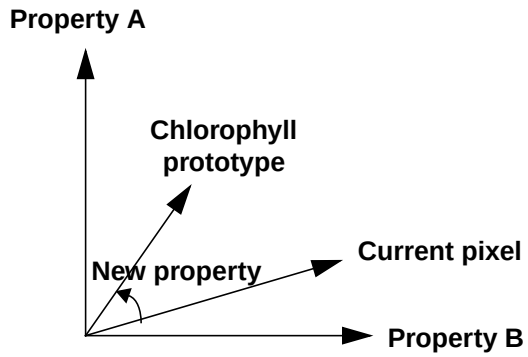

FIGURE 4. Elimination of solar influence

Construction of a chlorophyll presence property. This property will be invariant to solar intensity variations, as the variations will scale the properties A and B by the same amounts. The angles of each pixel's property vector will thus not be affected by the lighting conditions.

Another quite distinct reflectance pattern is that of iron-oxide. This is interesting, as many man-made objects (such as roofs, street-lamps etc.) contain iron-oxide. By creating prototype curves for man-made materials and for natural materials we may classify the scene pixels as either artificial or natural — a good starting point for man-made object detection [28]. More details about this will follow in the implementation sections, where we make practical use of it.

2.2.4 Geometrical properties

The next set of features of interest are geometrical properties. These are fortunately not prone to being equivariant to the exposure conditions. In order to extract geometrical properties we must broaden the focus of attention of the sensor and look at several pixels at a time. This is accomplished by filter sets that create new properties for scene

pixels from known properties in a surrounding region. There exists a large number of filters that extract features such as: *elongation* (linedness), *edgedness*, *orthogonality*, *orientation*, and *width* (frequency).

object shape measures

Included in the geometrical properties are also *object shape measures*. These are by necessity computed after segmentation. If we have a set of pixels that can be said to belong to an object we may compute shape measures based on their spatial distribution. Typical shape measures are *elongation*, *curvature*, *size*, and *compactness*.

Note that at least some of these measures are also available on the filter set level. The difference is that now each *object* gets one measure, on the filter set level each *pixel* gets one measure.

2.2.5 Leaving the things-in-themselves

We are now ready to describe the objects we want to detect in terms that a logical sensor can be made to "understand". To use the words of Kant: We are ready to define the *appearance of the objects* with respect to our system [29].

roads

A road is an elongated structure with long runs of homogeneous width. Roads intersect and fork, and they usually end in bridges or urban areas. Roads also have homogeneous texture and edges that are parallel most of the time. Due to the quality of the sensory information some of these properties may not always be detectable.

urban areas

Urban areas contain a large amount of parallel and orthogonal lines and edges. Urban areas also commonly contain roads and have roads leading to them. Usually urban areas are not elongated structures.

These descriptions constitute our *a priori* knowledge of the objects we want to detect. They should somehow be translated by the logical sensors into mathematical and morphological descriptions that suggest which kinds of filters should be used for detection.

2.2.6 Detection of Objects

When we have somehow obtained filters, and other algorithms that generate property descriptors for the properties mentioned above, it is time to actually find objects in the scene. This is accomplished by clustering spatially related pixels that have many properties in common. The most common approach to this task is to use *classification*. (See Appendix B. “Classification” on page 75.)

spatial relationships

Classification will not inherently consider spatial relationships: Unless we supply the classification machine with properties that describe spatial relationships we will end up with “cities” of one pixel size in the middle of forests and so on. One (partial) cure for this is to low-pass filter some of the properties and use the original property as a measure of certainty. (See Appendix B. “Classification” on page 75 for details.)

It should now be obvious that classification is far from the last operation in the detection chain. Classification is however the bridge between pixel manipulation and object manipulation.

perceptual grouping

One way of considering spatial relationships without rule based AI is *Perceptual Grouping* [2]. This method is devised by Laurent Alquier et al. and is based on psychological models of how humans interpret scenes and on active contour functions. The idea is aimed at joining line segments, or to be more specific, to detect roads in satellite scenes.

After the pixels have been clustered into objects, these objects are assessed with respect to curvature and co-circularity. Adjacent objects with matching spatial properties are then grouped.

trimming of edges

There exist several methods to refine the object classifications once we have a coarse cluster that we know is correctly classified. For curve-like structures such as roads these are called *Snakes*, and for surfaces they are called *Velcro Surfaces* [25]. Both classes of algorithms use an energy function (typically constructed from some of the object properties) that have basins in the structures we want to detect. The idea is then to minimize the energy for each object.

2.2.7 Reasoning about Objects

Even further up the detection chain one starts to consider spatial relationships between the detected objects. (See “The object property approach” on page 18.) This kind of classification refining will usually consist of some kind of AI-system that infer knowledge about the objects using some kind of grammar. However, implementation of high-level vision algorithms are out of the scope of this thesis work.

2.3 Road Extraction

Roads constitute a difficult class of pixels to discern. Even in high-resolution satellite scenes, roads are usually not more than one or a few pixels wide. The shadows cast by nearby trees, and hills make the detection even harder. For these reasons, most road extraction algorithms are highly specialized.

specialized algorithms

The road extraction problem has been investigated by an almost countless number of people. However, most of the existing algorithms today are specialized at detecting roads in scenes of a specific resolution. This is the opposite of what we want to accomplish with the logical sensor notion.

detection and tracking

Most of the existing road-tracking systems work by combining a local road-tracking algorithm with a global algorithm to find out where to start and stop the tracking. These two stages are usually called *road-finding* (or road-seed generation) and *road-tracking*.

automation

The road-finding phase is seldom automated. Often detection is manual (the operator tells the tracker where the road starts and stops), or semi-automated (the system gives a suggestion to the operator). Some attempts at automation have been made, for example Frédéric Leymarie et al. [19] have studied semi-automatic systems in order to eliminate the need of an operator.

generic systems

One of the more generic systems is ARF (A Road Finder) [21] and its associated road seeds generator RoadF [30]. ARF is (just like the system proposed in the next chapter) working on three different levels of abstraction. This system has two main low-level trackers, and the higher levels make these cooperate. In principle this system could have

been implemented using the logical sensor scheme, but to be able to cope with different resolutions, we would still need several algorithms, and let our system choose among these according to the resolution of the input scene.

2.3.1 Road-finding

In automatic road-seed generation, one usually computes a filter across the entire scene. This filter is supposed to give high responses for possible road pixels, and is usually constructed as a derivative filter, or as a filter sensitive to high spatial frequencies. These filters could be for example gradient estimators (such as the *Sobel* class of filters), *Canny* edge detectors or local frequency estimators.

wavelet transforms

Armin Gruen et al. [11] uses wavelets to transform the scene into a wave-space where high frequencies are enhanced. Their system is adapted to SPOT scenes.

wide roads

The RoadF system [30] works in slightly higher resolution imagery, and defines road seeds as centre pixels between two antiparallel intensity edges. i.e. each road has first a positive gradient slope, then a negative on the other side (or vice versa). This approach obviously won't work with roads of one pixel width.

2.3.2 Road-tracking

When tracing a suggested road section it is common practice to look at a cross section of pixels orthogonal to the road. For example one could use filter-masks that compute the mean of an assumed road segment, and the means on each side of this segment. If the intensity difference is notably larger than the internal differences on the assumed road segment, this is seen as an indication that the segment belongs to the road. This approach has been used in [14]

and in [8]. One obvious limitation in this approach is that, in order to be able to work with different input scene resolutions, we must be able to select filters from an exhaustive set. None of these systems can do this at present.

The ARF system [21] is slightly more generic; it follows a path in which the cross-section looks most similar to the current cross-section model of the system. This approach does not have the limitations mentioned above. However, it is more difficult to implement.

orientation field

Another way of tracing roads could be to use a local orientation field instead of the intensity-level field of the original scene as a basis for the road tracking algorithm. This approach has been tried in a slightly different context by Michele Covell [5]. She uses a local orientation field to create sketches of images by tracing the contours of objects. It would be interesting to investigate the feasibility of this approach on the road extraction problem.

curvature constraints

To get fewer false trails from the tracking algorithms, one can put an upper bound on the curvature of roads. (The system described in [14] does this.) This might (although it actually reportedly improves the results) not be such a good idea after all, as there de facto exist roads with sharp turns (although it probably would be better if it didn't!).

global algorithms

Sylvain Airault et al. [1] make some interesting notes on the road extraction problem, by suggesting the use of algorithms that are of more global nature to improve the results. Their system, however works on sub-meter resolution scenes, and their approach (segmentation) is thus difficult to adapt to satellite scenes.

Chapter 3

An attempt at man-made object detection

In this chapter the implementation results of this thesis work are presented. A detection system scheme is outlined, and its components are described in detail. It should however be noted that this system has **not** been completely implemented. In the subsections those parts that have actually been devised are presented.

3.1 System hierarchy

The hierarchy of the detection system about to be presented is based on the three levels of vision used in Computer Vision. (See “Computer Vision” on page 4.) Figure 5 below shows the general system structure.

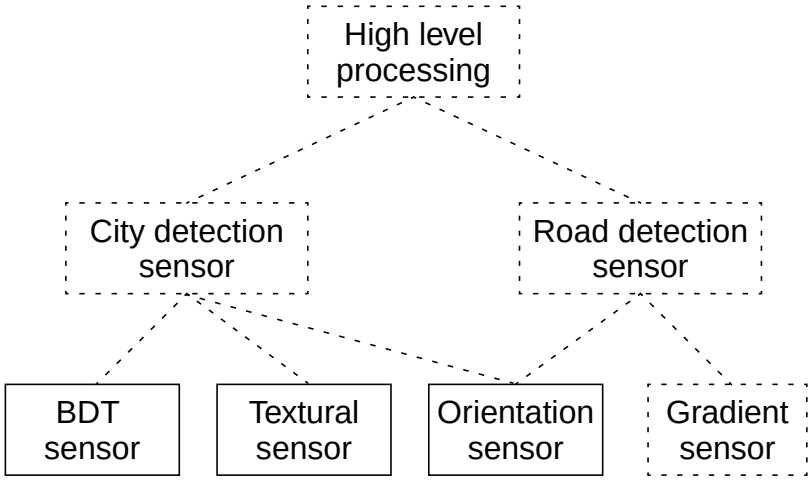

FIGURE 5. System outline

Logical sensors in the proposed detection system. Boxes with dotted outlines are not yet implemented.

The proposed man-made-object detection system has two main processing branches: one for detection of cities (left part of Figure 5) and one for detection of roads (right part of Figure 5).

3.1.1 High-level processing

The purpose of the top node in the detection tree (the “High-level processing” box) is to take into account spatial relationships between the detected objects, in order to improve the detection quality. This is also the controlling part of the system, as it directs the lower levels by setting their parameters. The high-level processing is also responsible for detection and resolution of logical sensor conflicts, such as overlapping objects.

The exact functionality of the top node is not yet determined. This could be another logical sensor if one desires a simple system, but it could also be some kind of AI-system, for example a *Blackboard system* (See Appendix C. “Blackboard Systems” on page 78) or a rule based Expert system. Another solution could be to make the system semi-automatic and place a human operator here.

3.1.2 City detection

The city detection will work in two steps. First the spectral properties of the city is taken into account by the **BDT-sensor,** and the textural properties are considered by the **Textural-sensor** and the **Orientation-sensor**. These three sensors will produce property fields that are later used by the **City-detection-sensor**. The City-detection-sensor is responsible for the initial segmentation and the discarding of too small “cities” (i.e. noise).

3.1.3 Road detection

The road detection works in a way not too different from the city detection. First coarse crest-lines (or road-seeds) are detected by the **Gradient-sensor**. These crest-lines are then tracked, joined and smoothed by the **Road-detection-sensor** using a local-orientation vector field as a guideline. The local-orientation field is provided by the **Orientation-sensor**. An alternative to the Gradient-sensor could be a multi-dimensional classification, using the properties orientation-presence (the magnitude of the local orientation) and width (local frequency). The quadrature-filters used could be weighted with respect to phase (roads are locally even functions) and width. After an initial road detection the Road-detection-sensor will trim the edges of the road-segments using a Snakes-like algorithm.

3.1.4 Information flow

If we consider the information flow within the system when this much has been said, the detection system certainly appears to have only bottom-up flows. To implement such a system and think that it could actually work is fairly naive. If we want the lower-level operations to remain simple we are forced to allow a certain amount of control information to run from the higher levels and down. It should however be possible to make all the sensors on the lowest level autonomous. This has been a major design goal for the sensors described in the subsequent sections.

iterative scheme

The **Road-detection-sensor** and the **City-detection-sensor** will need a lot of control information from above. These sensors will therefore work in an iterative manner, making slightly better detections in each consecutive pass. Typical control information is adjustments of threshold levels, classification cluster centres and so on.

Should the top node of the system tree be implemented as a *Blackboard System* (See Appendix C. “Blackboard Systems” on page 78) the information flow on the higher levels will be quite different. The detection-sensors will share their partial solutions to the problem with each other, and with any other system component (or *Knowledge Source*) that is added further on. One big difference, if the Blackboard scheme is chosen, is that the sensors on the level interfacing with the blackboard will no longer be strict logical sensors, as they will be actively fetching their new parameters from the blackboard.

shape estimation specialists

In a Blackboard system one could easily let all class detection specialists (such as road and city detectors) share *shape estimation specialists*. Each object on the blackboard will consist of a *frame* with fields for geometrical properties such as curvature and elongation. Whenever needed these fields can be filled in, regardless of which class the potential object belongs to.

3.2 Design of the BDT-sensor

BDT stands for Background Discriminant Transformation, and is a linear transform operating on multispectral images. The idea behind the BDT-algorithm is that a scene is composed of two main classes; foreground and background. Background constitute those areas that are definitely not of interest, while the foreground is assumed to be the rest of the scene.

3.2.1 Properties of the BDT transform

BDT is, just like PCT (Principal Component Transform) a linear transform of the property-space, working on one pixel at a time. The transform will maximize the variance in the foreground objects compared to the background objects. One could also say that the inter-class variation is maximized, and the within-class variation is minimized (See Figure 6 below).

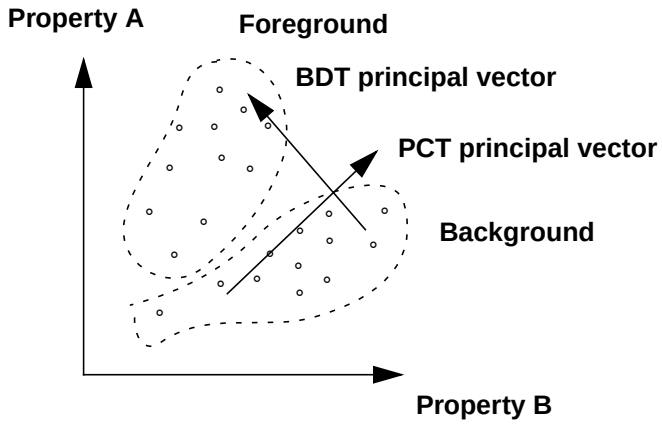

FIGURE 6. BDT compared to PCT

As PCT tries to maximize the variance for the entire set of pixels, the principal vector will not necessarily be good at discriminating the foreground and the background (the next vector probably will though in this example). In BDT the most discriminating direction is found directly.

BDT exhibits several properties that make it well suited to both satellite-image data, and to a logical sensor implementation. BDT is:

- **Scale invariant in property space**. This is needed in order to work with imagery from different sources. (See [27] for a mathematical proof.) This implies that the variance and mean value of the individual bands does not affect the algorithm performance.
- **Adaptive**. In that the background class may be re-computed for each scene.
- **Suitable for sensor fusion**. More bands can easily be added.
- **Robust**. The training can be used on similar scenes with adequate results. Even training from other satellites can be (and have been) used [28], provided that the wavelength bands are equivalent.

The use of BDT in the proposed system will be to extract a man-made-ness property of objects in the scene. This approach has previously been investigated by Shettigara et al. [28].

The reason for investigating this approach lies in the very characteristic reflectance functions of chlorophyll and iron-oxide. (See “Principles of man-made object detection” on page 21 for an explanation.)

3.2.2 Prerequisites of the algorithm

Before we can apply BDT, we will need to make a statistical analysis of the scene. We now view the entire scene as a matrix, **T** of size [<n> * <b>] where <n> is the number of pixels and <b> is the number of bands. Each column in **T** will correspond to a band and each row will correspond to a certain pixel. In a similar way we construct a sub-scene matrix, **B** containing known background pixels. To compute the main orientation of the transformation we will need the following measurements:

- $\mathbf{C_t}$ - the covariance of the scene matrix, **T** (size [<b> * <b>])
- $\mathbf{C_b}$ - the covariance of the sub-scene matrix, **B** (size [<b> * <b>])

- μ_t - the mean value of the columns in **T** (size [1 * <b>])
- $\mu_{\mathbf{b}}$ - the mean value of the columns in **B** (size [1 * <b>])
- r - the ratio of background and foreground pixel-counts.

3.2.3 The actual algorithm

We start by computing $\mathbf{C_a}$ – the inter-class covariance:

$$C_a = r \cdot (\mu_b - \mu_t)^t (\mu_b - \mu_t) \tag{3.1}$$

For details on how this and the following expressions are derived, the reader is directed to Shettigaras article [27].

The system best suited for maximizing foreground variance is now given by the eigenvectors of the matrix **D:**

$$D = C_b^{-1} C_w \tag{3.2}$$

Where

$$C_w = C_t - C_a \tag{3.3}$$

is the “within group” covariance.

To find the eigensystem of **D** we first decompose $\mathbf{C_b}$ into two matrices that are each others transpose:

$$C_b = MM^t \tag{3.4}$$

Cholesky decomposition

This is accomplished by Cholesky decomposition (see the book “Numerical Recipes in C” [26] for an implementation). Note that Cholesky decomposition only works on symmetric, positive definite matrices. Of course $\mathbf{C_b}$ has these properties.

We may now create a matrix **Q**:

$$\boldsymbol{Q} = \boldsymbol{M}^{-1}\boldsymbol{C}_{\boldsymbol{w}}(\boldsymbol{M}^{-1})^{\boldsymbol{t}} \tag{3.5}$$

The eigenvectors of **Q** can be computed (for example by Jacobi rotation [26]) as **Q** is symmetric. The idea is that the eigenvectors of **Q**, $\mathbf{q_i}$ are related to the eigenvectors of **D**, $\mathbf{w_i}$ through the following transformation:

$$\boldsymbol{w_i} = k_i(\boldsymbol{M}^{\boldsymbol{t}})^{-1} \cdot \boldsymbol{q_i} \tag{3.6}$$

(Where k_i are scaling factors.) The reason for this is that:

$$eig(\boldsymbol{Q}) = eig(\boldsymbol{M}^{-1}\boldsymbol{Q}) \tag{3.7}$$

(See Gnanadesikans book [9] for an explanation.)

If we are only interested in a property-field describing the degree of man-made-ness, we need only project our pixels onto the eigenvector corresponding to the largest eigenvalue of **D**.

overturning the eigenvectors

Due to the fact that an eigenvector is not defined as an explicit vector, but rather as a linear sub-space, the eigenvector algorithm sometimes return a vector that yield positive values for the background instead of for the foreground upon projection. When this happens we simply overturn all eigenvectors so that they point in the antipodal (the direction furthest away from the current) directions. The process of overturning the eigenvectors is mathematically a sign change, and it is accomplished by:

$$\boldsymbol{w} = \boldsymbol{w}' \cdot \frac{\boldsymbol{w}' \bullet \boldsymbol{w}_p}{\left|\boldsymbol{w}' \bullet \boldsymbol{w}_p\right|} \tag{3.8}$$

where $\mathbf{w_p}$ is a prototype to $\mathbf{w}$ which has the correct general direction. If we have no prototype yet we might as well use a vector that is the antipode to the average background direction.

We are now ready to do the actual transformation. As we are only interested in the principal direction of the transform, the transformation becomes an ordinary projection of scene pixels onto $\mathbf{w}$:

$$g(x, y) = \boldsymbol{p}(x, y) \bullet \boldsymbol{w} \tag{3.9}$$

Here g(x,y) is the resulting discriminant function of our man-made-ness property.

normalization

However to make the property more robust to intensity variations (pixels that are intense in all bands, such as cloud-pixels, will always give large results), we *normalize* the result through division with the norm of each pixel. This will result in a value in the range [-1,1]. Finally the result is shifted by 1 (to make all values positive) and scaled by L/2, where L is the total number of available intensity levels:

$$g(x, y) = \left(1 + \frac{\boldsymbol{p}(x, y)}{|\boldsymbol{p}(x, y)|} \bullet \boldsymbol{w}\right)\frac{L}{2} \tag{3.10}$$

3.2.4 BDT as a logical sensor

The input to the BDT sensor (Figure 7 below) is assumed to be a multiband satellite scene. The algorithm itself does not impose a limit to the number of bands. The input bands are combined with descriptions of which wavelength intervals they have sensed.

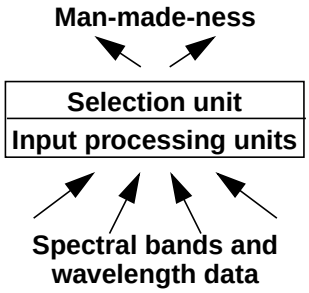

FIGURE 7. The BDT logical sensor

3.2.5 Identifying the background

The BDT-sensor is equipped with detailed knowledge of plant reflectance in the form of a reflectance curve. The reflectance curve is used to extract coordinates for an initial, coarse transformation. The coordinates are obtained by averaging the reflectance curve in the wavelength-interval associated with each band in the scene (See Figure 8 on page 41). Note that the transformation now will be *independent of the scaling* of the reflectance curve: The transformation vector points in the same direction regardless of scale, and the vector is normalized before the projection is computed.

The initial transform is only used for finding a training area for the background. This area is selected as all pixels where the response of a low-pass (Gaussian) filtered version of the initial projection is below a certain limit. Equipped with this training area we may now run the original algorithm.

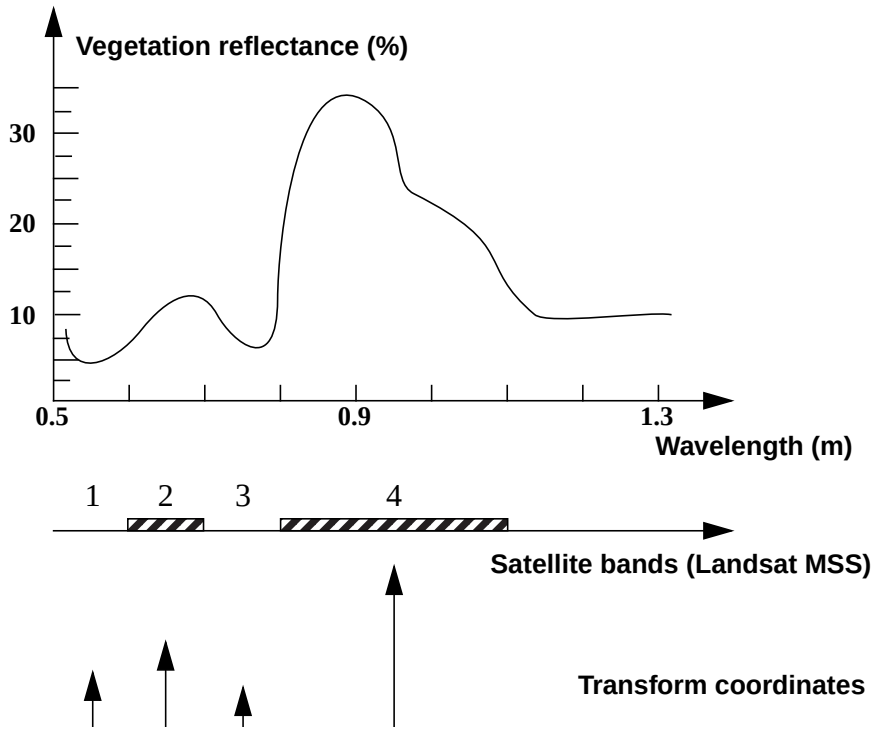

FIGURE 8. Reflectance curve

Extraction of initial transform coordinates using a reflectance curve.

3.2.6 The fraction parameter

We now have just one parameter left, the ratio of background and foreground pixel-counts. If this ratio is overestimated, one or several of the eigenvalues in **Q** will become negative. This has been suggested by Shettigara as a way to find an upper limit for the ratio [27]. The method is however a bit time consuming, as it involves multiple eigenvalue computations. However, there is an easier way. We could try to apply Cholesky decomposition on the matrix Q. This will fail if the matrix **Q** is not positive definite. The possibility of using this feature of Cholesky decomposition (in combination with for example a binary search algorithm) ought to be investigated. It would indeed be nice if we could make the BDT-algorithm completely automatic.

3.2.7 BDT results

To illustrate exactly what the BDT transform accomplishes, we will first look at a sample image from the French satellite SPOT depicting the Swedish city of Nybro. (See Figure 9 below.) As we can see the outline of the city is not hard to find. This is because band 2 of SPOT is located at 0.61–0.68 µm (the red part of the spectrum) where the chlorophyll absorption is strong (man-made objects are thus more intense).

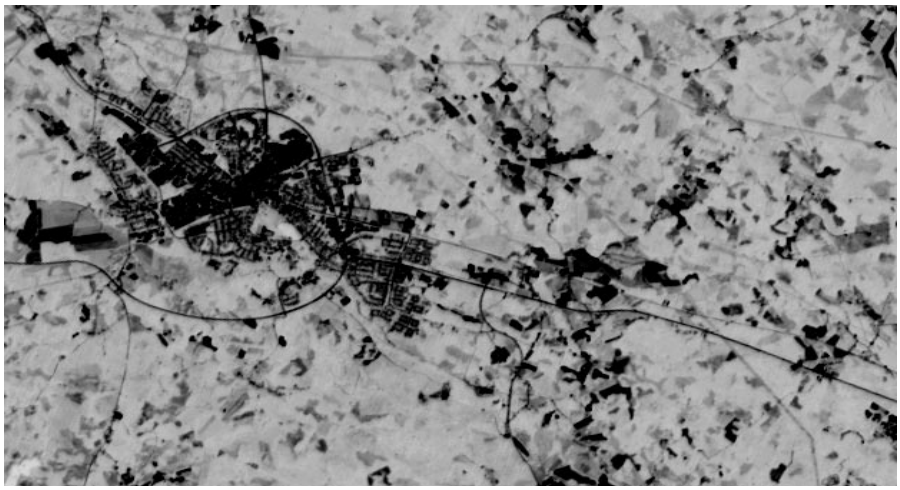

FIGURE 9. Nybro area scene

SPOT XS band 2 (contrast stretched).

If we look at Figure 10 on page 43 where the algorithm has been used, the discrimination abilities are not dramatically better than what could have been accomplished by plain thresholding of band 2. These results hardly justify the efforts.

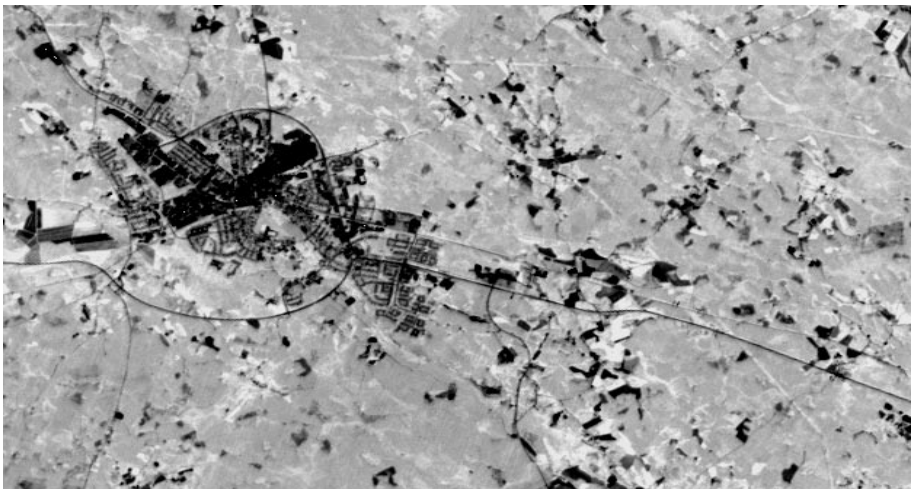

FIGURE 10. Non-normalized BDT of the Nybro area

If we look at Figure 11 below, where the transform has been normalized we notice that the contrast is smoother, and some false hits have been eliminated. The main reason for the improvement is that normalization reduces the influence of shading of roads and leakage of reflected light onto nearby structures. If we could only get rid of the noise outside the city we would now be satisfied with the results. For this however, we will need other tools.

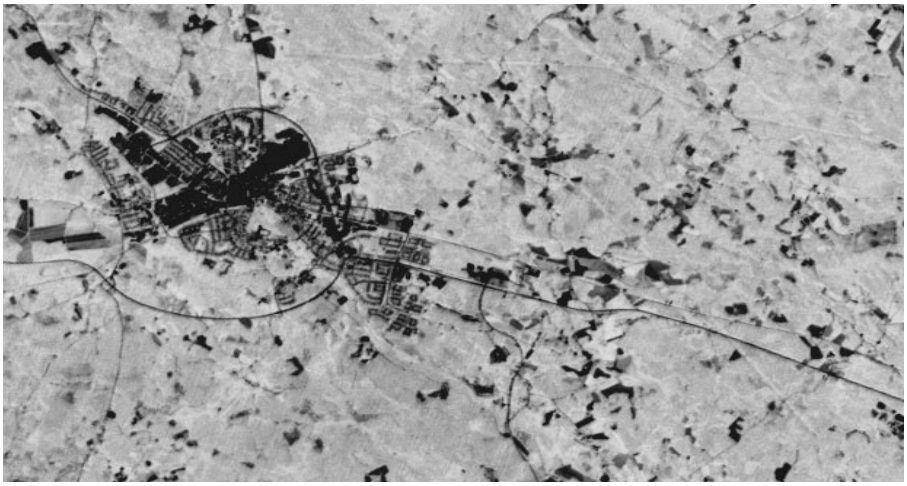

FIGURE 11. Normalized BDT of the Nybro area

NOTE: All the test-images have had their contrast adjusted in order to look better in print. For this reason SPOT band 2 and the non-normalized BDT look very much alike this scene. If we decide to threshold the scenes, the differences will be more apparent.

3.2.8 Problems with BDT

One disadvantage with BDT is that the algorithm assumes that there are two main classes in a scene. While this works fine on many scenes that cover land-areas, it fails completely on coastal regions. (See scene in Figure 12 below.) In this kind of scenes we have not two, but (at least) three distinct classes, and the new one, the sea class, is causing trouble.

FIGURE 12. Coastal scene north of Kalmar

SPOT XS band 2.

If we train BDT with only vegetation as background class, the water regions will be included in the foreground (See Figure 13 below).

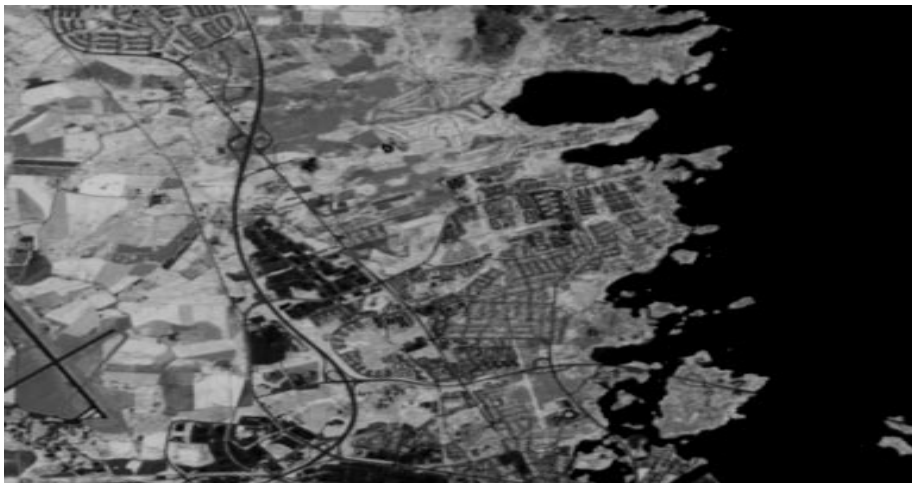

FIGURE 13. Coastal scene trained with vegetation

If we try to train BDT such that water and vegetation is one class the result will be a property where cities and some of the vegetation is enhanced. Neither of these alternatives is very appealing.

One solution to this problem is to extract the water class first, and remove all water pixels before BDT is applied. (See Figure 14 below). Water extraction is easily accomplished by direct classification of the original scene.

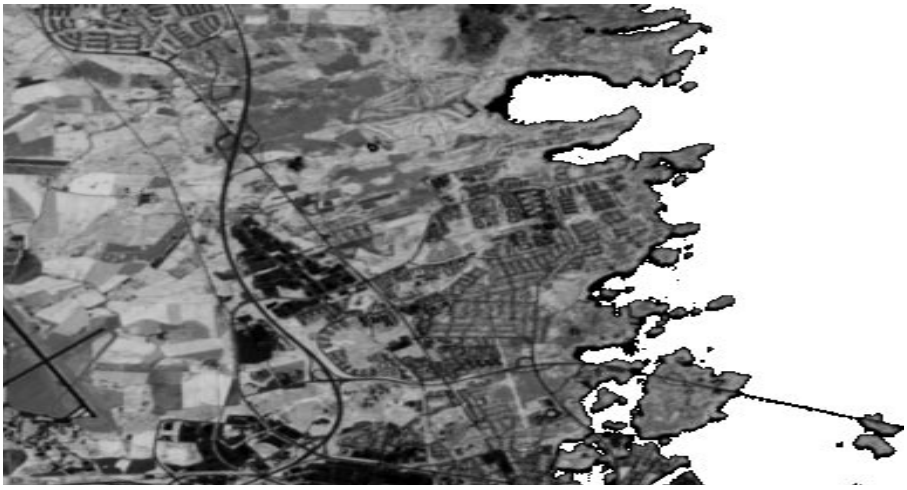

FIGURE 14. Coastal scene after water removal

Another solution could be to ignore the problem altogether on this system level, and let the next level take care of this, for example by always choosing the water class in case of a conflict with the city class. Which solution is the best has not yet been evaluated, but the second alternative has the possible disadvantage that the BDT transform *could* suggest a non optimal transformation direction, as part of the background class (the water) is included in the foreground class.

3.3 Design of the Orientation sensor

The Orientation-sensor will be used in both city and road detection, but for two radically different purposes. The city detection will use the output as a geometrical (or textural) property field, while the road detection will obtain a local orientation field to be used in road-tracking.

quadrature

Quadrature is a property that complex-filter kernels exhibit when the real and the imaginary parts extract features that are orthogonal. Typically the real-part of the filter extracts lined-ness and the imaginary-part extracts edged-ness.

quadrature filters

Quadrature filters constitute a special class of filters that have their properties defined in the Fourier domain. The reason for using quadrature filters in orientation estimation is that such filters can be made phase invariant. The actual filter-kernels are constructed in the spatial domain by finding the closest approximation to the ideal filter (which is defined in the Fourier domain). Details on how to implement quadrature filters will not be discussed here, but they can be found in the book “Signal Processing for Computer Vision” [10].

local orientation

In the Orientation-sensor we will use quadrature filters to extract the presence of orientation. The sensor will combine four filter responses into a single vector-field, describing the *local orientation* in the scene. The local orientation is defined as the direction of local variation in a scene [10]. This means that ***the local orientation near a line or an edge in a scene will be perpendicular to the feature in question***.

3.3.1 Construction of a local orientation estimate

The quadrature filters are designed such that the magnitude of each filter response describes the presence of orientation along one of four evenly spaced directions. (See Figure 15 below.)

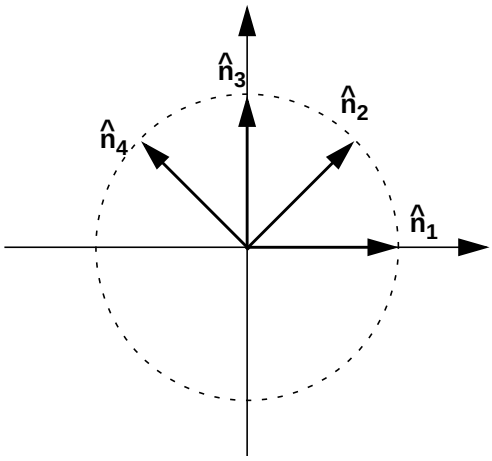

FIGURE 15. Quadrature filter directions

The vectors n_1 through n_4 indicate the orientations that the quadrature filters are sensitive to.

The four filter output magnitudes q_k are mathematically described as:

$$q_k = A(\hat{\boldsymbol{x}} \cdot \hat{\boldsymbol{n}}_{\boldsymbol{k}})^2 \tag{3.11}$$

where **x** is the dominant local orientation of the scene that is sought. The scalar A is proportional to the local amplitude, and is independent of the orientation [10].

Through the definition of the scalar product:

$$\boldsymbol{x} \cdot \boldsymbol{y} = |\boldsymbol{x}||\boldsymbol{y}| \cos\varphi \tag{3.12}$$

we see that q_k is related to the angular difference φ, between the principal direction of the filter **n**, and the sought local orientation in the scene, **x**:

$$q_k = A(\cos\varphi)^2 \tag{3.13}$$

(We end up with only the cosine term as both vectors have the norm 1.)

local orientation angle

We now define the local orientation angle ϕ as zero along $\mathbf{n_1}$ and increasing counter-clockwise. The quadrature-filter responses of q_1 and q_3 can now be written as:

$$q_1 = A(\cos\phi)^2$$

$$q_3 = A\left(\cos\left(\phi + \frac{\pi}{2}\right)\right)^2 = A(\sin\phi)^2$$

Using this circumscription we can obtain one coordinate of the sought orientation by subtracting q_3 from q_1:

$$q_1 - q_3 = A(\cos\phi)^2 - A(\sin\phi)^2 = A\cos(2\phi) \tag{3.14}$$

In a similar way we get:

$$q_2 - q_4 = A\cos\left(2\left(\phi - \frac{\pi}{4}\right)\right) = A\sin(2\phi) \tag{3.15}$$

Thus, a representation of the local orientation may be constructed as:

$$\boldsymbol{z} = \begin{bmatrix} q_1 - q_3 \\ q_2 - q_4 \end{bmatrix} = A\begin{bmatrix} \cos(2\phi) \\ \sin(2\phi) \end{bmatrix} \tag{3.16}$$

double angle representation

Note that the coordinates of **z** are related to the double angle of the orientation. This is desirable as we now have only one representation of each orientation, even though there are two directions of variance along each orientation. (See Figure 16 below.) For an in-depth discussion of why this representation is desirable, the reader is directed to the book “Signal Processing for Computer Vision” [10].

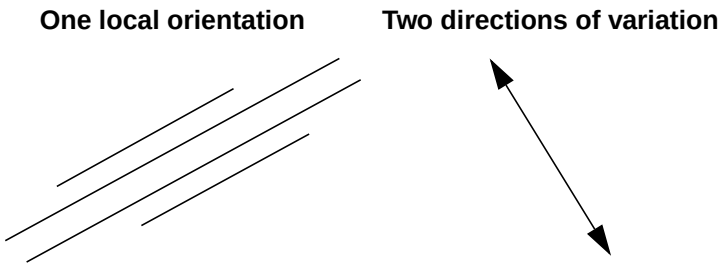

FIGURE 16. Orientation and direction

local amplitude

Once we have computed the filter responses q_1 through q_4, we may extract another property, namely the *local amplitude* of the signal. The filter responses were:

$$q_k = A(\hat{\mathbf{x}} \cdot \hat{\mathbf{n}}_k)^2$$

If we decide to write **x** as a sum of components along, say $\mathbf{n_1}$ and $\mathbf{n_3}$ we get:

$$q_k = A((x_1\hat{\mathbf{n}}_1 + x_3\hat{\mathbf{n}}_3) \cdot \hat{\mathbf{n}}_k)^2 \qquad (3.17)$$

Where x_1 and x_3 are the coordinates of **x** in the basis ($\mathbf{n_1}$, $\mathbf{n_3}$).

For q_1 and q_3 this becomes:

$$q_1 = Ax_1^2$$

$$q_3 = Ax_3^2$$

And as $\mathbf{n_1}$ and $\mathbf{n_3}$ are orthogonal these filter responses will sum to:

$$q_1 + q_3 = A(x_1^2 + x_3^2) = A \tag{3.18}$$

This line of reasoning is valid for q_2 and q_4 as well, so a good estimate of the local amplitude will be:

$$A = \frac{1}{2}\left(\sum_{i=1}^{4} q_i\right) \tag{3.19}$$

This is a property which is useful in texture discrimination.

3.3.2 Design parameters

Until now, nothing has been said about how the filters are actually designed. In the Fourier domain the degrees of freedom in a quadrature filter are expressed as one *angular function* and one *radial function*. The angular functions of the four filter-kernels are coerced into $\mathbf{n_1}$ through $\mathbf{n_4}$, but we are free to model the radial functions to suit our needs. Usually the radial function is modelled as a log-normal function:

$$R_i(\rho) = e^{-C_B \ln(\rho/\rho_i)} \tag{3.20}$$

scale dependency

Here ρ is the radial parameter, ρ_i is the centre frequency and C_B is a parameter that is a function of the filter bandwidth. It is now (mathematically) evident that *local orientation estimation depends on the scale at which we view the scene*.

This seems to imply that the spatial resolution of the input sensor needs to be known before we design our filter. This is however not the case, as there are only a limited range of resolutions that are useful for road detection.

For resolutions below 30 square meters most roads are less than one pixel wide. Thus, long sections of the roads are not visible at all, and road detection will practically be very difficult. For resolutions higher than 30 square meters we might design several filter sets beforehand, and select one filter set once the spatial resolution is known. Another alternative is to design a filter with a wide pass-band such as one adapted for 1 through 30 square meter orientations.

3.3.3 Orientation as a logical sensor

The Orientation-sensor (see Figure 17 below) will receive all available bands as input. However, as the orientation estimation is rather time-consuming, only one band will be selected as the input of the computation. This band should lie within the range 0.5 through 0.8 µm, where the chlorophyll absorption is strong.

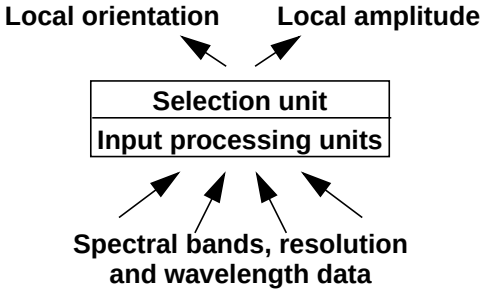

FIGURE 17. The Orientation logical sensor

multiple filter-kernel sets

Initially, the sensor will have only one set of filter-kernels and thus accept only scenes with spatial resolutions of 20 through 30 square meters. This range includes the common **Landsat TM** and **SPOT multispectral** scenes. (See the appendix “Earth monitoring satellites” on page 80) In a later stage more kernels may be added. The selection unit (see Figure 17 above) will then select an appropriate set of kernels based on the scene resolution.

3.3.4 Usage of local orientation

The local orientation field is used in the road detection branch of the system. The purpose is to trace and join the road-seeds we have extracted using the Gradient-sensor. Local orientation is also useful for extracting textural properties, and the Orientation-sensor will be used for this purpose in the city detection, where we also will make use of the local amplitude property discussed in “Construction of a local orientation estimate” on page 47.

3.3.5 Orientation sensor results

To illustrate exactly what the local orientation and the local amplitude measures are the algorithms have been tested on the Nybro scene. (See Figure 9 on page 42.) Unfortunately the local orientation is a vector property, and can thus not be presented in black and white only. Figure 18 below shows the magnitude of the local orientation field. We can see that roads and city structures are enhanced. These are the areas where the local orientation is certain.

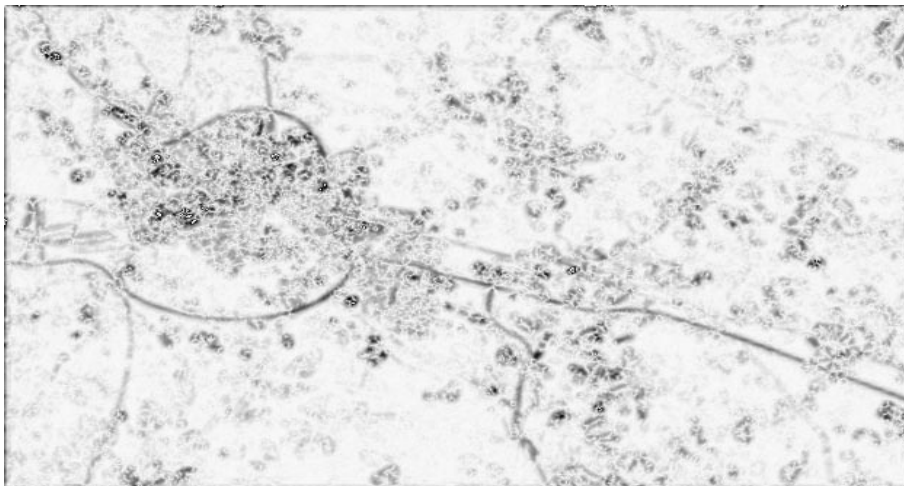

FIGURE 18. Magnitude of local orientation for Nybro scene

The effect of a local amplitude is shown in Figure 19 on page 53. This is obviously a property that is useful for discriminating cities.

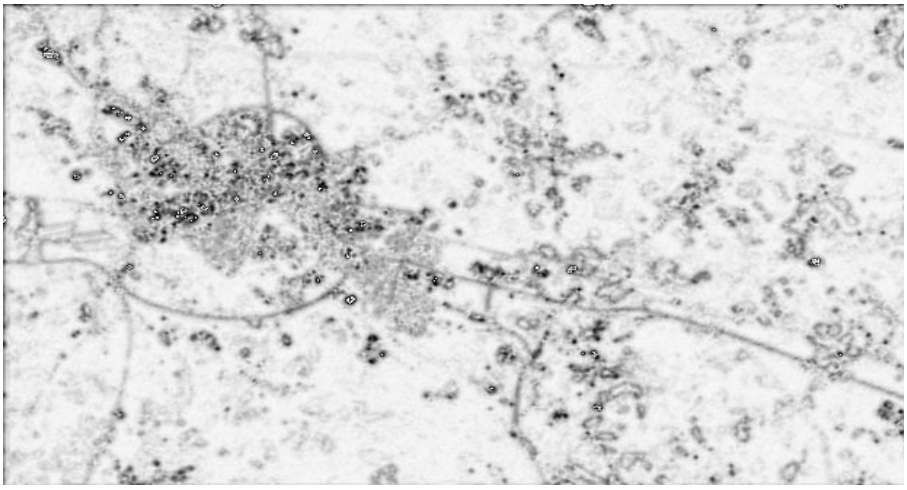

FIGURE 19. Local amplitude for Nybro scene

3.4 Thoughts on a Gradient sensor

During this thesis work a gradient algorithm has also been implemented. The reason for implementing this algorithm has not been to find an optimal road-seeds detector, but rather to examine how well a stand-alone gradient operator performs at finding roads.

A measure of the gradient magnitude of a two dimensional scene can be obtained from two orthogonal partial derivative measures. The magnitude is computed as the square-root of the squared partial derivatives of a scene **I**:

$$|\boldsymbol{D}| = \sqrt{\boldsymbol{D}_{\boldsymbol{x}}^{2}(\boldsymbol{I}) + \boldsymbol{D}_{\boldsymbol{y}}^{2}(\boldsymbol{I})}$$

Typical approximative directed derivative kernels are:

$$\boldsymbol{D}_{y} = \begin{bmatrix} 1/2 \\ 0 \\ -1/2 \end{bmatrix}$$

$$\boldsymbol{D}_{\boldsymbol{x}} = \begin{bmatrix} 1/2 & 0 & -1/2 \end{bmatrix}$$

The reason for choosing these measures becomes evident if we decompose the $\mathbf{D_x}$ partial derivative kernel:

$$\boldsymbol{D}_{\boldsymbol{x}} = \begin{bmatrix} 1 & -1 \end{bmatrix} \bullet \begin{bmatrix} 1/2 & 1/2 \end{bmatrix}$$

(The dot-operation above denotes convolution.) The first term is easily identified as a difference i.e. an approximation of the derivative. The second term is a typical low-pass filter kernel. Low-pass filtering is added to reduce the

influence of the truncation error inherent in the approximative derivative. We shall now pursue this line of reasoning one step further and convolve each partial derivative with an orthogonal smoothing kernel:

$$\boldsymbol{D}_y \bullet \frac{1}{4}\begin{bmatrix}1 & 2 & 1\end{bmatrix} = k\begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix}$$

$$\boldsymbol{D}_x \bullet \frac{1}{4}\begin{bmatrix}1\\ 2\\ 1\end{bmatrix} = k\begin{bmatrix}1 & 0 & -1\\ 2 & 0 & -2\\ 1 & 0 & -1\end{bmatrix}$$

We have now arrived at the commonly used *Sobel Operators* [15]. The output of this operator is shown in Figure 20 below.

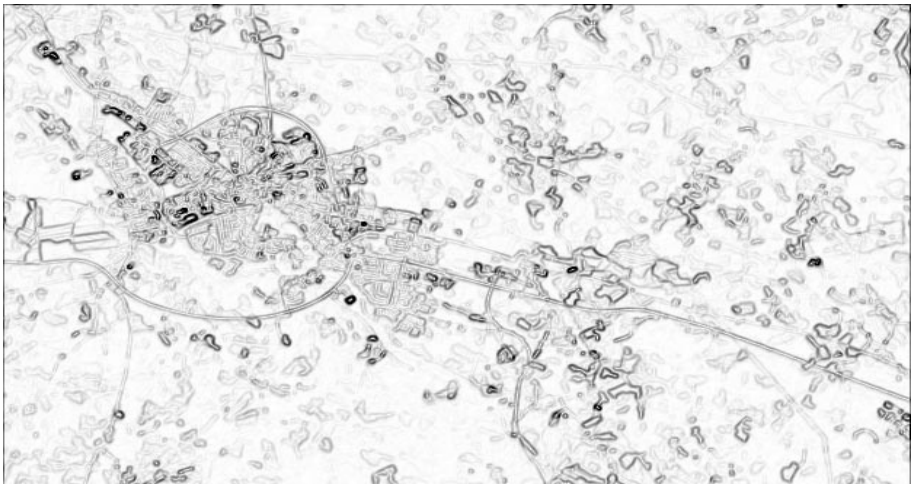

FIGURE 20. Gradient magnitude for Nybro scene

While the gradient magnitude is large where the scene contains roads, it is obvious that more information from the scene needs to be considered if we want to extract road-seeds.

Note that the Sobel operators are not the best known way to detect edges. However, they are easy to implement, and their purpose here was only to find out what a stand-alone edge detector could accomplish.

3.5 Design of a Textural sensor

One thing that makes cities stand out in a scene is the presence of a mesh of streets. The streets are separating and leading to rectangular (or at least polygonal) buildings. The kind of ordered pattern that this creates is called a *texture*. Urban areas exhibit a quite distinct texture. This is the reason why the city detection branch of our system should include a textural sensor.

texture parameters

Bernd Jähne has in his book “Digital Image Processing” [15] provided us with a list of parameters that describe textures. These are organized in a hierarchical manner according to complexity, and are listed below:

- Average intensity level
- Local variance
- Local orientation
- Characteristic scale
- Variance of local orientation
- Variance of characteristic scale

The applicability of some of these measures to city discrimination will be examined in the following sections. The measures involving characteristic scale have not been tested, as they are fairly complex to implement. Nor do they exhibit any properties that suggest that they should be applicable to this kind of detection problem.

3.5.1 Average intensity level

While the average intensity level may be a useful texture measure, it is very closely related to the kind of properties that the BDT-sensor extracts. BDT extracts intensity information from all available bands, and combines it into one property especially constructed for city discrimination. A low-passed version of the BDT-property field subsumes the average intensity level information in *all* bands.

3.5.2 Local Variance

In every-day language local variance can be said to correspond to smoothness and raggedness of a surface. Most of the ideas on local variance presented here originates from the book “Digital Image Processing” [15].

In statistics local variance is defined as:

$$V_x(n) = \frac{1}{N-1}\left\{\sum_{k \in \boldsymbol{L}} (x_k - \mu_x(n))^2\right\}$$

where μ_x(n) is the local mean value of the signal and **L** is the region with N samples in which we compute the variance. In two dimensions this becomes:

$$V_{xy}(m, n) = \frac{1}{N^2-1}\left\{\sum_{k,\, l \in \boldsymbol{L}} (x_{kl} - \mu_{xy}(m, n))^2\right\}$$

The size of the region **L** in these equations determines how sharp (or local) the filter should be. For example a 5x5 pixel environment produces a less sharp looking property than a 3x3 pixel environment.

isotropic operations

With a local variance operator we want a measure of how the intensity varies *near the current pixel*. The operator V is however not so well suited to that purpose. The crucial word here is *near*. The region we have selected for our operation is quadratic, this means that we will consider pixels further away from the current in the diagonal directions. In other words the operator V is not *isotropic* or direction independent.

Gaussian filters

By replacing the mean operation (summation and division with (N^2-1)) with a Gaussian weighted average (or similar), we can make the variance operation isotropic. To do this we multiply each sample in the current neighbourhood with the corresponding coefficient of a *Gaussian filter*

kernel. A Gaussian filter kernel uses values from the two-dimensional PDF (probability distribution function) of the normal distribution. In order not to displace the mean level of the intensities, the filter coefficients should be scaled so that they sum to 1.

We have now devised a new variance operation:

$$V_{xy}(m, n, \sigma) = \sum_{k, l \in \boldsymbol{L}} G(\sigma, k, l)(x_{kl} - \mu_{xy}(m, n))^2 \quad (3.21)$$

Where G(σ, k, l) are the Gaussian kernel weights.

scene representation

The computation error introduced by the use of a non isotropic kernel may not be that big, but we should carefully avoid it anyway, as the use of Gaussian filters is a more representation independent method. One should always remember that ***the scene does not consist of pixels in a square grid, it is merely represented in that way.***

standard deviation

When we replace the plain averaging with a Gaussian convolution we introduce a new parameter to the operation, the standard deviation of the Gaussian PDF (from now on denoted STD). This parameter allows us to adjust the locality of our filter in sub-pixel accuracy. A rectangular kernel with size 3x3 is approximately equivalent to a Gaussian kernel with STD = 1.0065 and a 5x5 kernel corresponds to STD = 1.798.

frequency sensitivity

When examining the results of the operation in Equation 3.21 for different STD values, it seems like the STD parameter controls the high-pass content of the resulting property. This is not altogether the case. The pixelwise squaring causes a few large differences in intensity to be higher valued than many small differences i.e. it makes the variance

operator sensitive to higher frequencies than if we had just computed the magnitude of the differences. Thus the STD parameter should be seen only as a means to locality regulation, and not as a frequency sensitivity parameter.

intensity adjustment

If we compare the histograms of the variance operation for small and large STD values respectively, we notice that the larger the STD parameter is, the stronger is the response of the filter. If we want to make our operator work on several kinds of scenes with for example the same threshold levels we need to adjust the intensities such that they are independent of the STD parameter. If we plot the average intensity as a function of the STD parameter (See Figure 21 below) we notice that this relationship is neither linear, nor logarithmic to any degree.

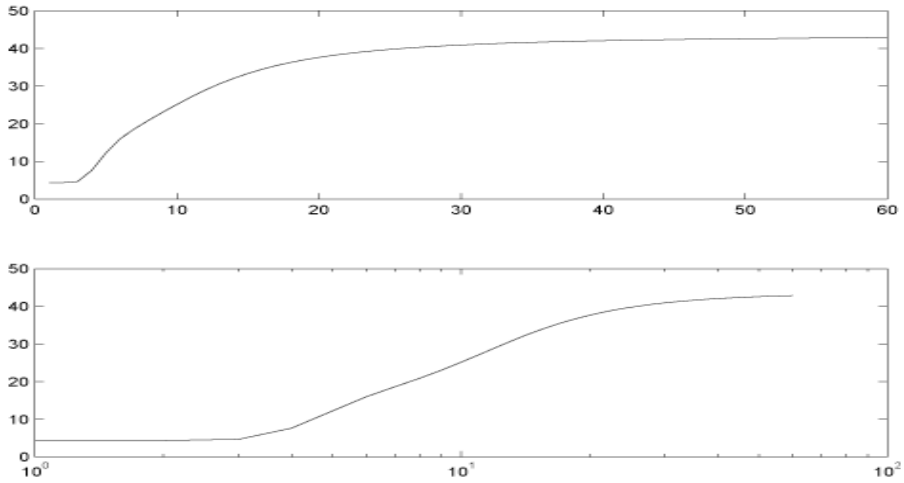

FIGURE 21. STD dependency

Linear and logarithmic plots of STD dependency.

For this reason the intensity adjustment is accomplished through a lookup-table, from which we interpolate a scaling factor for the intensity:

$$A_f = \frac{S}{T_l[i](1-f) + T_l[i+1]f} \qquad (3.22)$$

Where T_l is our lookup-table consisting of average intensities from the non-adjusted algorithm and S is a scaling factor.

The variables f and i are computed as:

$$f = d\sigma^2 - \lfloor d\sigma^2 \rfloor$$

$$i = \lfloor d\sigma^2 \rfloor$$

Where d is a parameter denoting the density of our lookup table (The incomplete brackets denote a *floor* or "truncate to integer" operation). For example d = 10 means ten samples per unit of σ^2.

An example of the local variance operation is shown in Figure 22 below.

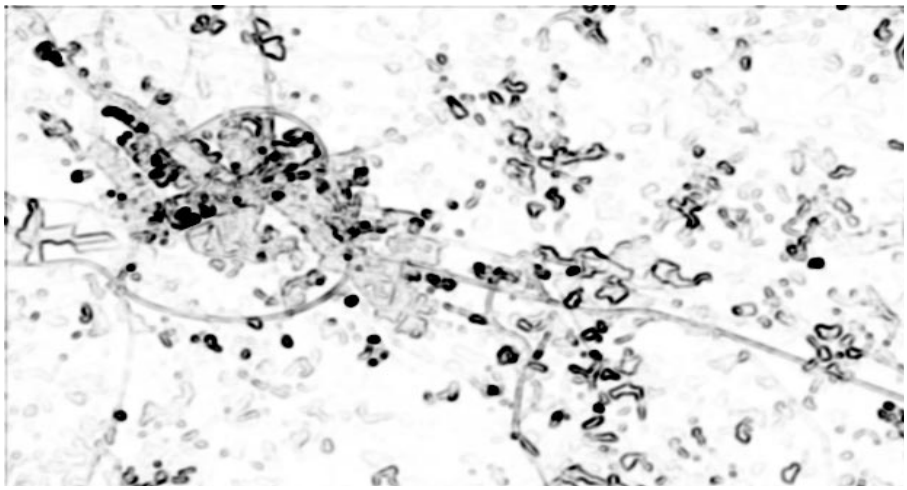

FIGURE 22. Local variance for Nybro scene

STD = 1.5.

When we use a fairly low STD value (as we have done in Figure 22) we end up with a property that preserves the true edges of the city. This property field could actually be used by an energy minimisation algorithm to find the city outlines.

3.5.3 Local Orientation

The local orientation feature has previously been discussed in the section “Design of the Orientation sensor” on page 46. While local orientation is ample for some kinds of texture discriminations it is not applicable on city detection as is. On the contrary, cities are characterized by non-homogeneous local orientations. The magnitude of the local orientation, on the other hand is not all that bad in city discrimination. However, it will not be used as it is too similar to the local variance property.

3.5.4 Variance of Local Orientation

If we define variance of local orientation as the variance of the orientation angle scaled with the magnitude of the local orientation we end up with an operator that produces the result of Figure 23 below. This result is rather noisy, and will thus be difficult to use.

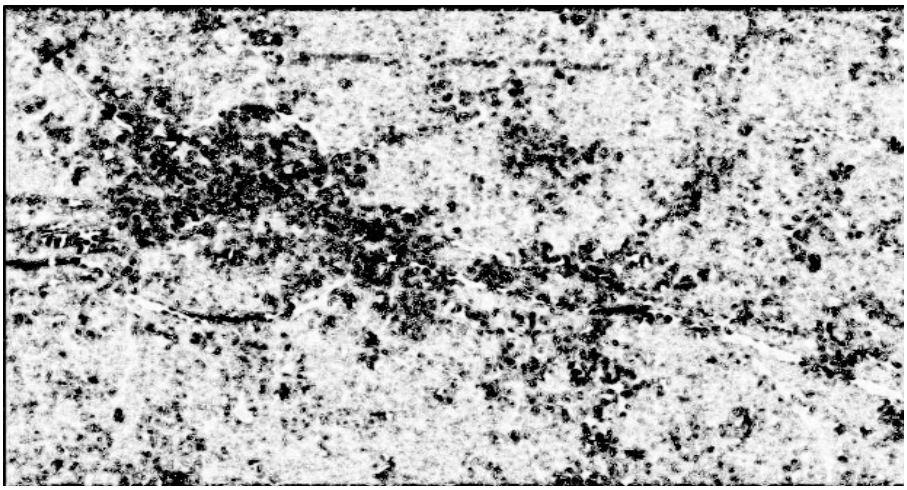

FIGURE 23. Variance of orientation for Nybro scene

STD = 1.

If we instead compute the variance of the local orientation along two orthogonal axes, and incorporate the magnitude in the components, we end up with the property shown in Figure 24 below. This result is obviously much more useful for city detection.

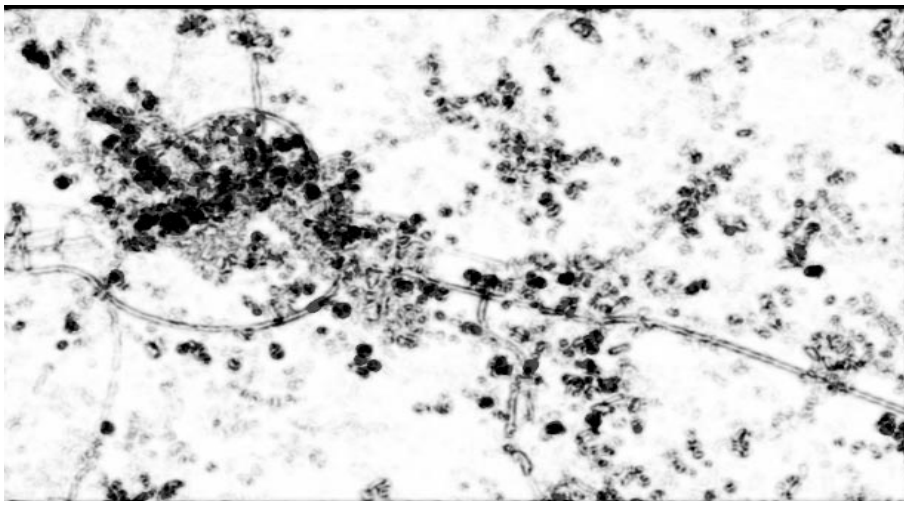

FIGURE 24. Component-wise variance of local orientation

STD = 1.

The components of the orientation combination are combined using the typical vector-norm:

$$|\boldsymbol{v}| = \sqrt{v_x^2 + v_y^2} \tag{3.23}$$

where v_x and v_y are the outputs of the variance operation on the x any y components of the local variation respectively (i.e. the coefficients of **z** in Equation 3.16).

3.5.5 Bands suitable for texture operations

The texture operations described in this section will not be applied to all the bands in a scene. The reason for this is that most of the textural properties extracted from different bands are either equivalent or useless in the discrimination process. As too many property fields may actually impede the performance of the classifier (see Appendix B

"Classification" on page 75) we will select only one band in each scene for textural processing. When testing the textural operation on **SPOT multispectral** and **Landsat TM** scenes these results were obtained:

SPOT bands

For SPOT multispectral the bands 2, and 3 are suitable. Band 1 is not good for discrimination of vegetation from rocks and metals, so the variance fails for this reason. Band 2 seems to bee slightly better than band 3, so this is the band we will choose.

Landsat bands

For Landsat TM, the bands 1, 2 and 3 works fine for variance operations. Among these, band 3 is superior, and will thus be selected. If we look at the wavelength ranges the bands cover, (see Appendix D "Earth monitoring satellites" on page 80) one would suspect that Landsat band 4 would also work, as it covers almost the same range as SPOT band 3. The reason why this is not the case is that the lower resolution in the Landsat scenes "hides" the mesh of streets within the cities.

band selection

From these results we can conclude that the selection unit of the Textural sensor should consider both resolution and wavelength range when selecting a band for texture operations. As only two data-sources have been tested yet, no algorithm for band selection is proposed. A table with entries for SPOT and Landsat will do for now.

Chapter 4

Discussion

In this chapter an attempt will be made to assess this thesis work and the problem definition in a more objective manner.

4.1 The Problem Definition

We will now attempt to compare what is said in the problem definition to what has actually been accomplished during this thesis work.

Now the work is finished, but no actual detection system has been produced (just a sketch). Those parts that have actually been implemented are all of low-level nature. However, the intention was never to produce a complete working system.

The implications of the logical sensor scheme on the information-flow is described in general terms in the section “Logical sensors” on page 8. It should also be possible to conclude how things will work in practice if one examines the section “System hierarchy” on page 32.

Not much actual implementation has been done on high-level methods, but this was never actually demanded in the problem definition (it would have been interesting to pursue the suggested paths though). Still, most of this work has been conducted with a high-level control mechanism in mind, and the principles of high-level vision has been thoroughly analysed.

To look at existing detection systems has been sort of a detective work. Most articles in this field are short, and describe only a small part of a complete system. What is said on detection systems in “Principles of man-made object detection” on page 21 is therefore compiled from a number of sources.

Much has been said on road extraction in the section “Road Extraction” on page 27, however not much has been tested or said on the subject in the implementation chapter. The reason for this is that the road extraction problem has been found to be fairly complex, and even a partial implementation would easily fill an entire masters thesis.

4.2 On the results

The main results in this thesis are an overview of existing detection systems, three implemented logical sensors and a system outline.

detection systems

The overview of the detection systems is written in a fairly general manner. There is however no guarantee that all existing algorithms fit into this description. This is equally valid for the road-extraction section.

the BDT-sensor

The BDT-sensor has many properties favourable to image interpretation in remote sensing. One big drawback with the BDT-algorithm was that it failed near water. This fact was never mentioned in the articles where it was presented (probably because it was never tested on near water scenes). However, a solution to this problem has been found, and the algorithm is still quite useful for automated city detection.

The usefulness of the BDT-algorithm may be partly a coincidence though. As the algorithm tries to maximize the variance in the foreground class while minimizing the variance in the background class there is a special case where BDT will fail completely to discriminate the foreground (see Figure 25 below).

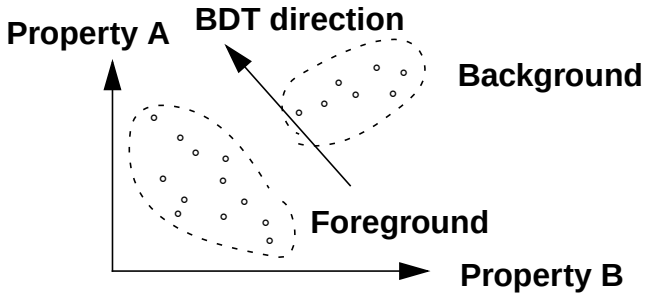

FIGURE 25. BDT failure

Fortunately this situation does not appear to occur with vegetation as background and man-made structures as foreground. If it did, we could always use the last transform component, where the background variance is maximized and the foreground variance is minimized. This situation does however need to be detected by the system somehow in order to correct the anomaly. How this should be accomplished is an open question.

the orientation-sensor

The orientation-sensor *seems* to be useful. However, as it has its main application in road-tracking, which has not been implemented, a complete evaluation is not yet possible.

the textural-sensor

While the texture operations *local variance* and *variance of local orientation* seem to be useful for city discrimination a textural-sensor should be able to choose *one* algorithm. At present, the number of tests that have been made, is not large enough to say when one algorithm is usable and when it is not. The kinds of tests necessary require that some kind of sensor on the mid-level is at least partially implemented.

the system outline

As I have gained too little practical experience of AI systems in remote sensing, I have left the system outline incomplete on purpose. Few design specific matters are actually decided, but using the system hierarchy split in three levels should ease a future implementation. It ought to be obvious when reading the report what each level should do and not do.

Chapter 5

Summary

This chapter will summarize the results of this thesis work. It will also summarize what has been said in the report about how the work should be continued.

5.1 Conducted work

This section will summarize what has been done during this masters thesis work.

detection systems

The general principles of existing man-made object detection systems today have been analysed and described. A special section is devoted to an overview of road-extraction algorithms.

system outline

A system outline for detection of man-made objects is presented in this report. The system is based on the three levels of vision used in Computer Vision. The two main branches of the system are responsible for detection of cities and roads respectively. The kinds of object properties considered are *geometrical*, *radiometric*, and *spatial* (inter object relations). In the lower levels of the system a logical-sensor system modelling is used to achieve data source flexibility and modularity within the system. Please note that the presented system has **not** been completely implemented.

BDT logical sensor

A BDT (Background Discriminant Transformation) logical sensor has been implemented. The original BDT-algorithm has been improved by normalization of the resultant projections, and a way of automatically finding a training region for the background is proposed. A method for handling scenes containing both water and urban areas is proposed.

orientation logical sensor

A local-orientation logical sensor has been implemented. The principles of how this sensor should be adjusted to handle scenes of different spatial resolutions are discussed.

gradient logical sensor

A filter for gradient magnitude has been implemented. The resultant filter (of *Sobel* type) was found to bee unsatisfactory for extraction of roads and even for road-seeds as is.

textural logical sensor

A logical sensor for texture estimation has been partly implemented. The texture measures found useful were *local variance* and *variance of local orientation*. Both measures have been constructed in an isotropic manner, and they can adapt to different spatial resolutions through one parameter adjustments without output-magnitude alterations.

The textural operations have been found to work best on SPOT band 2 and Landsat band 3. Using several bands for one textural operation has been found to be a waste of time, as the results are either almost identical or not very useful.

5.2 Continuation

This section will summarize the suggestions given in the report to how the work should be continued to produce a working system.

blackboard systems

This report suggests (at least indirectly) that the feasibility of the *Blackboard system* scheme should be investigated before implementing the high-level parts of the system. There is however no hurry in deciding how the high-level vision in a system should work. Especially the Blackboard system scheme is ideal for waiting with the actual system assembly and concentrating on the lower-levels first.

the BDT sensor

The report suggests that the BDT sensor should be modified such that the background-foreground ratio need not be supplied to the sensor.

the orientation sensor

The report suggests that the orientation sensor should be enhanced with multiple filter-sets in order to handle different scene resolutions.

the textural sensor

Details on which textural measures are best, and for which scene-types they should be used ought to be investigated further. If the system should be able to handle many different kinds of scenes an algorithm for input band selection might be needed.

the gradient sensor

In order to assess the Gradient sensor performance one has to implement the Road-detection sensor at least partially.

mid-level vision algorithms

The first thing to do now should be to implement some mid-level vision algorithms, as they ease the low-level sensor assessment. A City-detection sensor, a Road-detection sensor and some geometrical object property measures should be implemented first.

A. Preprocessing of satellite images

This appendix describes in which ways satellite images are processed before they can become input to a feature detection system. It is not needed for the understanding of the rest of this thesis, but is included anyway, as a background, and for increased understanding of what kinds of scenes we are dealing with.

A feature detection system such as the one this thesis is investigating will not deal with the raw signal stream emitted by a satellite. The raw data is modified in three steps before it is usable. First the data is re-sampled to a quadratic grid, then it is geometrically corrected and finally it is converted to a map projection.

scan-line compensation

The initial *scan-line compensation* is needed because of unequal spacing of the samples in the signal stream. The satellites Landsat-4 and 5 for example have alternating left to right and right to left scan-lines that are slightly tilted due to the motion of the satellite (See Figure 26 below). The first re-sampling will cure this.

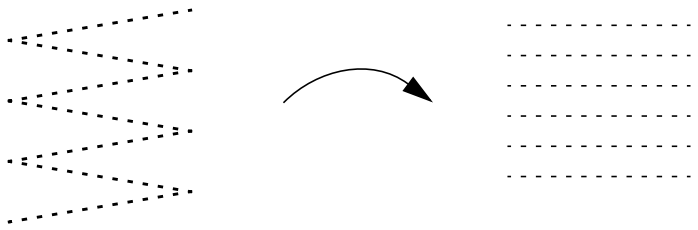

FIGURE 26. Scan-line-conversion

geometrical correction

The *geometrical correction* is needed because the satellite looks at the Earth from a different angle in each pixel (See Figure 27 below). This will cause pixels in different regions to correspond to surface areas of different sizes.

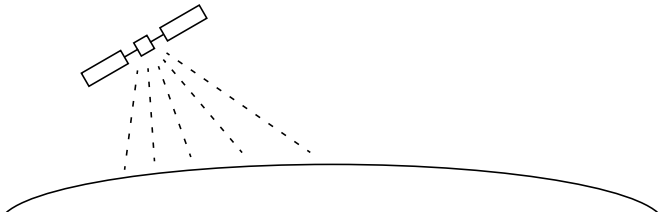

FIGURE 27. Geometrical correction

cartographic projection

The final projection to map coordinates (See Figure 28 below) is needed to achieve a metric in the image. This is because we want to be able to measure distances and area. For this we need a *datum* (a fixed point on the surface which relates the projection to the actual area on the surface) and a model of the Earth's shape (usually some kind of rotation generated elliptic shape). For details on how these *preprocessing steps* are performed the reader is directed to the book "Remote Sensing and Image Interpretation" [20].

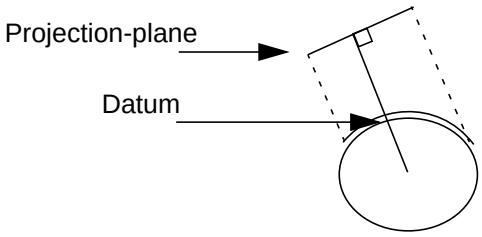

FIGURE 28. Cartographic-projection

The proportions in this figure are strongly exaggerated.

B. Classification

The goal of this thesis is to detect a certain class of objects. By detection of an object is meant classification of pixels as either belonging to the object or not. Due to the approximation inherent in the scene representation (the description of a continuous world by discrete pixels) the classification always involves a compromise. Somewhere a line has to be drawn to separate the set of pixels belonging to the object from those that don't.

Classification is conducted in feature space; the pixels are seen as points in a multi-dimensional space with various object properties on the axes. Figure 29 below illustrates the concept of two-dimensional classification. The points belonging to one class are discriminated from those belonging to another by means of a *discriminant*. In the two dimensional case the discriminant is a line, but if we have more than two properties the discriminant becomes a plane or a hyper-plane.

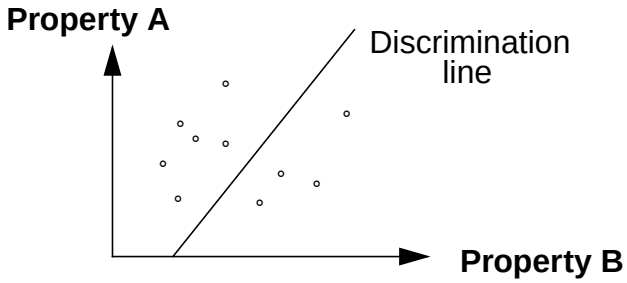

FIGURE 29. Two dimensional classification

discriminant function

Should we want to segment the image in more than two classes, the use of discrimination lines becomes awkward. Instead we now construct one *discriminant function* for each class we want to discern. A discriminant function is typically constructed such that it yields a positive value for those points that belong to the class and negative values for all other points.

the curse of correlation

The more properties of an object we know, the easier it ought to be to discriminate the classes, provided that the interpreter is not “drowned” by all the properties. This can happen if several properties are similar (mathematically *correlated*), and the only difference between them lies in the noise. It is desired that only those properties that are needed for detection, that is *discrimination* of an object, can be selected before we initiate the discrimination.

principal component analysis

One way to reduce the number of properties if several of them are correlated is to apply a PCA- (Principal Component Analysis) transform. The PCA-transform (PCT) is a linear transform of the property-space. PCT will produce new uncorrelated properties, and will maximize the variance of the data in the first few transform components. In this way the number of properties can be reduced, almost without any loss of information, by removing some of the last transform components (See Figure 30 below).

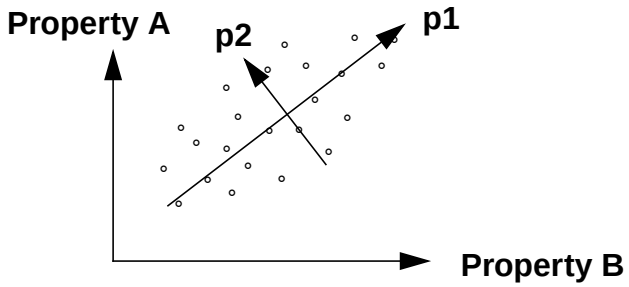

FIGURE 30. PCA example

*PCA finds the direction of maximum variance in a data-set. The principal component above is **p1** and the secondary variance direction is **p2**.*

measure of certainty

Once the decision has been made that a pixel belongs to an object, a good idea would be to have some kind of *measure of certainty* of the classification, as the classification is only an approximation anyway. This measure could, for example be the magnitude of the discriminant function. A measure of certainty is useful in the case of a conflict at a later

stage of detection, as the same pixel might be selected again, but as belonging to another object. A measure of certainty can also be included on a higher level, stating to what certainty an object is present in the scene at all.

representation

When discussing classification it should be noted that ***this is the first stage in the detection process where we actually coerce the scene into pixels***. A scene is always represented by pixels, but before the classification these should be seen merely as sample points, **not** as actual elements in the scene. One should not forget that before we have classified the pixels in a scene we could at any point construct a continuous (or analogue) representation from the pixels. For this reason, the classification should be conducted as late as possible (in those rare cases where there exists an option).

C. Blackboard Systems

This thesis has mainly concerned low-level computer vision. The algorithms discussed have been designed with the higher levels of vision in mind though. In high-level computer vision one does not usually work with individual pixels but with the clusters of pixels that have been grouped into objects by the lower level algorithms. One of the approaches to object organizing systems that exist today is the *Blackboard System*.

A Blackboard System (see Figure 31 below) consists of a number of information processing units known as *knowledge sources* (KS) or specialists [4]. The KS communicate via a blackboard. The blackboard is a place where the KS fetch their information and place their results. In this way all KS contribute to the solution of the problem, but only when they are actually needed.

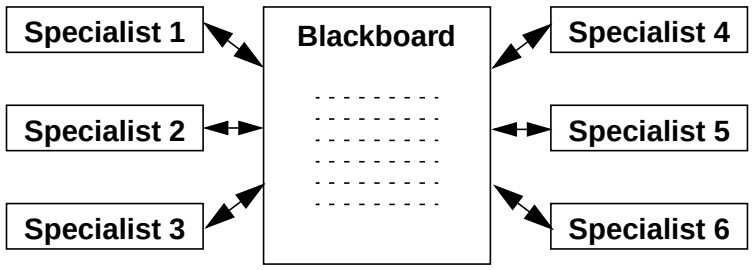

FIGURE 31. Blackboard System structure

There are a number of features that make blackboard systems quite ample for producing descriptions of complex scenes [4]:

- The specialists are working independently. They do not know anything about each others ways of solving problems. They simply fetch data from the blackboard and produce results. This makes specialists easy to design, and to add or remove from the system.

- There is no coercion of the ways individual KS reason. One KS may use forward chaining, an other backward chaining and so on.
- The blackboard model does not enforce any information representation, the specialists may use any kind of language (as long as they understand each other). In practice there is a trade-off between a very expressive representation that requires complex specialists and a more simple representation with simple specialists.

Typically objects will have an associated pixel-region and some structural attributes in a *frame* like manner (See “Computer Vision” on page 4). As the scene interpretation proceeds, objects will be joined, discarded, shrunk or expanded depending on how they are related to other objects.

D. Earth monitoring satellites

This appendix contains lists of features of sensors aboard some Earth-monitoring satellites currently in orbit. The numbers come from the book “Remote sensing and Image Interpretation” [20] and from various documents on the ESA Internet site [7].

The sensors to be described operate from satellites within the following programs:

- **Landsat**
  This is NASA´s Earth monitoring satellite program. Satellites currently in orbit are Landsat-4 and Landsat-5. Both orbit the Earth at an altitude of 705 km completing just over 14.5 orbits each day. These are equipped with sensors named **MSS** (Multi Spectral Scanner) and **TM** (Thematic Mapper). A new satellite with an improved **TM** sensor, **ETM+** is scheduled for launch in July 1998. The name of this satellite will be Landsat 7 [17].
- **SPOT** (Systeme Pour l'Observation de la Terre)
  This was originally a French Earth observing program, but now Sweden and Belgium are also participating. The satellites currently in orbit are SPOT-1, 2 and 3. All have the same types of sensors. The SPOT satellites orbit at an altitude of 830 km, and make almost 14.2 orbits each day. (SPOT-3 has malfunctioned since an accident the 14:th of November 1996.) Each of these satellites have two identical sensors called **HRV** (High Resolution Visible). These can operate in either panchromatic or multispectral mode. A new satellite, SPOT 4, with improved sensors will be launched in March 1998 [22].
- **NOAA** (National Oceanic and Atmospheric Administration)
  NOAA is an American meteorological satellite program. One sensor of interest in Earth monitoring and cartogra-

phy is the **AVHRR** (Advanced Very High Resolution Radiometer) incorporated on the NOAA-6 through NOAA-12 satellites.

- **ERS** (European Remote sensing Satellite)
  This is a satellite program administered by ESA (European Space Agency). The satellite currently in orbit is named ERS-2. (ERS-1 was operational 1991 to 1995). The sensor of interest in cartography is AMI (Active Microwave Instrument), a SAR-type (Synthetic Aperture Radar) sensor.
- **NASDA** (The National Space Development Agency of Japan)
  NASDA currently has three Earth-observing satellites in orbit. These are JERS-1 (launched February 1992), ADEOS (launched August 1996). JERS-1 features one **SAR** and two **OPS** (Optical Sensor) sensors [24]. The sensor of interest in ADEOS is called **AVNIR**.
- **Radarsat**
  Radarsat is a remote sensing satellite operated by the Canadian Space Agency. It orbits the Earth at an altitude of 798 km with 14.3 revolutions each day.
- **IRS** (Indian Resource Satellite)
  IRS is a series of Indian satellites launched for NRSA (National Remote Sensing Agency of India). The series include the IRS-IA / lB, IRS-lC and IRS P2 / P3 missions.

TABLE 1. MSS (Multi Spectral Scanner)

band	range (μm)	resolution (side of square in meters)
1	0.5–0.6	82
2	0.6–0.7	82
3	0.7–0.8	82
4	0.8–1.1	82

MSS is a sensor aboard the Landsat satellites. It has four bands with a resolution of 82 m in 64 (6 bit) intensity levels. MSS images are most useful for large area analysis, such as geologic mapping.

TABLE 2. TM (Thematic Mapper)

band	range (μm)	resolution (side of square in meters)
1	0.45–0.52	30
2	0.52–0.60	30
3	0.63–0.69	30
4	0.76–0.90	30
5	1.55–1.75	30
6	10.4–12.5	120
7	2.08–2.35	30

TM is a sensor aboard the Landsat satellites. It has seven bands with resolutions of either 30 or 120 m in 256 (8 bit) intensity levels. Most receiving systems are however able to improve the resolution to 28.5 m after geometric correction of the data. TM images are useful for a wide range of applications, such as the road and populated area detection this thesis concerns.

TABLE 3. The SPOT HRV sensor

band	range (μm)	resolution (side of square in meters)
1	0.50–0.59	20
2	0.61–0.68	20
3	0.79–0.89	20
4	0.51–0.73	10

SPOT HRV (High Resolution Visible) sensor: Bands 1 to 3 are used in multispectral mode. Band 4 is used in panchromatic mode. The sensors have 256 (8 bit) intensity levels.

TABLE 4. AVHRR on board NOAA-6, -8, -10, and -12

band	range (μm)	resolution (side of square in meters)
1	0.58–0.68	1100 / 4000
2	0.72–1.10	1100 / 4000
3	3.55–3.93	1100 / 4000
4	10.5–11.5	1100 / 4000
5	band 4 repeat	band 4 repeat

AVHRR (Advanced Very High Resolution Radiometer): Resolution 1.1 km (LAC – Local Area Cover) near nadir (satellite ground trace) 4 km at equal sampling of whole image (GAC – Global Area Cover).

TABLE 5. AVHRR on board NOAA–7, –9, and –11

band	range (μm)	resolution (side of square in meters)
1	0.58–0.68	1100 / 4000
2	0.72–1.10	1100 / 4000
3	3.55–3.93	1100 / 4000
4	10.3–11.3	1100 / 4000
5	11.5–12.5	1100 / 4000

AVHRR (Advanced Very High Resolution Radiometer) continued.

TABLE 6. AMI (Active Microwave Instrumentation) in IMAGE mode

band	range (mm)	resolution (side of square in meters)
C–band	37.5–75	30

AMI (Active Microwave Instrumentation) on-board ERS-2 is operating in the C-band (radio wavelengths).

TABLE 7. SAR sensor on board JERS-1

band	range (cm)	resolution (side of square in meters)
L	15–30	18

The SAR (Synthetic Aperture Radar) sensor on JERS-1. This sensor has its main applications in snow-cover analysis and high-resolution cartography.

TABLE 8. OPS sensor on board JERS-1

band	range (μm)	resolution (in meters)
1	0.52–0.60	18x24
2	0.63–0.69	18x24
3	0.76–0.86	18x24
4	0.76–0.86	18x24
5	1.60–1.71	18x24
6	2.01–2.12	18x24
7	2.13–2.25	18x24
8	2.27–2.40	18x24

Optical sensors on-board JERS-1 ranging from visible (bands 1 to 4) to short wave infrared (bands 5 to 8). Bands 3 and 4 have the same range, and are used for stereo imaging. The application of this sensor is high-resolution cartography.

TABLE 9. The ADEOS AVNIR instrument.

band	range (μm)	resolution (side of square in meters)
1	0.42–0.50	16
2	0.52–0.60	16
3	0.61–0.69	16
4	0.52–0.69	8
5	0.76–0.89	16

The ADEOS AVNIR (Advanced Visible and Near Infra-red Radiometer) optical sensor. Bands 1, 2, 3 and 5 are used in the multi-band mode. Band 4 is used in the panchromatic mode. Band 5 is in the near-infrared range. Applications are high resolution land and coastal zone imaging.

TABLE 10. The Radarsat sensor.

mode	range (mm)	resolution (in meters)
Standard	37.5–75 (C band)	25x28
Wide swath 1	37.5–75 (C band)	48–30x28
Wide swath 2	37.5–75 (C band)	32–25x28
Fine	37.5–75 (C band)	11–9x9
Narrow scan	37.5–75 (C band)	50x50
Wide scan	37.5–75 (C band)	100x100
Extended high	37.5–75 (C band)	22–19x28
Extended low	37.5–75 (C band)	63–28x28

The Canadian Radarsat sensor in various modes of operation. Applications are ice reconnaissance and cartography.

References

[1] Sylvain Airault et al. *Road detection from aerial images: a cooperation between local and global methods*. SPIE Vol 2315

[2] Laurent Alquier et al. *Perceptual Grouping and Active contour functions for extraction of roads in satellite pictures.* September 1996. From a conference titled: Image and Signal Processing. SPIE Vol. 2955

[3] Véronique Clément et al. *Interpretation of remotely sensed image in a context of multisensor fusion using a Multi-spectralist architecture.* Unite de Recherche INRIA-Sophia Antipolis, France October 1992

[4] Daniel D. Corkill. *Blackboard Systems.* Article in AI Expert 6(9) pp 40–47 September 1991

[5] Michele Covell. *Autocorrespondence: Feature-based Match Estimation and ImageMetamorphosis.* Proc. IEEE International Conference on Systems, Man and Cybernetics.1995
Available online at:
http://www.interval.com/papers/smc95/smc95.html

[6] Robert S. Engelmore and Anthony Morgan ed. *Blackboard Systems.* Addison-Wesley 1988. ISBN 0-201-17431-6

[7] The ESA site.
http://gds.esrin.esa.it/

[8] Donald Geman and Bruno Jedynak. *Detection of Roads in Satellite Images.* Proceedings of the International Geoscience and Remote Sensing Conference 1991

[9] Ram Gnanadesikan. *Methods for Statistical Data Analysis of Multivariate Observations.* John Wiley & Sons. New York 1977 ISBN 0-471-30845-5

[10] Gösta H. Granlund and Hans Knutsson. *Signal processing for Computer Vision,* 1995 Kluwer Academic Publishers.

[11] Armin Gruen et al. *Linear Feature Extraction with Dynamic Programming and Globally Enforced Least Squares Matching.* pp 8–94 in *Automatic Extraction of Man-Made Objects from Aerial and Space Images.* Monte Verità 1995. Birkhäuser Verlag Basel. ISBN 3-7643-5264-7.

[12] T. Henderson et al. *A Fault Tolerant Sensor Scheme.*Proceedings of the seventh International Conference on Pattern Recognition. July 1984. pp 663–665

[13] Stéphane Houzelle and Gérard Giraudon. *Contribution to multisensor fusion formalization.* INRIA Sophia Antipolis, France 1994. "Robotics and Autonomous Systems" 13 (1994) pp 69–85.

[14] Bruno Jedynak et al. *Tracking Roads in Satellite Images by Playing Twenty Questions.* pp 243–253 in *Automatic Extraction of Man-Made Objects from Aerial and Space Images.* Monte Verità 1995. Birkhäuser Verlag Basel. ISBN 3-7643-5264-7.

[15] Bernd Jähne. *Digital Image Processing.* Third edition. Berlin 1995 Springer Verlag. ISBN 3-540-59298-9

[16] Bart Kosko. *Fuzzy Thinking.* HarperCollins Publishers 1994. ISBN 0 00 654713 3

[17] The site of the Landsat Program. http://geo.arc.nasa.gov/sge/landsat/landsat.html

[18] Gunnar Larsson-Leander. *Astronomi och Astrofysik 2:a upplagan.* Falköping, Sweden 1977. Liber Läromedel. ISBN 91-23-71074-8

[19] Frédéric Leymarie et al. *Towards the Automation of Road networks Extraction processes.* France 1996. SPIE Vol 2960.

[20] Thomas M. Lillesand and Ralph W. Kiefer. *Remote Sensing and Image Interpretation 3:rd ed,* New York 1994. Wiley & Sons Inc.

[21] D.M. McKeown et al. *Cooperative methods for road tracking in aerial imagery*. Proceedings of Computer Vision and Pattern Recognition June 5–9 1988.

[22] The Mission Management Office site. http://sscapws2.atsc.allied.com/803/SPOT.html

[23] NASA's introduction to remote sensing. http://code935.gsfc.nasa.gov/Tutorial/TofC/toc1.html

[24] The NASDA site. http://yyy.tksc.nasda.go.jp/Home/This/thisindex_e.html

[25] W. Neuenschwander et al. *From Ziplock Snakes to Velcro Surfaces.* pp 105–114 in *Automatic Extraction of Man-Made Objects from Aerial and Space Images.* Monte Verità 1995. Birkhäuser Verlag Basel. ISBN 3-7643-5264-7.

[26] William H. Press et al. *Numerical Recipes in C.* Second edition Cambridge University Press 1992. ISBN 0-521-43108-5

[27] V. K. Shettigara. *Image enhancement using background discriminant transformation.* Int. J. of Remote Sensing 1991, vol 12 no. 10. pp 2153–2167

[28] V. K. Shettigara et al. *Semi-Automatic Detection and Extraction of Man-Made Objects in Multispectral Aerial and Satellite Images*. pp 63–72 in *Automatic Extraction of Man-Made Objects from Aerial and Space Images*. Monte Verità 1995. Birkhäuser Verlag Basel ISBN 3-7643-5264-7.

[29] Mou Tsung-san. *Lectures on Kant's work "Critique of Pure Reason"*. http://www.nevada.edu/home/15/chancl/html/pre.html lecture 14.

[30] A. Zlotnick et al. *Finding Road Seeds in Aerial Images*. Image Understanding Vol 57. No. 2. March 1993 pp. 243–260.

Index

S

T

U

V

W

Avdelning, Institution
Division, department

Department of Electrical Engineering
Computer Vision

Datum
Date

1997-12-17

Språk
Language

☐ Svenska/Swedish
☒ Engelska/English
☐ _____

Rapporttyp
Report: category

☐ Licentiatavhandling
☒ Examensarbete
☐ C-uppsats
☐ D-uppsats
☐ Övrig rapport
☐ _____

ISBN

ISRN

Serietitel och serienummer
Title of series, numbering

ISSN

LiTH-ISY-EX- 1852

URL för elektronisk version

Titel
Title

Detection of Man-made Objects in Satellite Images

Författare
Author

Per-Erik Forssén

Sammanfattning
Abstract

In this report, the principles of man-made object detection in satellite images is investigated. An overview of terminology and of how the detection problem is usually solved today is given. A three level system to solve the detection problem is proposed. The main branches of this system handle road, and city detection respectively. To achieve data source flexibility, the *Logical Sensor* notion is used to model the low level system components. Three *Logical Sensors* have been implemented and tested on Landsat TM and SPOT XS scenes. These are: BDT (Background Discriminant Transformation) to construct a man-made object property field; Local-orientation for texture estimation and road tracking; Texture estimation using *local variance* and *variance of local orientation*. A gradient magnitude measure for road seed generation has also been tested.

Nyckelord
Keywords

remote sensing, computer vision, logical sensors, road detection, man-made objects, spectral signature, local variance, quadrature filter