



**Airborne minefield detection: pilot
project
RMA/SIC Final report
First draft**

Yann Yvinec
Pascal Druyts
Vinciane Lacroix
Youssef Ouaghli
Idrissa Mahamadou

Acknowledgement

This research project formed part of the European Pilot Project entitled *Airborne Minefield Detection: Pilot Project*. This project was funded by the European Commission, the governments of Belgium, Germany, Luxembourg, Portugal, the United-Kingdom and the International Institute for Aerospace Survey and Earth Sciences (ITC), The Netherlands. The project consisted of two phases: a field trial in Belgium and a test over several areas in Mozambique.

The project partners involved in both phases are: CAE Aviation (L), EOS (UK), Eurosense (B), VUB (B), Geograf (P), RMA (B), SSC (S) and ZEO (G). Project partners involved mainly in the first phase are: Aerodata (B), Aerosensing (G), IGI (G), Recon Optical (UK). Project partners involved in the second phase are: CND (Moz), NPA (N). The project was co-ordinated and managed by ITC.

Contents

1	Introduction	7
2	Semi-automatic image processing tools for minefield detection	9
3	Selection of minefield indicators	11
4	Detection of anti-tank mines on the ground	13
4.1	The model of mine used	13
4.2	Principle of the detection	14
4.3	Contrast estimation	14
4.4	Threshold estimation	15
4.5	Detection of partially occluded circles	22
4.6	Evaluation	22
5	Detection of poles	27
5.1	Introduction	27
5.2	Detection of individual poles	27
5.3	Detection of alignment of poles	31
5.4	Conclusion	31
6	Detection of elongated areas	33
6.1	Introduction	33
6.2	Segmentation	34
6.3	Contour-based detector for elongated areas	37
6.4	Skeleton-based detector for elongated areas	42
6.5	Conclusion	46
7	Detection of trenches and foxholes	47
7.1	Introduction	47
7.2	Rectification	48
7.3	Matching	48
7.4	Reconstruction	49
7.5	Conclusion	51
8	Analysis of satellite imagery	53
8.1	Introduction	53
8.2	Extraction of water areas	53

8.2.1	Histogram analysis	53
8.2.2	Classification methods	58
8.3	Detection of bright spots	59
8.4	Conclusion	59
9	Compression	63
9.1	Motivation	63
9.2	The method	64
9.2.1	Principle	64
9.2.2	Test Images	65
9.2.3	Zero Order Entropy	66
9.2.4	Lossless and Lossy compression	68
9.2.5	Improving lossless compression using spectral de-correlation	72
9.3	Influence of scanning on compression	80
9.4	Pole detector on compressed image	82
9.5	Conclusion	86
10	Graphical user interface	87
11	Recommendations	97
11.1	How to choose the indicators	97
11.2	How to build the detectors	97
11.3	How to collect the data	97

Chapter 1

Introduction

Mine clearing is a long and dangerous task. A lot of time is lost clearing areas free of mines that were selected just to be on the safe side [1]. Deminers are looking forward to methods that can reduce the suspected areas.

Researchers are now considering the possibility to perform this area reduction by means of airborne surveys [6, 7]. Human photo-interpreters are, to date, unchallenged in the analysis of these images. Unfortunately in order to survey all suspected areas of a country and have the resolution needed to detect relevant minefield indicators, a tremendous amount of data is to be collected. This makes the analysis a very time-consuming task.

Image processing, by speeding up some repetitive tasks or pointing some regions of interest to the analyst, could be useful.

The project *Airborne minefield detection: pilot project* was launched to study the feasibility of detecting the boundaries of minefields by airborne survey. A first campaign took place in Belgium for testing and in the final campaign, in Mozambique, an aircraft equipped with several sensors has been flown over suspected minefields in Buzi, Bandua, Mameme and Songo [7].

As part of the Royal Military Academy, a partner of this project, the Signal and Image Centre (RMA/SIC) was involved in image processing and especially in investigating what image processing could bring to airborne minefield detection. RMA/SIC has focused on the possibility to improve and speed up the image interpretation by means of image processing tools. This report presents the global work of RMA/SIC in the scope of this project. More details on the participation of RMA/SIC in the first phase of tests in Belgium can be found in [2].

Image processing tools can be used in two distinct ways to speed up the interpretation. First, it can be used to help the detection by selecting regions of interest that would be afterwards studied by a photo-interpreter. Second, image processing can provide the photo-interpreter with enhanced images to help the interpretation. In the first case, the photo-interpreter is given some areas to study where there might be something of interest. In the second the photo-interpreter is given new information, or information displayed in a new way, to improve the understanding of a given zone and help the classification as useful minefield indicator or false alarm. In both cases, some minefield indicators must be defined.

The work of RMA/SIC here involves two steps. First, some indicators have been selected for their relevances in minefield detection as well as for their easiness to be automatically detected on the available images. Then detectors devoted to each of these indicators have

been implemented.

In the **next chapter**, we describe how image processing should be used in this framework.

In **chapter 3**, the proper way to selected minefield indicators is discussed.

Then we present some indicators we developed: an anti-tank mine indicator in **chapter 4**, a pole detector in **chapter 5**, a detector of elongated minefields in **chapter 6** and the use of stereovision in **chapter 7**.

Satellite imagery was also used in the scope of this project. **Chapter 8** presents how image processing may be used with these images.

The goal of this project was to study the potential of airborne surveys and remote sensing for the detection of minefield boundaries. Even if this approach proves to be efficient, we are still faced with the huge amount of data (especially because of the high resolution needed). Therefore the use of compression tools is clearly indicated. In **chapter 9** the compression ratio reached as well as the effect of the image degradation are presented.

In an operational context, the semi-automatic tools developed have to be integrated in a Graphical User Interface (GUI). A presentation of what a Graphical User Interface dedicated to image processing is opportune. Our own GUI enhanced with features dedicated to this project is described in **chapter 10**.

Finally, some recommendations for developing and using image processing tools in future projects of airborne minefield detection are given in **chapter 11**. They concern the constraints that minefield indicators and indicator detectors must fulfill in order to be able to be used in such a semi-automatic procedure. Other constraints about the images and the way to collect them to improve the detection of indicators are also given.

Chapter 2

Semi-automatic image processing tools for minefield detection

The close analysis of the way a human photo-interpreter works makes it clear that image processing alone could not compete with human beings who use knowledge about history, geography, hydrology and, more generally, a large amount of information not always obviously correlated with the problem. Moreover one part of the job of photo-interpreters is to design a procedure of detection on a case-by-case basis. Such an approach is not easy to implement in computer algorithms. Even techniques such as artificial intelligence or expert systems may lack the adaptability of a human being.

Therefore we argue that there are two promising uses of image processing. Image processing can speed up the process by detecting minefield indicators to be defined by the photo-interpreters together with specialists in image processing. How to use these indicators, once detected, to infer the presence of a minefield is left to photo-interpreters. It can also be used to enhance regions of interest provided by either other image processing tools or a human being.

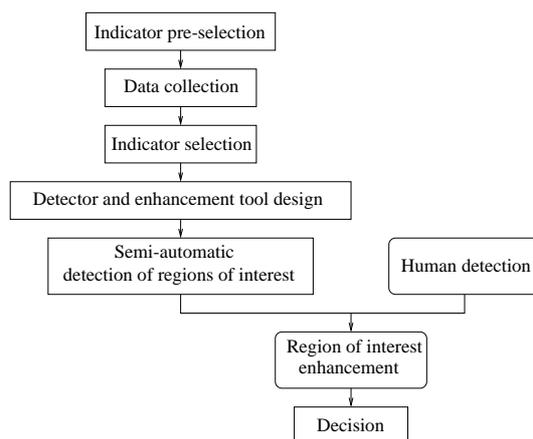


Figure 2.1: Semi-automatics scheme

Chapter 3

Selection of minefield indicators

Minefield indicators must be relevant, that is, their presence must significantly increase the probability to find a minefield in the neighbourhood of that indicator. For that reason, minefield indicators must be provided by photo-interpreters after an investigation in the country and with the help of local people or professional deminers. But these indicators must also be detectable by image processing. The ideal indicator candidate is simple: a model must be easily designed from the original description. Since the detector should be as general as possible, the model and therefore the indicator must be as independent as possible from the environment. For instance if a soil disturbance is an indicator of the presence of a mine in the ground, then a buried mine should create more or less the same kind of disturbances as often as possible. Otherwise a detector of soil disturbances is of little help: it may detect natural disturbances and not react to a buried mine

This report deals with the following indicators used with visible and colour infrared images:

- anti-tank mines on the ground (chapter 4)
- poles indicating the limitations of minefields and their alignments (chapter 5)
- elongated areas since in the zones of Mozambique we studied, minefields have often this shape (chapter 6)
- trenches and foxholes (chapter 7)

In the next chapters, detectors developed for each of these indicators will be described and the obtained results will be presented and commented.

Chapter 4

Detection of anti-tank mines on the ground

4.1 The model of mine used



Figure 4.1: Mine used to tune the algorithms

As pointed out by many partners, where mines are laid, some vegetation anomalies (e.g. missing vegetation) can be seen. Unfortunately, a huge amount of such regions exists and to reach a reasonable false alarm rate, complementary mine characteristics (specific patterns, signature in different spectral bands, etc) have to be introduced in the model. Therefore the typical spectral response (visible and near infrared) of an anti-tank mine lying on the ground was used. A typical example of such a mine may be found in minefield C. Test minefield C is part of the test minefield designed in Leopoldsburg (Belgium) by SEDEE/DOVO (Dienst voor Opruiming en Vernietiging van Ontploffingstuigen / Service d'Enlèvement et de Destruction

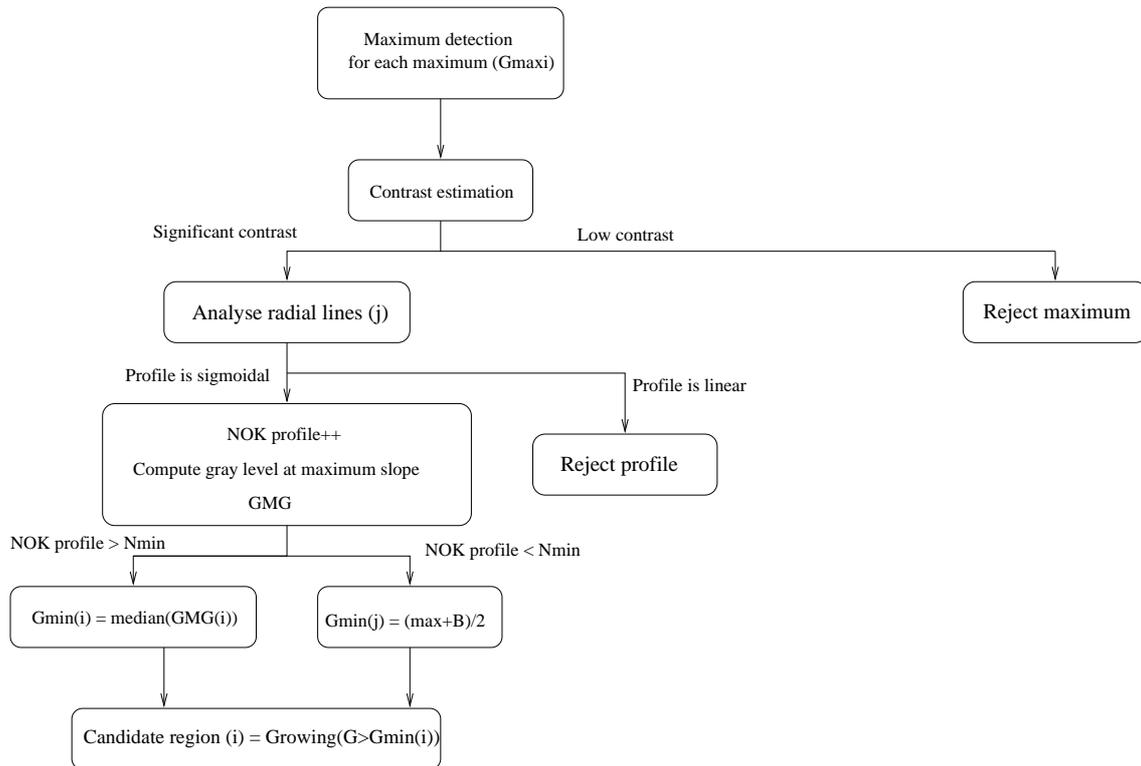


Figure 4.2: Growing for mine detection

des Engins Explosifs), the bomb disposal unit from the Belgian army. Note that Test minefield C was designed to be the training minefield for the project. For that reason all information about it had been made available to our Consortium. An example of a surface anti-tank mine is presented in Figure 4.1.

4.2 Principle of the detection

Mines appear as circular like bright blobs in the blue channel (this channel corresponds to the green colour in the Colour Infrared images). Therefore the detection algorithm first locates the pixel with the largest value (G_{max}) in the blue band. A growing is then performed around those local maxima (see Figure 4.2). The growing algorithm starts from the local maxima and grows the region as long as the candidate pixel grey level is bigger than a given value G_{min} . This approach is classic in image processing and the problem is often to select a good threshold (G_{min}). For this application, we used some knowledge about the searched object to compute automatically the appropriate threshold.

4.3 Contrast estimation

Knowing the approximate size of the searched mines and the resolution of the image, a circle (CB) that is certainly outside the mine (if the local maximum indeed corresponds to a mine)

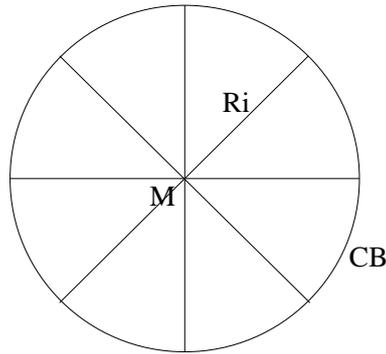


Figure 4.3: Radial lines for the search of the threshold

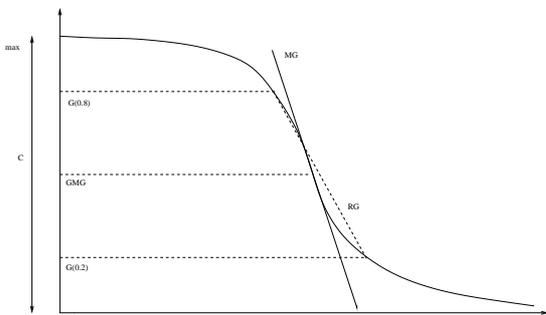


Figure 4.4: Sigmoidal profile

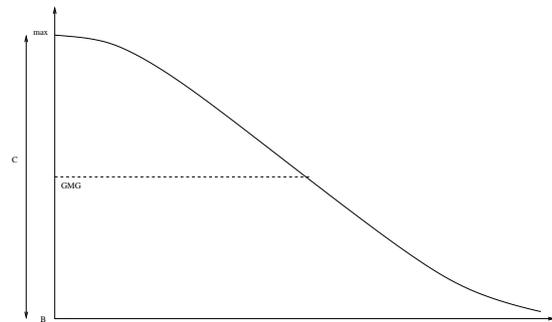


Figure 4.5: Linear profile

may be drawn on the image (see Figure 4.3). It is thus possible to estimate the background (B) value on that circle. This is done by computing the median of the grey level distribution on that circle. The contrast (C) is then defined as the difference between the grey level at local maximum and the background estimated value. If the contrast is too small, the local maximum is rejected and no growing is performed.

4.4 Threshold estimation

If the contrast is large enough, the best threshold is computed by analyzing the grey level profile inside the circle CB . For this, the grey level profile on a number of radial lines (R_i), originating from the local maximum and stopping at the background circle (CB), are considered (see Figure 4.3).

Each of these profiles is classified either as sigmoidal (Figure 4.4) or linear (Figure 4.5). This classification is based on the ratio maximum gradient (MG)/reference gradient. The reference gradient is the slope of the line linking the points $G(0.8)$ and $G(0.2)$ corresponding respectively to a decrease of 0.2 and 0.8 times the local contrast (C). If this ratio is large enough, the profile is well curved and the maximum gradient point may be determined reliably (for linear profiles, GMC becomes very sensitive to noise). For each radial line on which the maximum gradient ratio is large enough (sigmoidal profile), the grey level at the maximum gradient point (GMC) is stored in a table of valid thresholds.

If enough valid thresholds are found (GMC), the median of those thresholds is used as

overall threshold. If an insufficient number of threshold grey levels are found, we deal with a linear grey value profile and the threshold is chosen as $(\max-B)/2$ (see figure 4.5).

Figure 4.6 presents the results of the growing algorithm around mine 281 of minefield C.

Figure 4.7 presents the blue band (on which the growing is performed) of the same area. As can be seen on the window displayed by clicking the mine location (see Figure 4.8), the type of the mine was encoded; it is an anti-tank mine lying on the ground. The mine is well detected but some other regions are also found. To reduce the false alarm rate, geometrical and radiometric attributes (using all available spectral bands) are computed for each mine candidate and a criterion based on those attributes is then used.

The criterion is the following: the candidate must fulfil three conditions.

- The size of the holes in the candidate area should not exceed 20 % of the total area.
- Its area should be between 5 and 200 pixels.
- Its color should be at a mean angular distance smaller than 80 degrees from blue.

Figure 4.9 presents the attributes used and the result after filtering. In the presented region all false alarms were rejected, keeping only the actual mine.

Figure 4.10 presents the mine candidates found using the same approach on a part of the visible image.

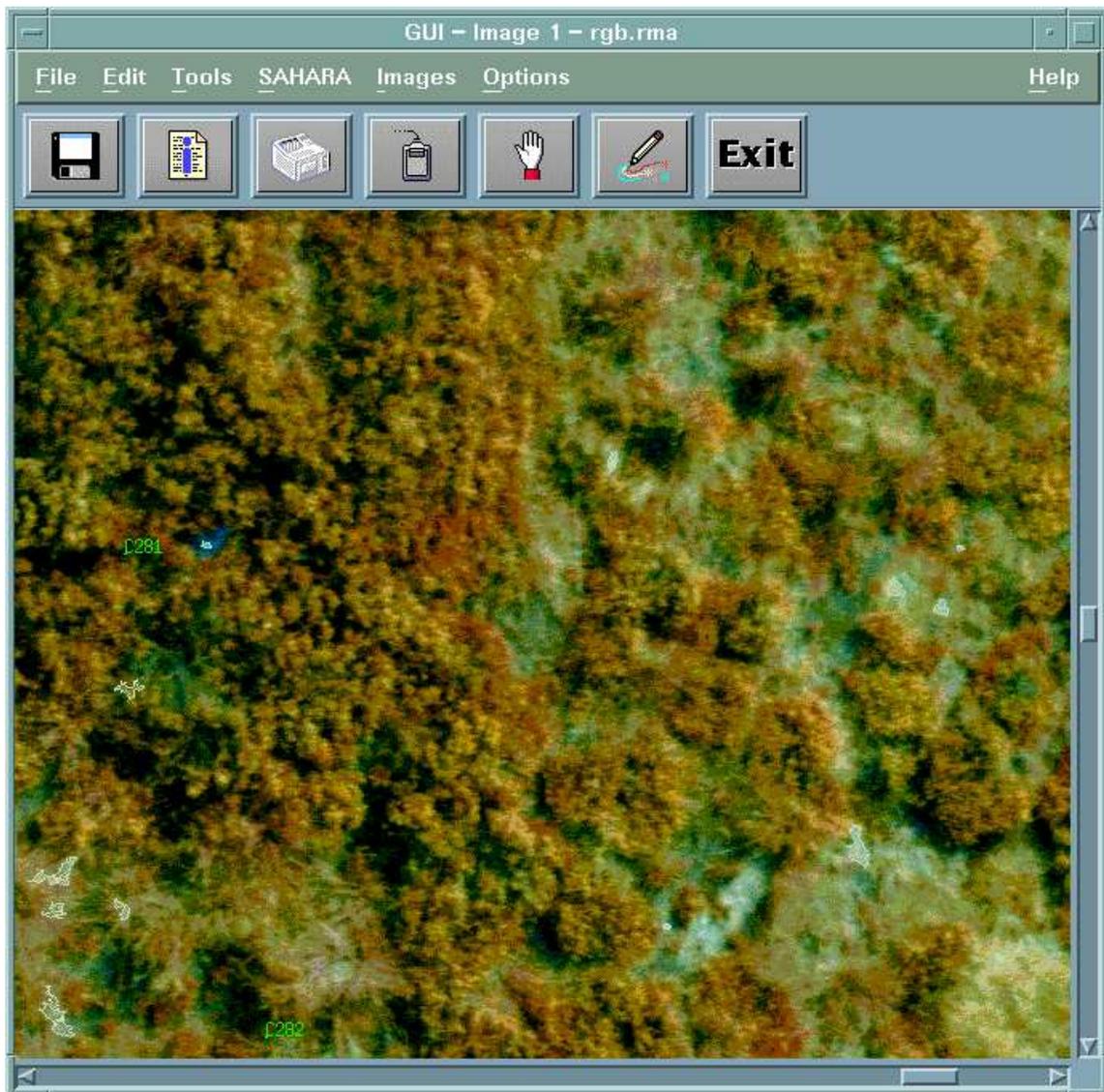


Figure 4.6: Result of growing around mine C281 on Colour Infrared image

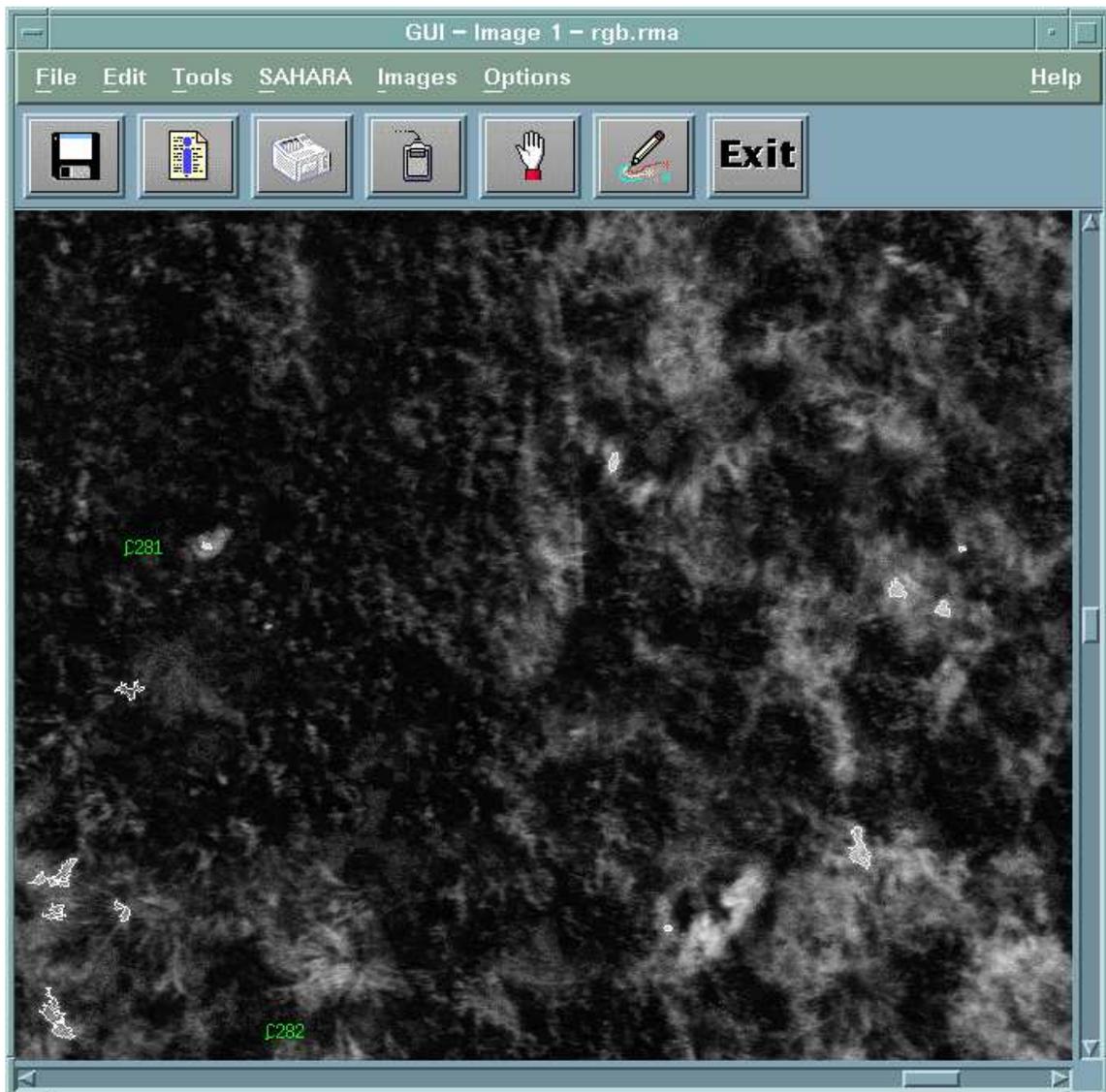


Figure 4.7: Blue band corresponding to Figure 4.6

View / Edit / Add / Delete
Attributes:

Upper Level...
Type: ATK
Name: M6
Position: Horizontal
Depth: On the ground
Ground: Sand/Bush
New Attribute...

Name:

Type:

Quantity:

Data:

Figure 4.8: Mine description

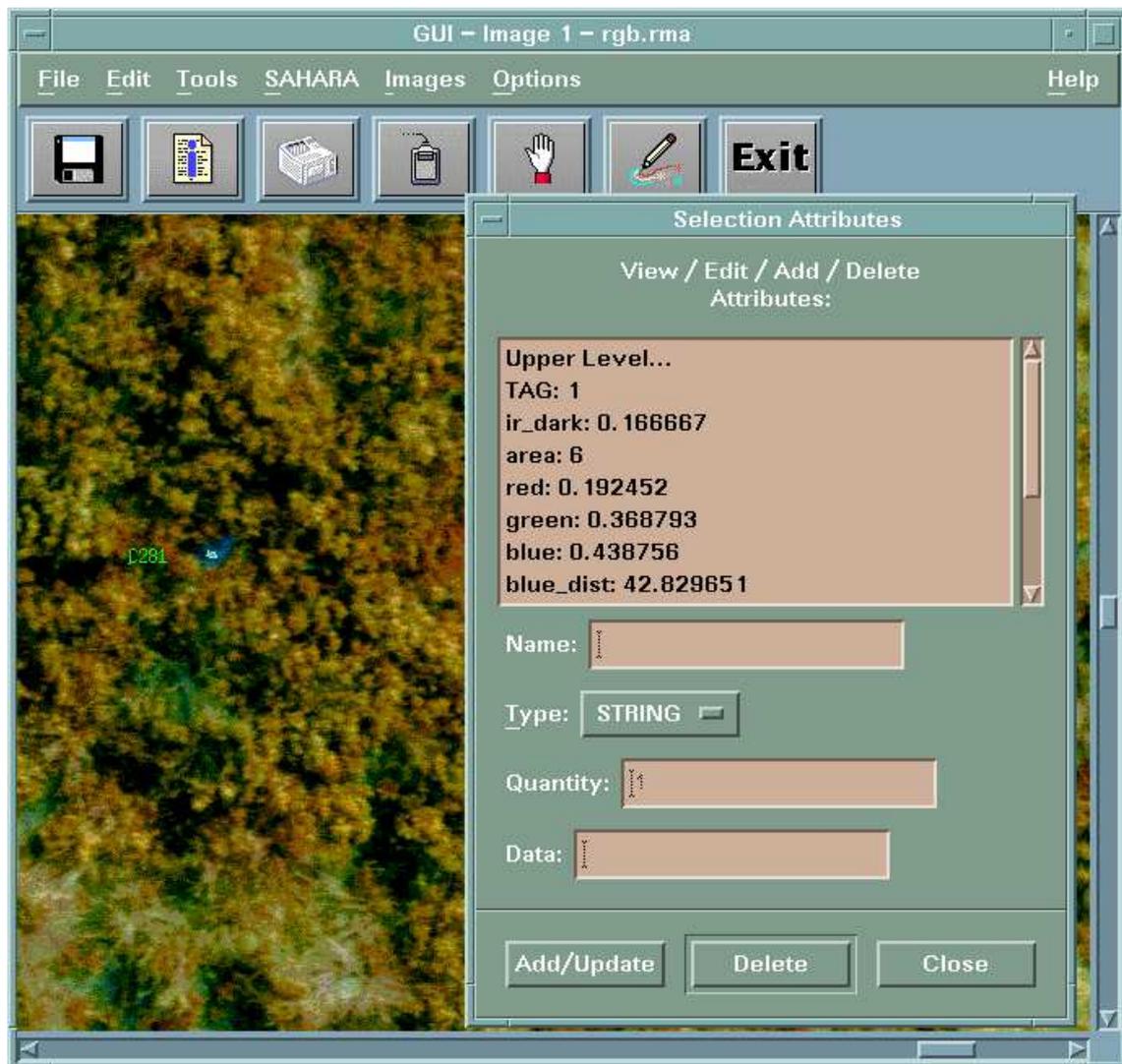


Figure 4.9: Criteria and results after filtering



Figure 4.10: Mine-like objects detected on visible image, with what seems to be a false alarm

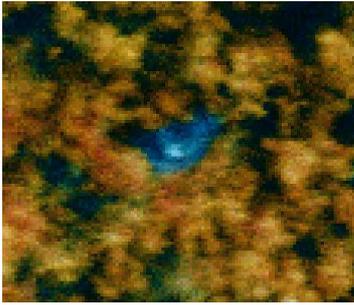


Figure 4.11: Close-up of the mine



Figure 4.12: Contour of the mine

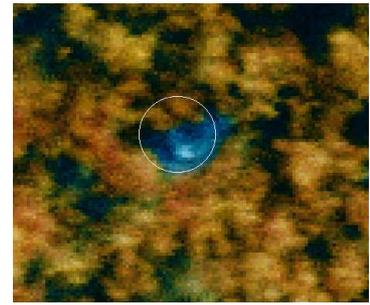


Figure 4.13: Circle estimation of the contour

4.5 Detection of partially occluded circles

The shape of the candidate regions may be used as discriminating feature. Mines lying on the ground present circular shapes. However, the mine is often partially occluded and we developed an algorithm that is able to recognize circular shapes even if they are partially occluded. A measure of the circularity is returned together with an estimation of the radius. The radius provides a better estimation of the size of the object than the visible area (partial occlusion) and is thus more discriminating.

The algorithm estimates a circle that goes through most of the pixels of a given list. Robust statistics are useful here when it is known that the pixels should lie along a part of a circle but some of them can be wrong and very far from the circle [4]. Several triplets of pixels are selected in the list and the circles defined by these three pixels are computed. For each circle, the distances of all the pixels of the lists to this circle and the percentiles of these distances are computed. The circle selected is the one giving the smallest percentiles of errors. The number of triplets is computed so that the probability to have at least one triplet without erroneous pixel, and thus a correct circle, is higher than 0.99.

Figure 4.11 shows a close-up of an anti-tank mine on the ground seen in colour infrared. Only a part of the mine is seen. The blue appearance is due to the fact that the colour green is seen blue in colour infrared.

Figure 4.12 shows the contour of the mine detected. Because of the occluded part, the shape is not circular.

Figure 4.13 shows the circle estimated from the previous contour. It is a good approximation of the position and size of the mine. If an absolute co-ordinate system is available, it is possible to have an approximation of the real size of the mine and then an indication of which type of mine we are dealing with.

4.6 Evaluation

The following images present the result of the automatic mine detection on the full visible and colour infrared images. Note that the size of the visible image is about 400 megabytes and that this image covers a region of about 500 m². The full image was processed in about 30 minutes on a Pentium Pro (200 MHz, 64 megabyte RAM) running under Linux. The position of known mines has been superimposed on both images. (Dots are known mines and crosses are mines proposed by the algorithm).

On the visible image (see Figure 4.14), a V-shaped minefield of anti-tank mines lying on the ground has been detected. Note that no mines of minefield C has been found. The anti-tank mine 281 (anti-tank mine lying on the ground and well visible) was first detected but rejected by the attribute-based filter. Using a better filter criterion, it could be possible to keep that mine without increasing the false alarm rate but little effort was spent for this fine-tuning because we believe that an algorithm that learns from example should be used in practice. This could not be tested because the database contains too few examples. In this context a learning scheme would probably lead to poor results (over-training = learning by heart rules that are only valid for the training set and that may not be extrapolated to new data).

On the colour infrared image that presents minefield C at a higher resolution but with only a small neighbouring region (the V-shaped minefield is not in the image, see Figure 4.15), three anti-tank mines (on and below the surface) of minefield C have been detected. Note that for the buried anti-tank mines, it is a stick lying near the mine that is detected. Even if the stick may be considered as a valid mine indicator, they were found 'by chance' because the algorithm was not developed to find such objects. We concentrated our efforts on the detection of indicators that could be useful in Mozambique and it appears that such sticks are not relevant in Mozambique. As a conclusion, the tool was developed to find anti-tank mine lying on the ground. Most of those mines were detected and the false alarm rate is reasonable. Even two buried mines were found because a stick left in the vicinity of the mines was detected. We believe that the false alarm rate could be reduced by further developments. As an example many false alarms in the colour infrared are found in the trees. A tree detection algorithm could reduce the false alarm rate significantly.

Although limited, the results presented in this chapter have shown that significant process acceleration can be reached by means of image processing. By removing the false alarms found in the trees of the colour infrared image, the found alarms could be grouped in about 10 regions of interest. If the photo-interpreter takes 30 seconds to look at each region of interest, 5 minutes would be needed to analyze the scene. Whereas if a full visual inspection was carried out at full resolution, the photo-interpreters would typically spend at least one minute for a region of 1,000 by 1,000 pixels. 150 such regions have to be analyzed for a full coverage of the colour infrared image leading to about two hours of interpretation. A process acceleration of about 60 may thus be expected. Such a speed-up would be very useful in an operational context where the amount of data to be analyzed would be tremendous. Without image processing and with the same assumptions as above, 650 man-hours would be needed for a complete visual inspection of a scene of 10 km².

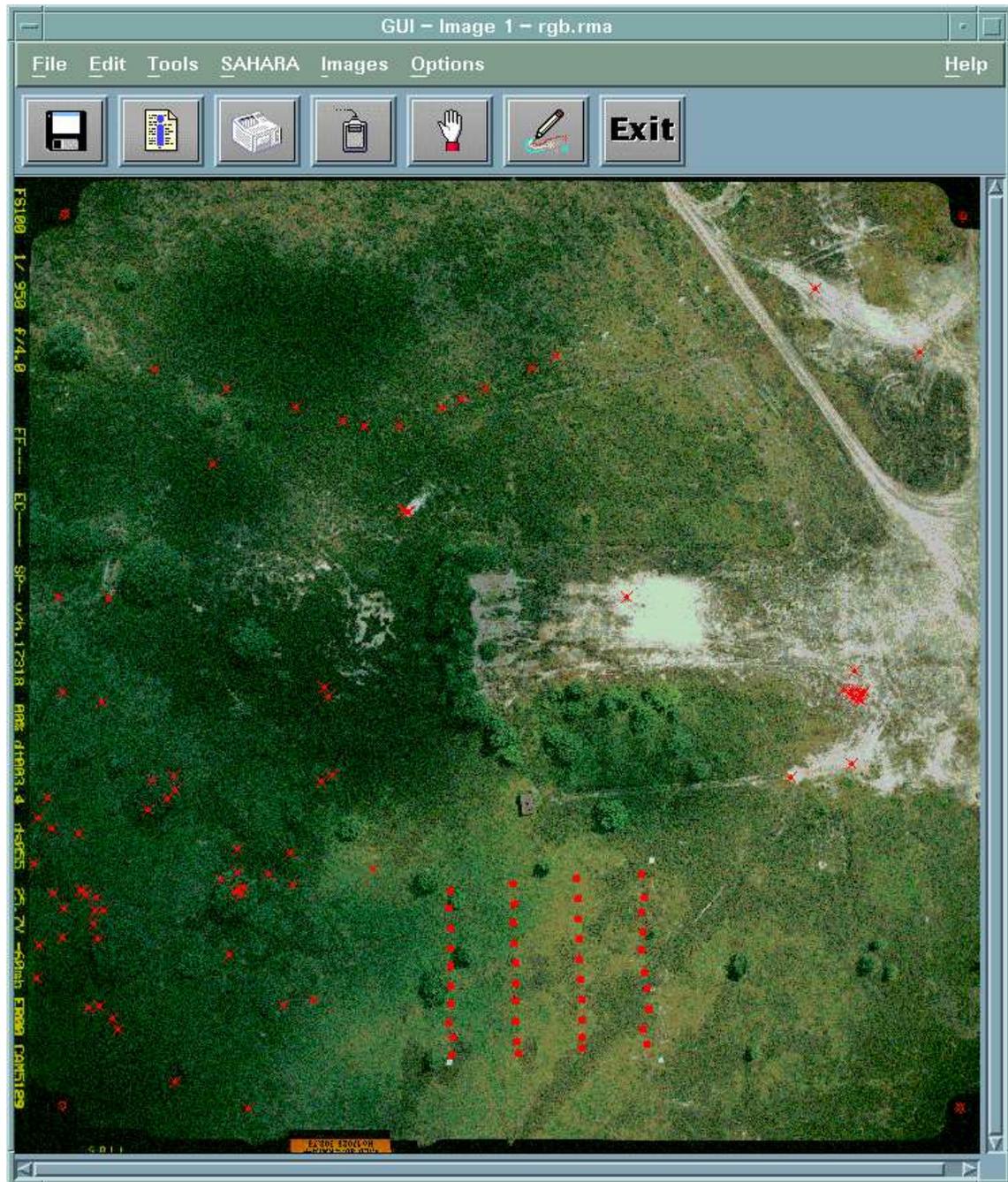


Figure 4.14: Results on a visible image

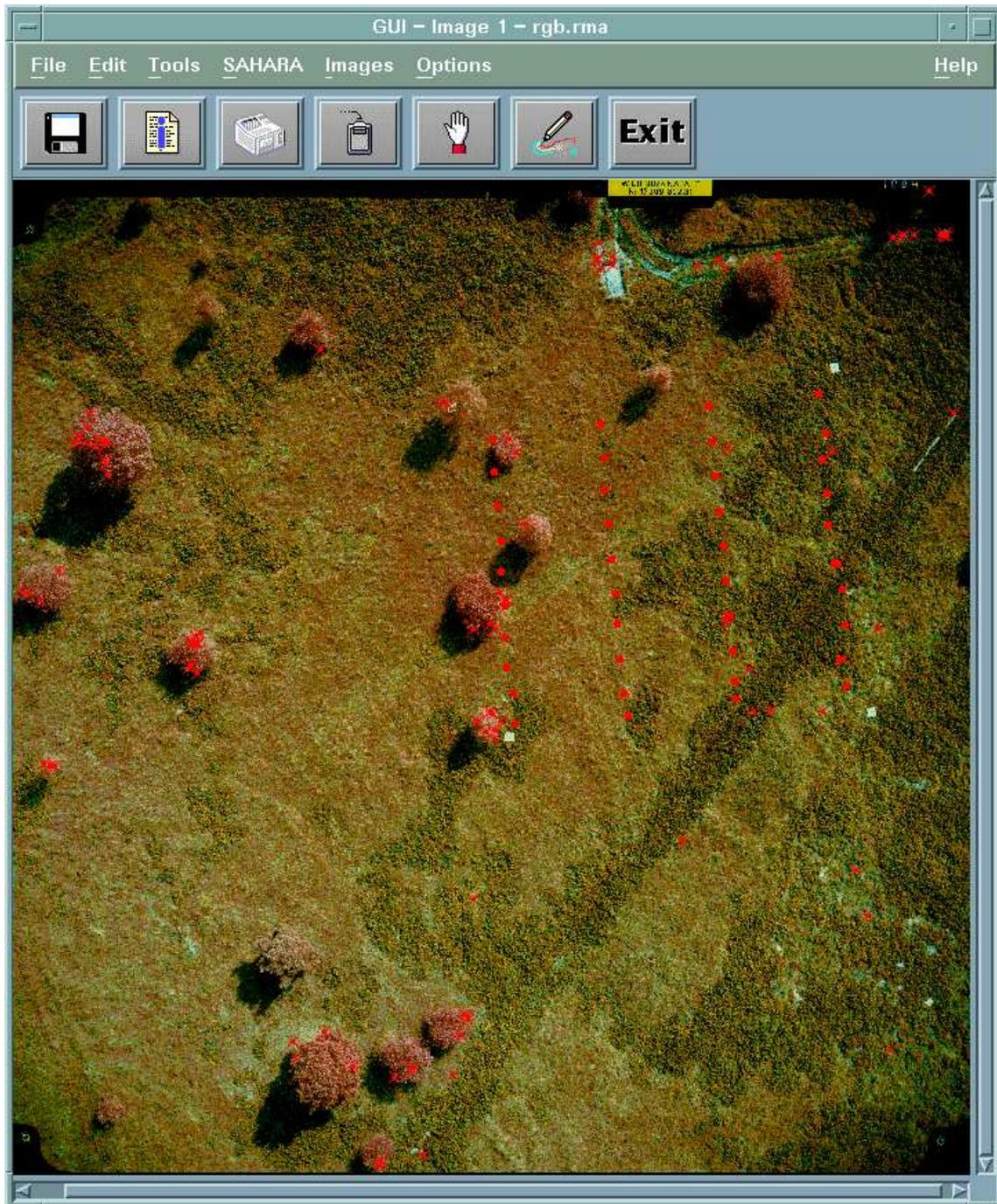


Figure 4.15: Results on a colour infrared image

Chapter 5

Detection of poles

5.1 Introduction

Some minefield in the Buzi area has been laid with a defensive purpose and is clearly marked by poles to warn the local people. These poles may be detected by semi-automatic tools. See Section 5.2. Furthermore these poles are aligned. In Section 5.3, pole alignments are looked for to reduce the number of false alarms.

5.2 Detection of individual poles

A pole as seen on airborne imagery is characterized by a V-shape. One branch is the pole itself and the other is the shadow of the pole. The poles and the shadows are characterized by elongated homogeneous regions that are in general well contrasted with the background. Those regions are limited by two parallel lines. The shadows are always dark and, in the case of the Buzi minefield, so are the poles.

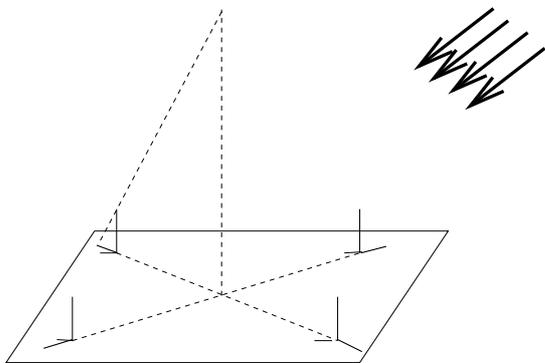


Figure 5.1: Projective projection of poles and shadows

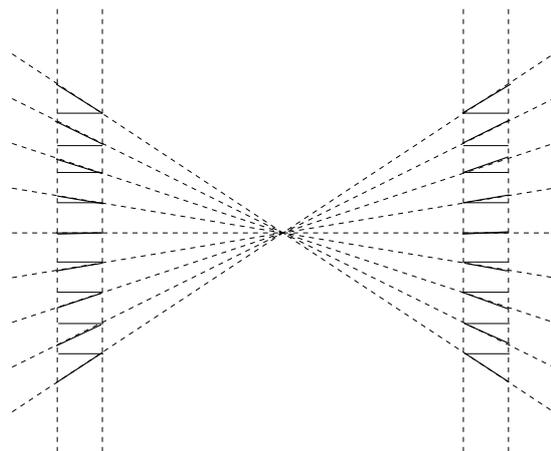


Figure 5.2: Theoretical positions of poles and shadows in an image (the poles are represented with thick lines, the shadows with thin lines)

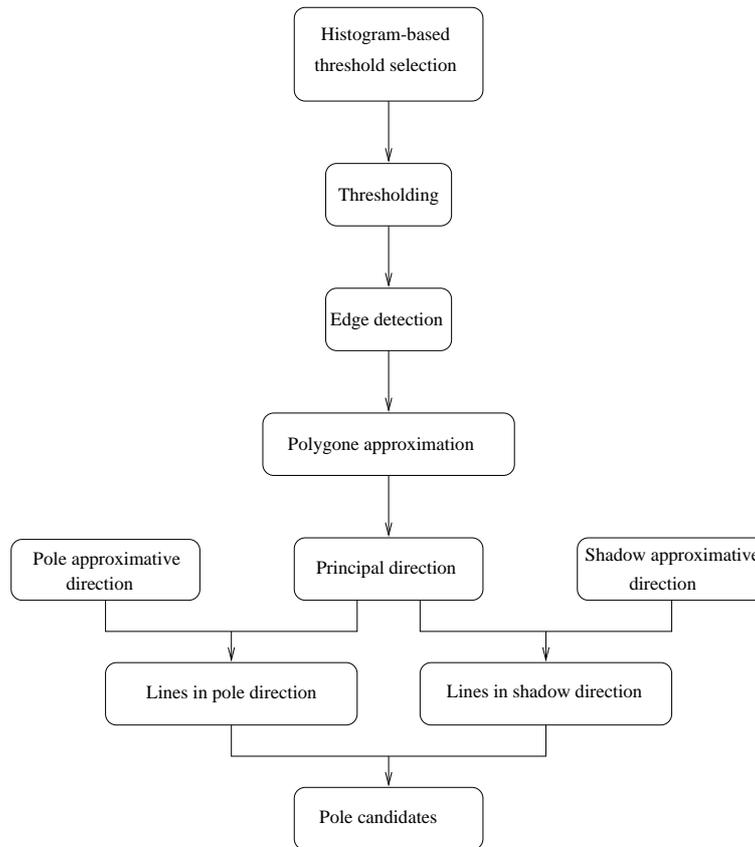


Figure 5.3: Pole detector

Usually, the poles are rather vertical. Therefore, all shadows are in the same direction. That direction may be computed if the orientation of the photo and the location of the sun is known (time of the day). The poles are imaged through a perspective projection (see Figure 5.1).

Therefore, as shown in Figure 5.2, all pole projection lines converge through the same point, in the image or outside the image. If the height of the poles is known, the length of the projection may be computed. This length is maximal in the corners and minimal in the middle (principal point) of the image. The length of the horizontal projection is constant on a vertical line (and vice-versa).

A pole detector has been built, using this very special configuration of directions (Figure 5.3).

First the histogram of grey levels is computed. It is used to select the threshold. The image is then thresholded to segment the dark regions. A number of false alarms may be rejected by keeping only regions with appropriate size. The contours of those regions are then computed (see Figure 5.4). Those contours are approximated by polygons to get a good estimation of the contour direction.

The contours may be classified as pole, shadow or false alarm according to their direction (see Figure 5.5).

Figure 5.6 presents a typical candidate obtained after thresholding and size-based filtering.

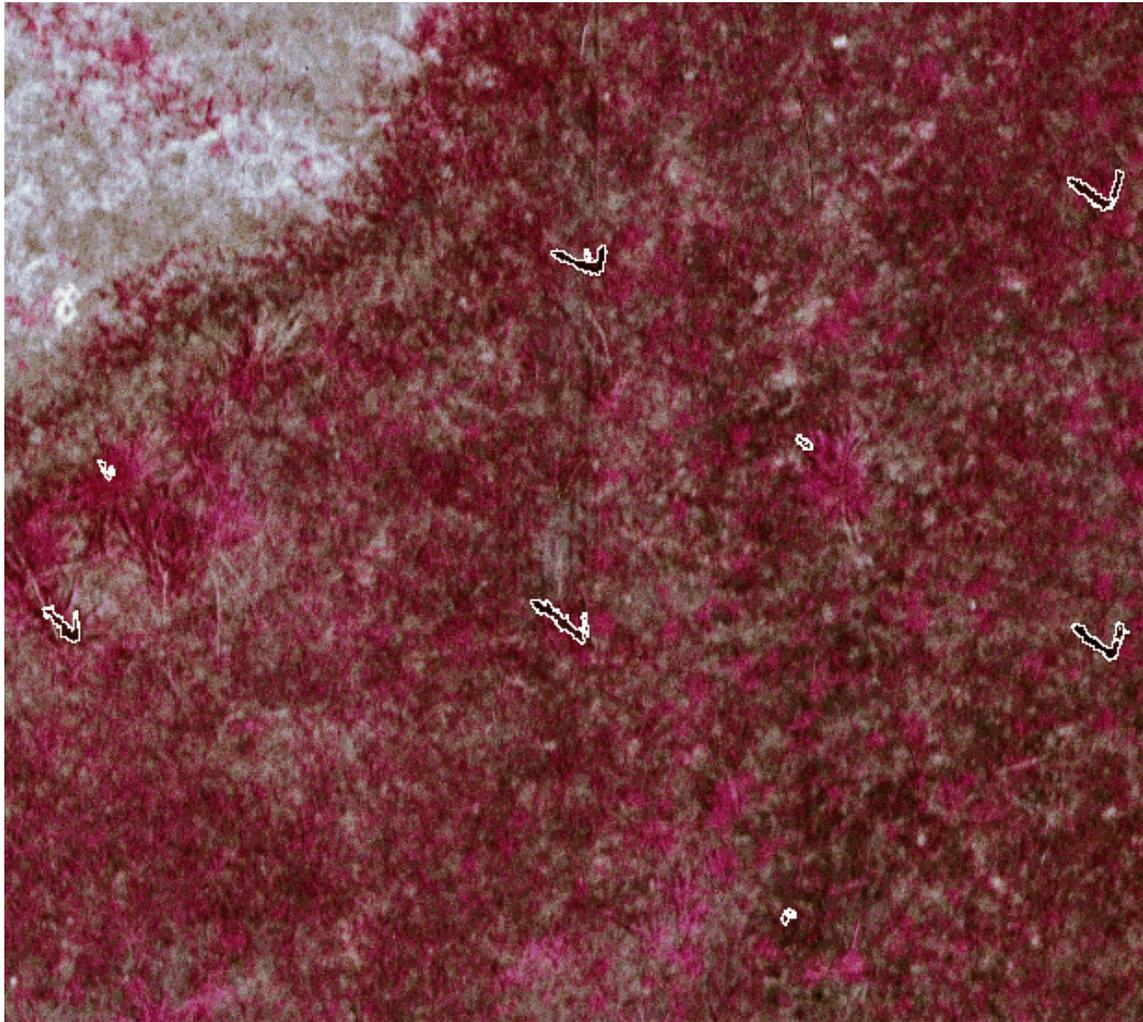


Figure 5.4: Thresholding of a pole

Figure 5.7 presents a typical example of remaining candidate pairs that are composed of two parallel lines with appropriate distance and appropriate direction. The number of false alarms could further be reduced by keeping only candidates for which the two parts of the V are found.

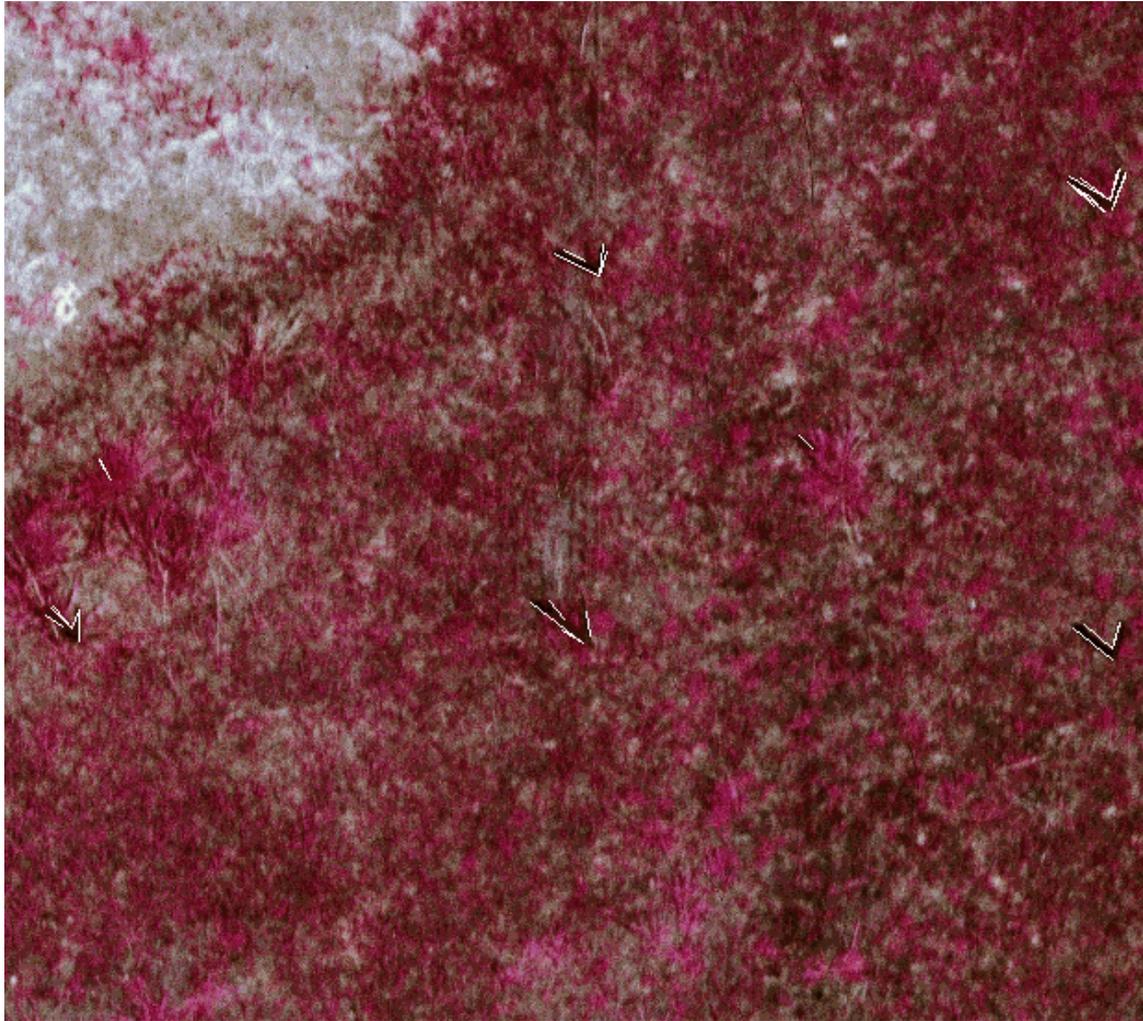


Figure 5.5: Pole detection



Figure 5.6: Close-up of the thresholding



Figure 5.7: Remaining candidate



Figure 5.8: Alignment of poles

5.3 Detection of alignment of poles

The method used has two steps that follow the detections of poles previously described.

First the Hough transform is computed to find the orientation along which there is the largest number of detected poles. The method accepts some errors in the alignment.

Then a regularity is searched in the chosen alignment. For this the ordinates of the poles detected along their common orientation axis are considered. The largest common divider of these numbers is the estimation of the distance between two poles. The algorithm, called the *modified Euclidean algorithm* [5], takes care of imprecision in the detection of the positions of the poles and the possibility to have not detected some of the poles.

Figure 5.8 shows the results that can be obtained automatically. All the white segments are pole candidates. For the example here, no filtering based on the direction has been made. Yet, the algorithm finds the two rows of poles. On the figure the left and right ends of the upper minefield boundary are shorter than the lower boundary because either a pole has been missed in the initial detection or it was not present in the image.

5.4 Conclusion

The pole detector presented here is efficient, fast and can be used on very large images. The detection of alignments reduces drastically the number of false alarms while not being handicapped by the non-detection of some of the poles. Used together these two algorithms are efficient tools to detect areas that are delimited by alignments of poles.

Chapter 6

Detection of elongated areas

6.1 Introduction

The minefield of Buzi (Mozambique) is an elongated area five-meter wide. Areas that are suspected minefields are not used by local population. There are no cultures and vegetation grow naturally, which may not be the case of areas just outside the suspected minefield. Since the vegetation on the minefield is different from the vegetation outside, especially in colour infrared, an elongated area with specific colour and texture is a good indicator.

Figure 6.1 presents a part of the Buzi minefield on which the methods developed for semi automatic detection of elongated areas will be presented.



Figure 6.1: Part of Buzi minefield



Figure 6.2: Contour-based approach result

Two approaches have been developed for the detection of elongated areas. In both cases the image is first segmented to extract uniform regions. The result of the first approach is presented in Figure 6.2. This approach uses the contours of the segmented regions to detect elongated structure. The second approach uses the skeleton of the segmented regions to detect elongated structures. The result of that approach is presented in Figure 6.3.

The segmentation method is explained in section 6.2. The contour and skeleton-based approaches are then described respectively in sections 6.3 and 6.4.

6.2 Segmentation

The segmentation process is the first step of the detection of elongated areas. The objective of such a detector is to pre-filter the available images to present only regions of interest to the photo-interpreter. In such an approach, the detector has to be launched on large images to provide regions of interest without the intervention of the photo-interpreter. The selected algorithm should therefore be fast and work in an unsupervised mode. This leads us to the choice of the unsupervised k-means algorithm. Once the k-means algorithm has been selected, one has still to define the input features and the type of distance used. Typically, the Euclidean or the Mahalanobis distances are used. Experience showed that the Mahalanobis distance leads to better results. This is due to the fact that this latter distance automatically adapts itself to the data distribution in the feature space.

The features used as input for the k-means algorithms should be selected with care as they highly condition the quality of the results as well as the amount of memory and computation power needed. We selected colour components in various colour spaces (RGB or HSV) and their variances.



Figure 6.3: Skeleton-based approach result

It was anticipated that the HSV (Hue, Saturation, Value) colour space would give better results as similar colour should lead to simpler clusters (thin in the H direction) more compatible with the implicitly hypothesized Gaussian distribution. However the H component is an angular coordinate. It therefore exhibits a discontinuity barrier and the clustering algorithm becomes somewhat more complicated because it has to take that barrier into account. The variance feature was used because it is a simple texture indicator.

To detect the searched structures (several kilometers long and about 5 meters wide), the full resolution image is not needed. The features are therefore computed at a coarser resolution by computing the mean and average of the colour component on n by n adjacent windows. This leads to a down-sampling by a factor n (typically 20). By this procedure, down-sampling and feature computation are advantageously combined. This leads to a good estimation of the local variance.

Experience showed that, as expected, HSV-based segmentation are better than RGB-based segmentation. However the use of the variance did not give any improvement. On the contrary it degraded the result by over-segmenting the regions of interest. This may be explained by the fact that the variance computation is a high pass filter which amplifies the noise. Furthermore, no significant texture variation is seen on the test images.

The segmentation obtained using the HSV mean as features is presented in Figure 6.4.

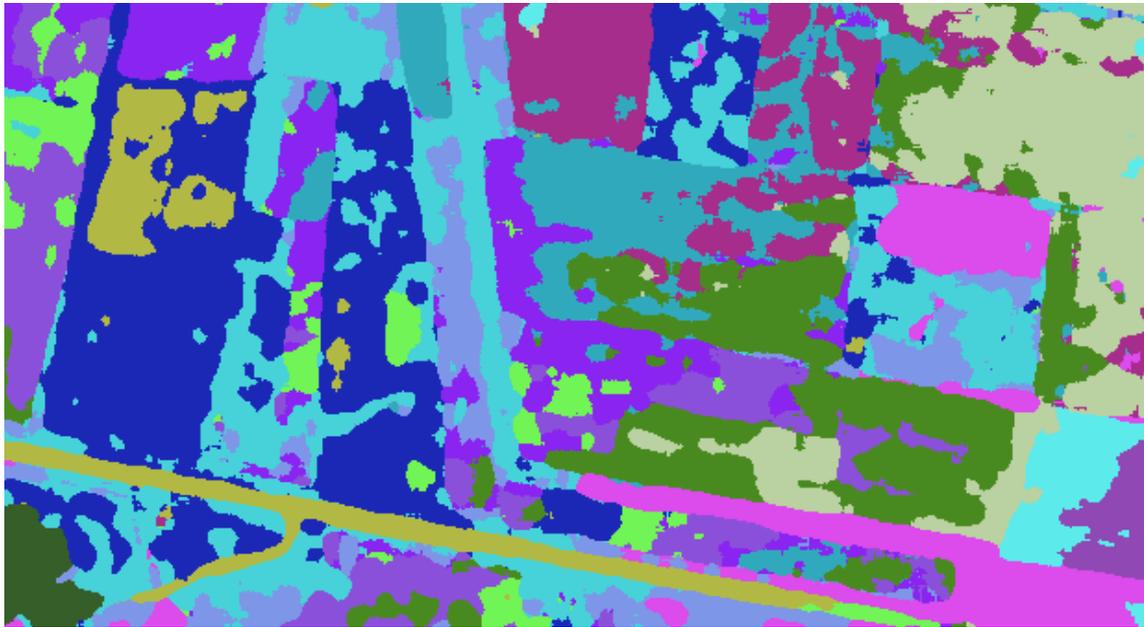


Figure 6.4: Segmentation Result

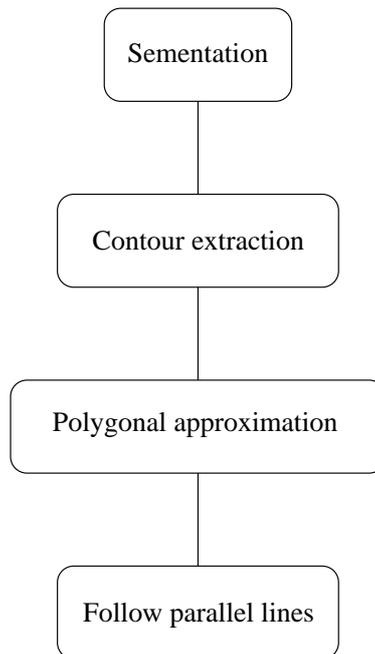


Figure 6.5: Contour-based detector processing scheme

6.3 Contour-based detector for elongated areas

Figure 6.5 presents the implemented processing scheme. The contours of the segmented regions are computed. Amongst those contours, two parallel borders at appropriate distances and with appropriate length are searched and returned as minefield candidates.

To decide whether two borders are parallel, their directions have to be computed. To get a reliable estimation of their directions, the polygonal approximation of the contours is used.

The contours extracted on Figure 6.4 are presented on Figure 6.6 and their polygonal approximation is shown in Figure 6.7.

The contour polygonal approximation is then used to search for parallel borders. Of course the segmentation is imperfect and the research for parallel borders must not fail in presence of gaps in the borders. For this, a gap bridging capability is introduced in the algorithm. This gap bridging is very powerful as the structure may be followed as long as only one border is detected (the two borders must be detected at some part to initiate the following and at the end to confirm the candidate).

To start the grouping procedure, a connection structure is computed encoding for each line the possible prolongation as well as possible parallel segments (opposite borders). An elongated structure is started when two parallel lines are found at a distance (W_0) within given bounds (prior knowledge of the width of the minefield). The minimal length and covering of the two initial segments may also be specified. An elongated structure of width given by the distance between the two initial lines is followed in both directions.

For each border of the lines the following constraints are used:

- perpendicular distance $< T_1$ (see Figure 6.8)

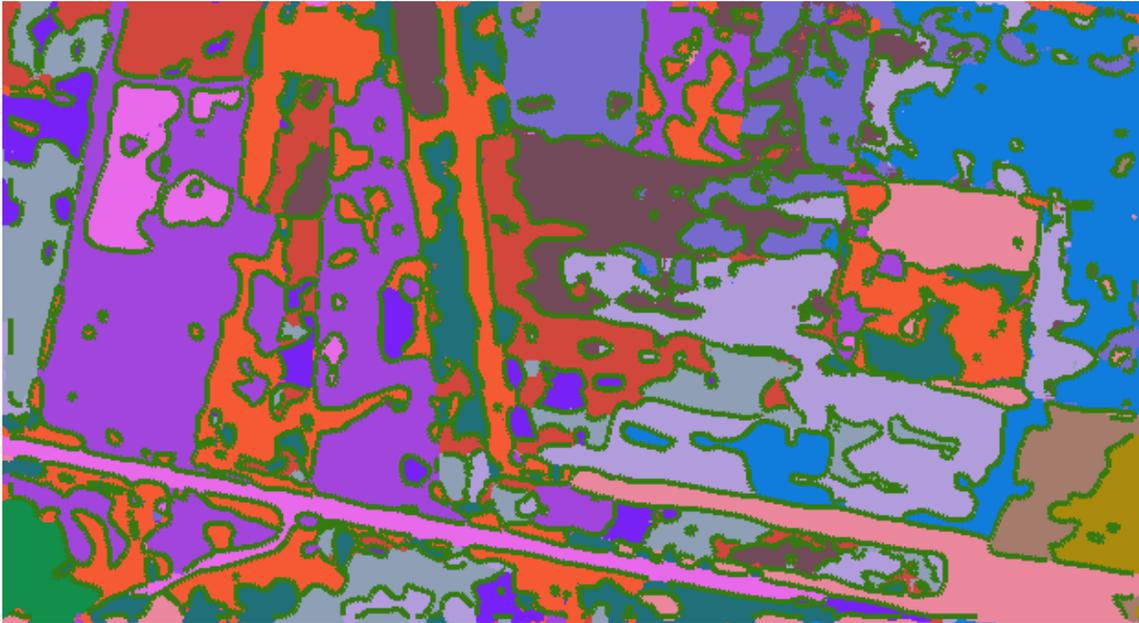


Figure 6.6: Contours

- angle $< T2$ for straight lines (see Figure 6.9)
- cover $< T3$ (see Figure 6.10)
- same gradient direction (same border of region)

Additional constraints are applied to the opposite borders of the elongated area:

- opposite gradient direction (opposite border of region)
- sufficiently parallel: angle $< T4$ (see Figure 6.11)
- cover $> T5$ (see Figure 6.12)
- distance (W) compatible with the seed borders distance (W0) $W - W0 < T6$

Once the borders of the elongated structures have been found, the middle (median line) of that structure may be computed. The median lines found using the contour polygonal approximation of Figure 6.7 are shown in Figure 6.13. By keeping only long median lines, the minefield candidates presented on Figure 6.14 are retained. Those candidates are then presented to photo-interpreter for further inspection.

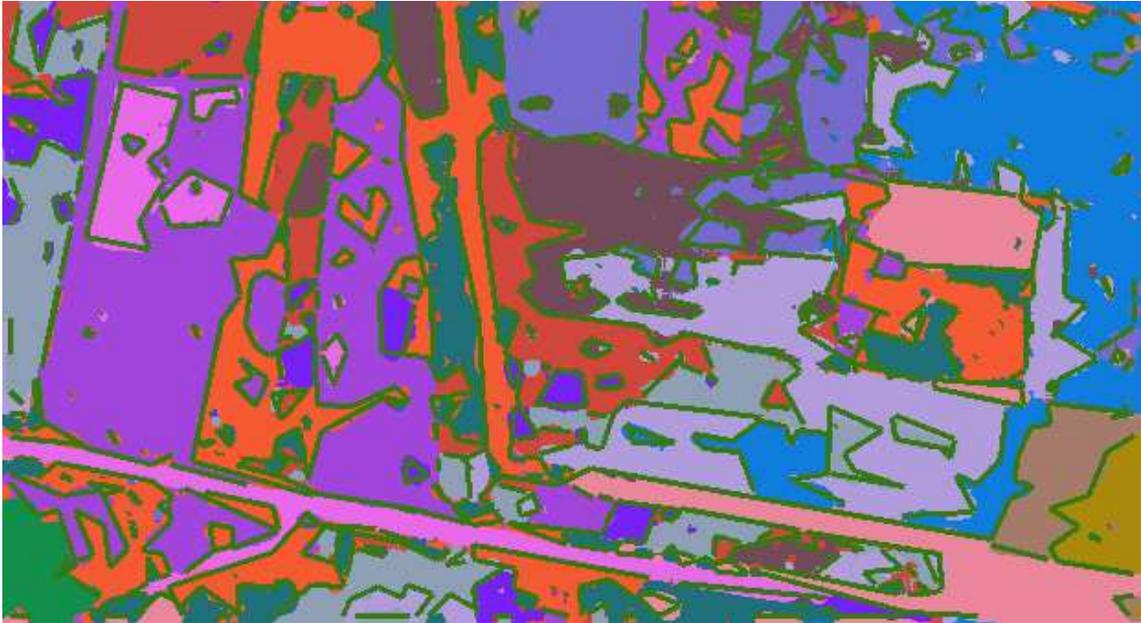


Figure 6.7: Polygonal approximation

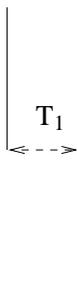


Figure 6.8:
Constraint:
*perpendicular
distance*

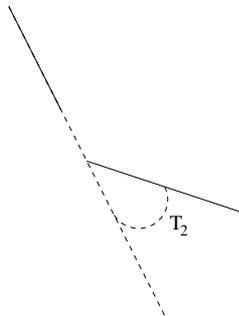


Figure 6.9:
Constraint: *angle*

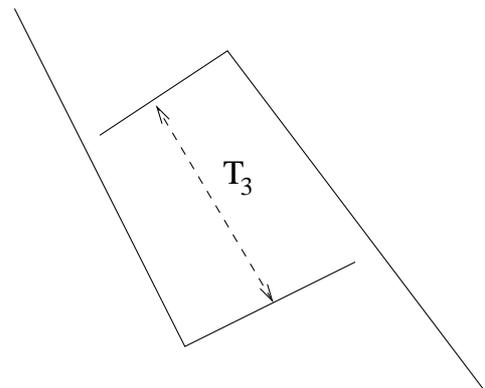


Figure 6.10: Constraint: *cover*

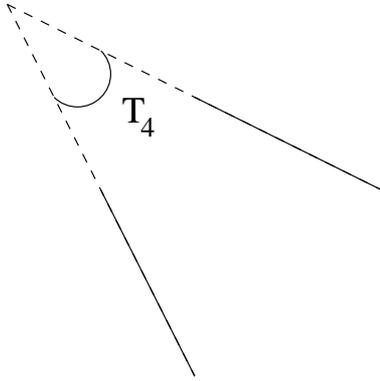


Figure 6.11: Constraint: *sufficiently parallel*

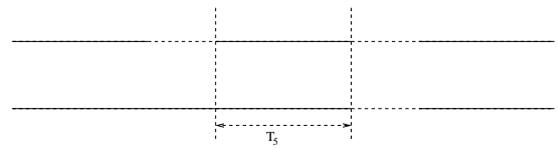


Figure 6.12: Constraint: *opposite cover*

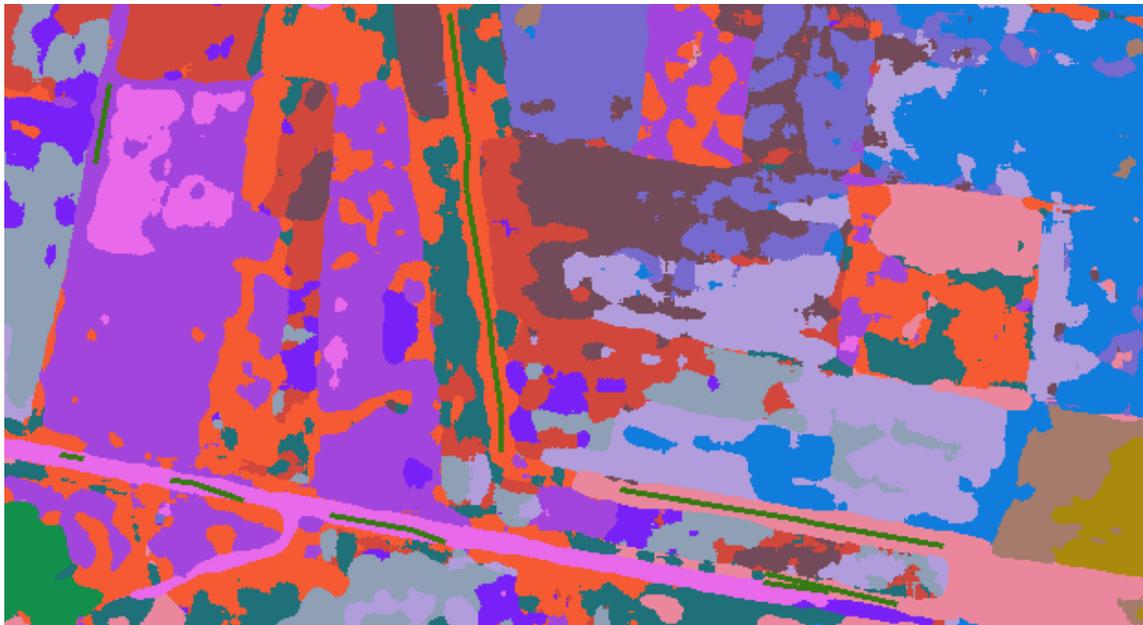


Figure 6.13: Median of parallel borders

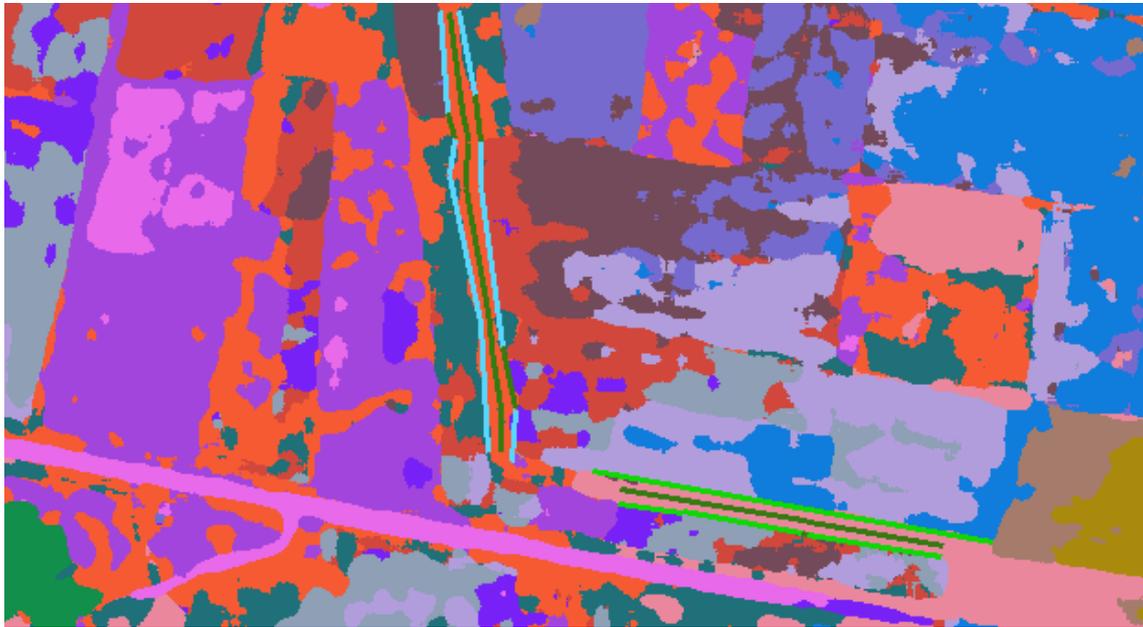


Figure 6.14: Detected minefield candidates

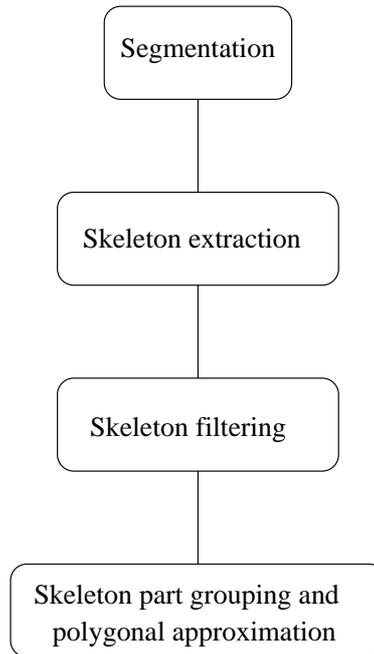


Figure 6.15: Skeleton-based detector processing scheme

6.4 Skeleton-based detector for elongated areas

Figure 6.15 presents the processing scheme used. The skeleton of the segmented regions and the width (distance from the skeleton to the border) of those regions are computed. Those skeletons are then filtered to keep only the parts that belong to regions with appropriate width. The remaining skeleton parts are grouped to provide elongated region median lines. Finally, the long median lines are kept as minefield candidates.

The skeleton of a region is the line at equal distance of the two borders. The skeleton may be computed by searching for ridges of the distance map. Indeed those ridges are at equal distance of at least two contour points. It is time consuming to compute an exact distance map. Therefore the chamfer algorithm which provides a fast and good approximation of the distance map has been used. That algorithm uses the contour image corresponding to the segmented image as input and propagates the distances to those contours in two passes the first works left to right and top down and the second from the right to the left and bottom up. We implemented a generalized version of the classic chamfer algorithm by also propagating the x and y coordinates of the closest point as well as the colour of that point. Using x and y output images, it is possible to recover the closest contour point or the two equidistant contour points for the ridges of the map (the skeleton).

The chamfer distance map corresponding to regions of Figure 6.4 may be seen on Figure 6.16.

The skeleton found on the chamfer distance map presented in Figure 6.16 is shown in Figure 6.17.

The middle of the searched area is part of the obtained skeleton but a lot of parasite skeleton parts are present. To remove those undesired parts, the skeleton is filtered as follows:

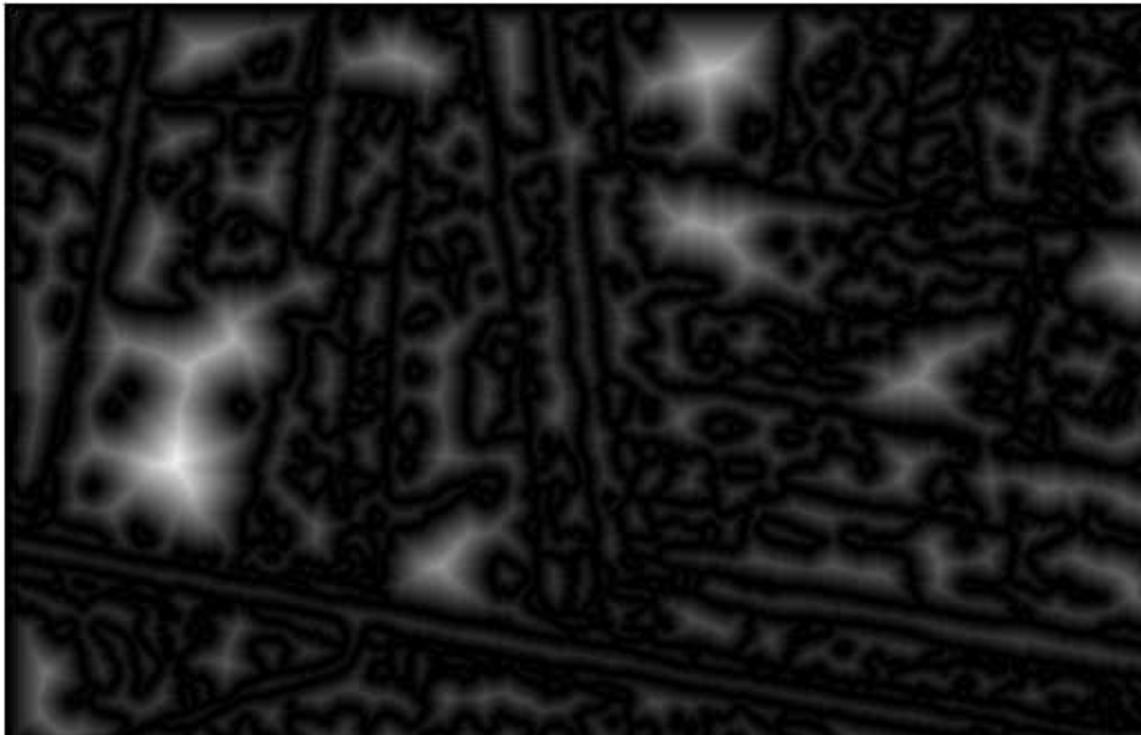


Figure 6.16: Chamfer distance map

- Keep only parts for which the distance to the border is compatible with the width of a minefield. The distance to the border may be read on the chamfer map. The result of this filtering is shown on Figure 6.18.

The remaining skeleton parts are then linked if they fulfill the following criteria:

- Parts are in the same region of the segmented image
- Extremity sufficiently close
- Segment linking two parts stays in the same region that the region in which the two parts are.
- Prolongation sufficiently straight.
- Compatible width (distance from skeleton to border).

The grouped skeleton parts provide the median line of elongated structures. Those median lines are shown on Figure 6.19. A polygonal approximation of the median lines are then compute and sufficiently long lines are kept as minefield candidates. Those candidates are shown in Figure 6.20 and should be considered by the photo-interpreter for further investigation.



Figure 6.17: Skeleton

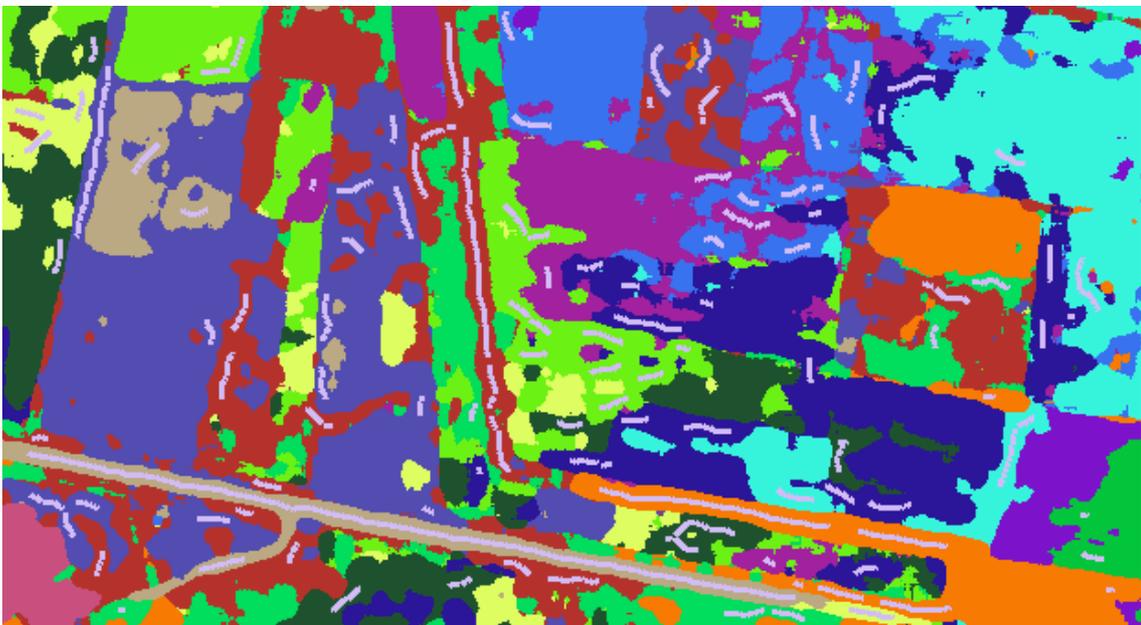


Figure 6.18: Filtered skeleton

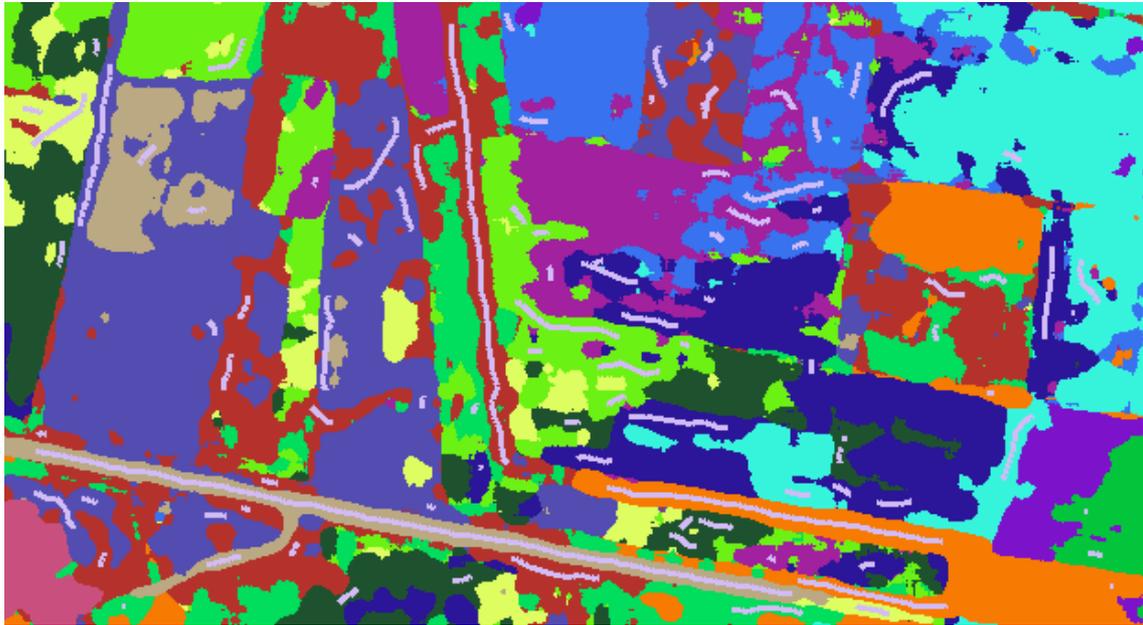


Figure 6.19: Linked skeleton parts

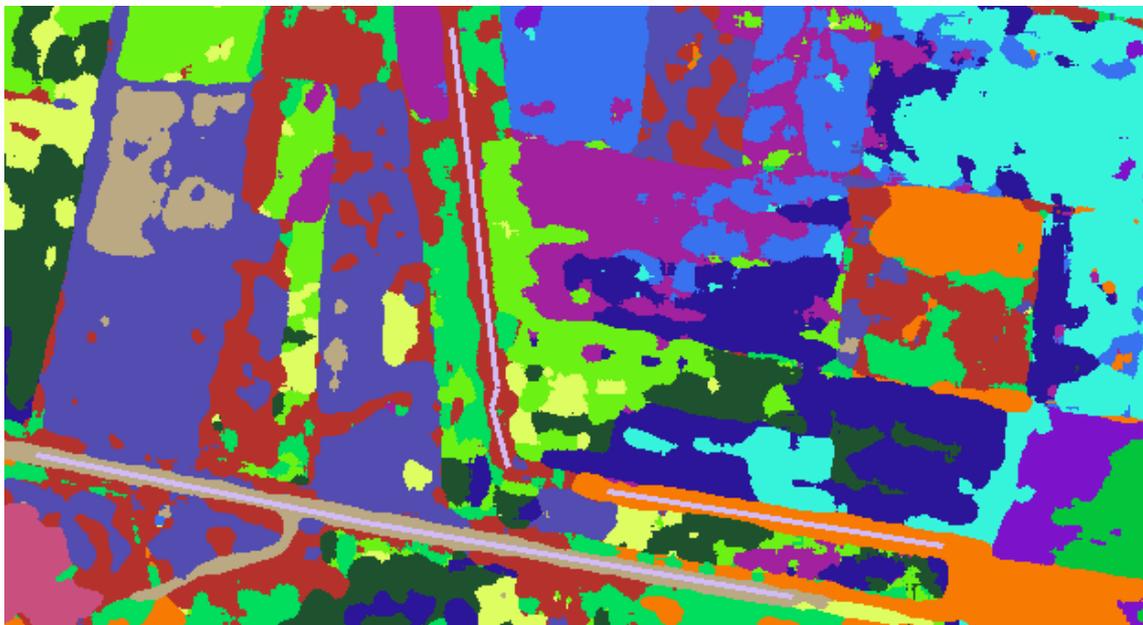


Figure 6.20: Long skeleton parts

6.5 Conclusion

These methods have been developed for the detection of elongated minefields. But the methods present very good results as the minefield was detected with only one false alarm that the interpreter would easily interpret as a road.

Regarding the processing time, both algorithms are foreseen to be used on downsampled images and are fast. The time consuming part is the segmentation which was therefore designed to be fast.

On a 300 megabyte image, the segmentation took around fifteen minutes on a Pentium II at 400 MHz under Linux with 256 megabyte RAM. See Table 6.1 for details.

Algorithm	Time	
Conversion into HSV	3min 20s	
Downsampling and feature computation	5min 46s	
K-means classification	5min 25s	
Contour-based minefield detection	20s	
Skeleton-based minefield detection		9s
Total	16min 11s	14min 20s

Table 6.1: Computation time

Chapter 7

Detection of trenches and foxholes

7.1 Introduction

The minefields of Bandua and Mameme were designed to protect military camps. Remains of these camps are good indicators for minefields. If the place of a former camp is detected, then the presence of a defensive minefield may be inferred. In this chapter, we show how stereovision may be used to detect semi-automatically the difference in depths due to trenches or foxholes. Relevant and easy to detect, they are good candidates for minefield indicators.

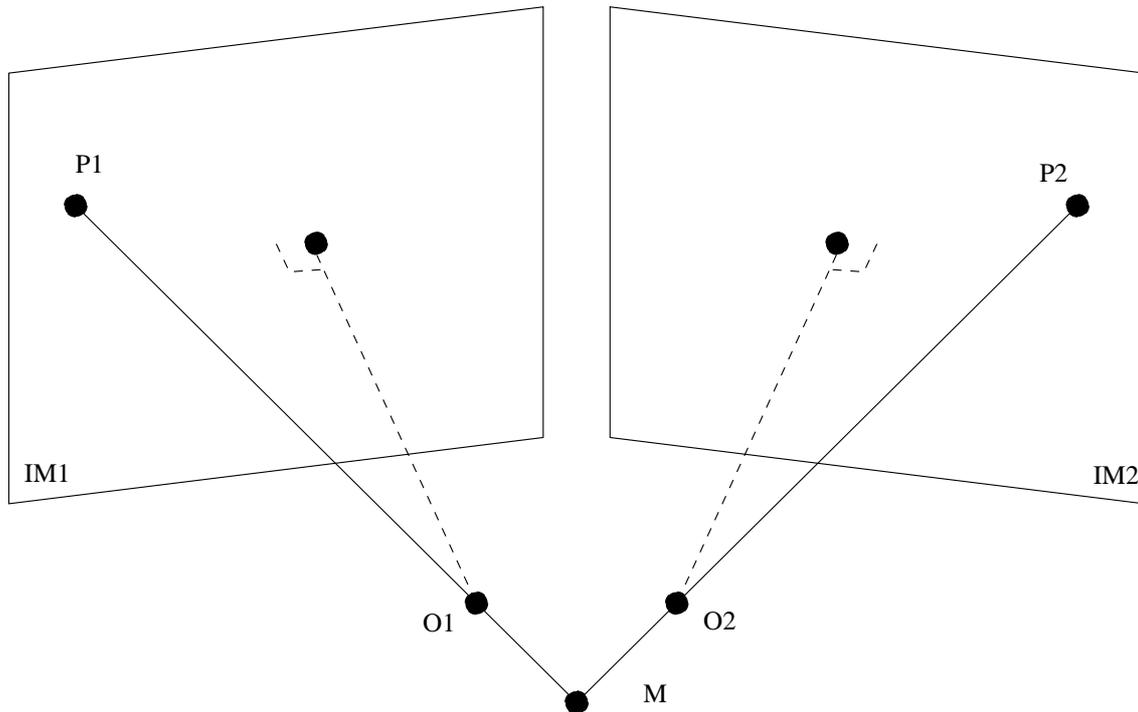


Figure 7.1: Stereovision principles

Just as we are able to have an idea about the depths of objects from the two images that we receive from our two eyes, it is possible to build a three-dimensional representation of a

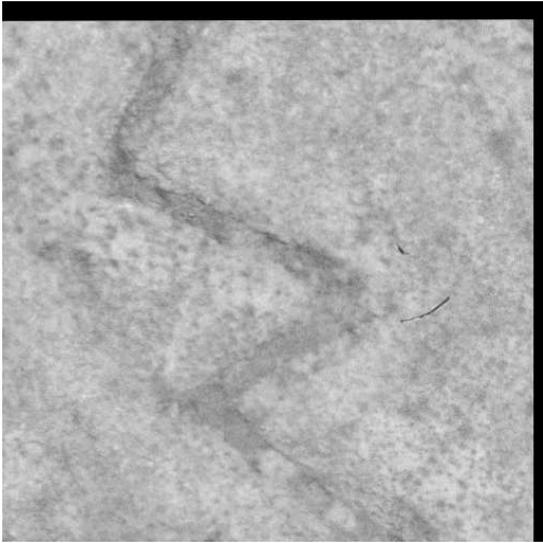


Figure 7.2: Rectified left image

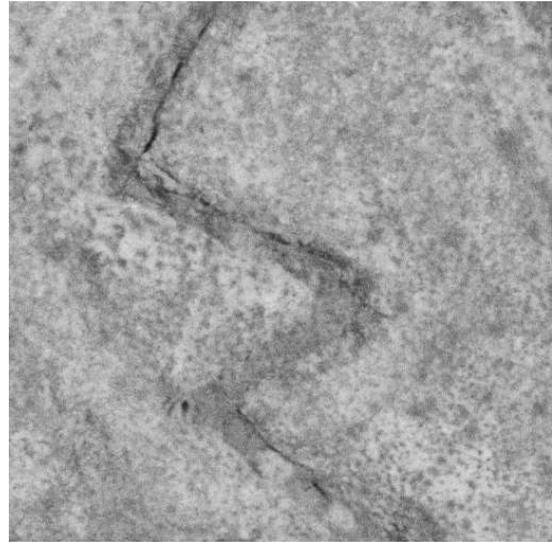


Figure 7.3: Rectified right image

scene from two images taken from two different points of view. The principles are given in Figure 7.1.

A point M in space is seen on the pixel P_1 on image IM_1 and pixel P_2 on image IM_2 . These pixels are the intersection on the lines between M and the optical centres O_1 and O_2 , and the image planes. P_1 and P_2 are called matched pixels. If we can, for each pixel on one image, find its matching pixels on the other images, then it is possible to find the position of M by triangulation.

A three-stage process is applied: rectification, matching and reconstruction.

7.2 Rectification

First a rectification is performed so that any couple of matched pixels are on the same row on the two images. That way, the search of the match is a one-dimensional problem (the search is done along one row of the image) and not a two-dimensional problem (we do not search in the whole image). This reduces the complexity of the task. In a manual rectification, couples of matched pixels, one on each image, are chosen manually and then a re-computation of the two images is performed so that matched pixels are on the same row on each image. In automatic rectification, the selection of these couples is automatic. These points are the basis for the weak calibration of the sensors [8]. That allow then a geometric transformation of the images. This transformation can be seen as a virtual modification of the sensor parameters and the relative positions of the sensors so that matched pixels are on the same row on each image. Figure 7.2 and Figure 7.3 show examples of a rectified stereo pair.

7.3 Matching

Then the matching is done for each pixel on one image. For that, we compute the correlation between a window centred on the pixel and windows centred on each pixel on the corresponding row on the other image. The estimated matched pixel is the one that gives the highest

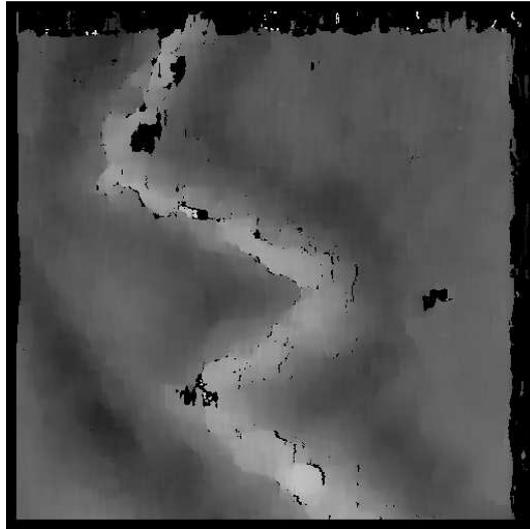


Figure 7.4: Disparity map

correlation value. Then a value called *disparity* is associated to each pixel, representing the number of column that the pixel must be shift to match the column of its correspondent in this other image. This disparity may be real, and not only integer. Such sub-pixel matching is used to reach higher accuracy. This method implies that the images are textured. Otherwise, such as in some areas in Bandua where the ground has a very homogeneous colour, results may be poor. The method used here is based on [3]. The results of this step is a disparity map. See Figure 7.4.

7.4 Reconstruction

If enough information about the way the images were taken is available, a three-dimensional reconstruction is possible giving heights or depths for all pixels. This reconstruction needs the parameters of the position and orientation of the camera (and therefore the aircraft), the height of the aircraft and the calibration parameters of the sensor. Without these data, only a projective reconstruction is possible, which is enough for the present application.

Figures 7.5 and 7.6 show an example of a trench in Bandua as seen on colour infrared image. On the right, the projective reconstruction is shown. We can clearly see the shape of the trench.

The second example (Figure 7.7 and 7.8) is a foxhole in Mameme. For display purposes, only half of the reconstruction is shown. There again, the presence of the foxhole is visible. Unlike in the previous example, here some smoothing has been applied to take care of small ground irregularities.

In these two examples, the rectification was performed manually. This work can be expanded in two directions: a metric reconstruction (to have height, since for the moment, the reconstruction being projective, there is no meaning in the Z-axis) and an automatic rectification.

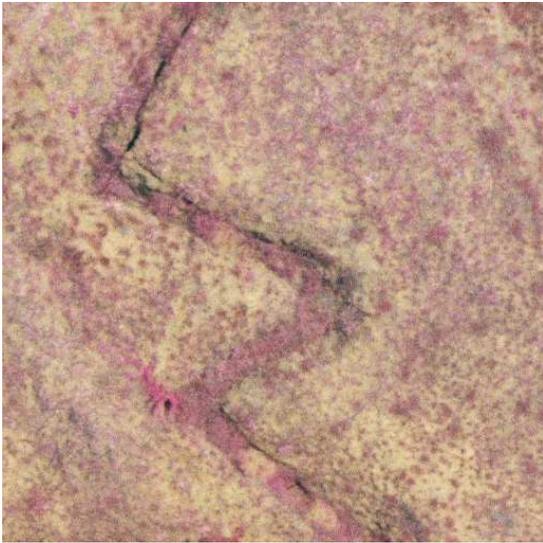


Figure 7.5: Trenches in Bandua

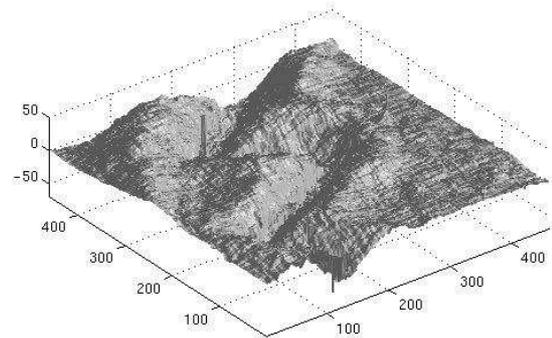


Figure 7.6: Projective reconstruction of the trenches



Figure 7.7: Foxhole in Mameme

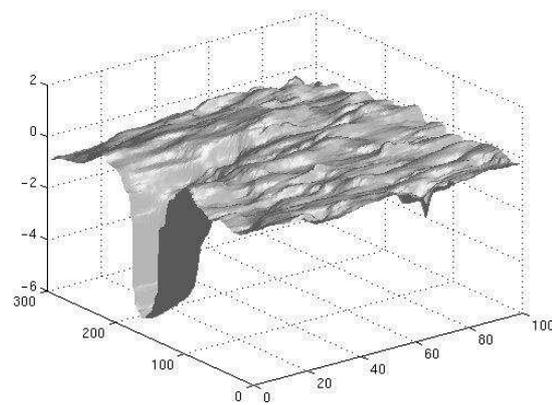


Figure 7.8: Projective reconstruction of the foxhole

7.5 Conclusion

Using stereovision, it was possible to precisely reconstruct the three-dimensional profiles of minefield indicators such as foxholes or trenches.

There are two ways to use these three-dimensional profiles. A detection based on the colour or the texture, or both, may be used to detect trenches and foxholes. These first detectors should not use three-dimensional information in order to be fast enough. They would therefore provide an important number of false alarms. The three-dimensional algorithms could then be used to recover the 3D profile and reject false alarms. Then the photo-interpreter could study more carefully the remaining candidates.

The reconstruction of the three-dimensional profiles may also be provided to the photo-interpreter at demand.

Chapter 8

Analysis of satellite imagery

8.1 Introduction

Satellite imagery was also integrated in the scope of the project to see whether it can show minefield indicators. For the development of semi-automatic tools to speed up the interpretation of these images, we concentrated on the extraction of water areas. Such areas are interesting because they are often of strategic importance. This is why defensive minefields might be expected around dams, for instance. Therefore, it is interesting to make the detection of these areas automatic so that a photo-interpreter can inspect them to search for other minefield indicators.

8.2 Extraction of water areas

Two strategies are envisaged. In the first one, it is assumed that there is a significative number of pixels belonging to the *water class*. This hypothesis is not unrealistic as the coordinates of image corners are well known and as we assume that geographical information such as the rough position of lakes or rivers are known. Thus, in a first step, the method analyses a part of the image (typical sizes range from 512x512 to 1,000x1,000) and performs a learning on this part; it extracts the parameters characterizing the present classes, then looks for these classes in the entire image. The other strategy does not involve a learning phase; it directly works on the whole image.

Different algorithms are used: histogram analysis (see subsection 8.2.1) and classification methods (see subsection 8.2.2).

8.2.1 Histogram analysis

Methods

The Thematic Mapper (TM) images in the fifth band offers the best contrast between water and non-water areas. Thus, in a part of image where there is water, two important modes should be present, and the lowest one should contain the water pixels. In a strategy without learning, it is wise to change first all pixels of 0 value to 255, that is the pixels that appeared because of geometrical correction, as illustrated in Figure 8.1. Indeed, if this precaution is not taken, the lobe containing water may undergo the influence of this important class.

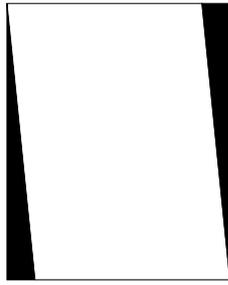


Figure 8.1: Pixels whose values come from geometrical correction

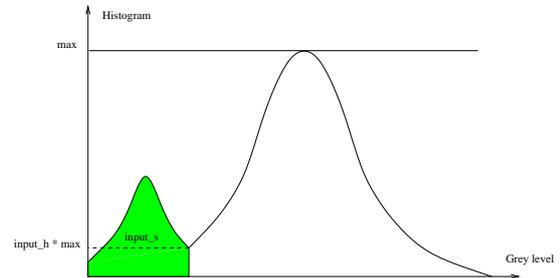


Figure 8.2: Inputs

Several algorithms are proposed for extracting the histogram modes; all have similar input/output and involve three steps.

Here are the input parameters:

- the histogram smoothing window,
- the minimum surface (`input_s`) of the histogram mode (in fraction of the entire image),
- the minimum value of the histogram at the extreme of the lobes (in fraction of the absolute maximum of the histogram) (`input_h`) as shown on Figure 8.2

The output of the program is the list of the extreme values of each of the detected lobes.

First the histogram is smoothed over a window whose size $[2w+1]$ is a program parameter (typically $2w+1=7$). That is, the histogram at g is the mean of the histogram h from $g-w$ to $g+w$. Second, all local extrema of the smoothed histogram are extracted, including the extremes of *landing* intervals. Then there are various methods to detect the lobes.

- (a) The first method tests if there is a lobe between two local minima. Let min_i be a minimum, and $cumu(g)$ be the cumulative histogram at grey level g . If

$$h(min_i) < input_h * max(h)$$

$$h(min_{i+1}) < input_h * max(h)$$

and

$$cumu(min_{i+1}) - cumu(min_i) > input_s * image_size$$

the algorithm considers the range from min_i to min_{i+1} as a lobe.

- (b) The second method searches for the absolute minimum and checks if there is a lobe either from 0 to this absolute minimum or from this minimum to 255. The lobe is present in the interval $[a b]$ if a or b is an absolute minimum and if

$$h(abs_min) < input_h * max(h)$$

and

$$cumu(b) - cumu(a) > input_s * image_size$$

Then each subinterval is recursively analyzed to extract a sub-lobe; the procedure stops when it cannot find any other minimum or the integral of the histogram over the subinterval is too small.

The third method and the fourth method look for the absolute maximum $absmax$ of the histogram and assume that there is one mode around that value.

- (c) In the third method, the first next minimum at the left min_left , i.e.

$$min_left < absmax$$

and at the right, i.e.

$$min_right > absmax$$

that are valid, i.e.

$$h(min_xx) < input_h * max(h)$$

are considered. The integral of the histogram over the two intervals $[0 \ min_left]$ and $[min_right \ 255]$ are computed. The interval over which this integral is the largest is a candidate for a secondary lobe if it is larger than $input_s * image_size$; let us call it the *secondary* interval, and its complement, the *primary* interval; both intervals are then recursively analyzed by the same procedure.

- (d) The fourth method is the same except that instead of considering the next nearest minimum, the absolute minimum at the left and at the right are considered. If none of them gives a partial histogram large enough, the next absolute minima are considered.

These methods have been tested on several TM images.

Summary of the results

- on a part of the image that contains water (default parameters: half smoothing window = 3, $input_s = 0.01$, $input_h = 0.8$)
 - (a) 0-26 27-199
 - (b) 0-27 27-199
 - (c) 0-27 27-255
 - (d) 0-27 27-255
- over a whole image with 0 pixels changed to 255
 - (a) 27-206 251-255
 - (b) 0-206 206-255
 - (c) 0-217 217-255
 - (d) 0-206 206-255
- over a whole image that do not contain pixels generated by the geometrical correction
 - (a) 27-206

(b) 0-206

(c) —

(d) —

In conclusion, only the method that considers all minima is able to separate the water area in the strategy without learning. Moreover, several methods also show another class of bright pixels (i.e from 199 or 206 to 255); we will display these pixels in section 8.3.

In Figure 8.3, we show the borders of the regions thresholded at 27. We may observe that the river and the lake borders are relatively well identified; however, many small rivers are probably not water areas and some parts of the river are also lost near the dam. The next step would be to start from these borders and to stretch them so that they fit local maxima of the gradient.

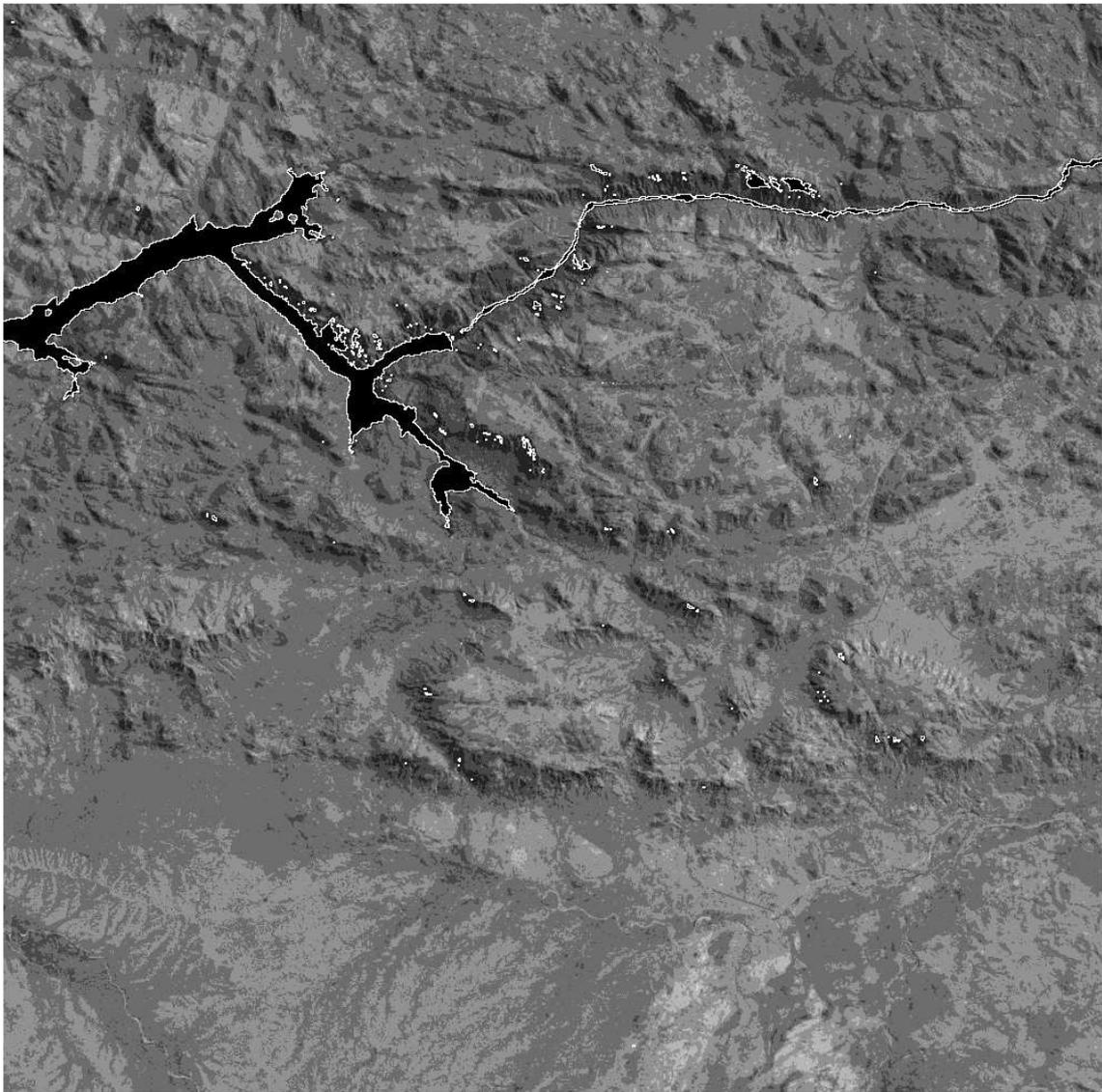


Figure 8.3: Water in satellite image on Songo area

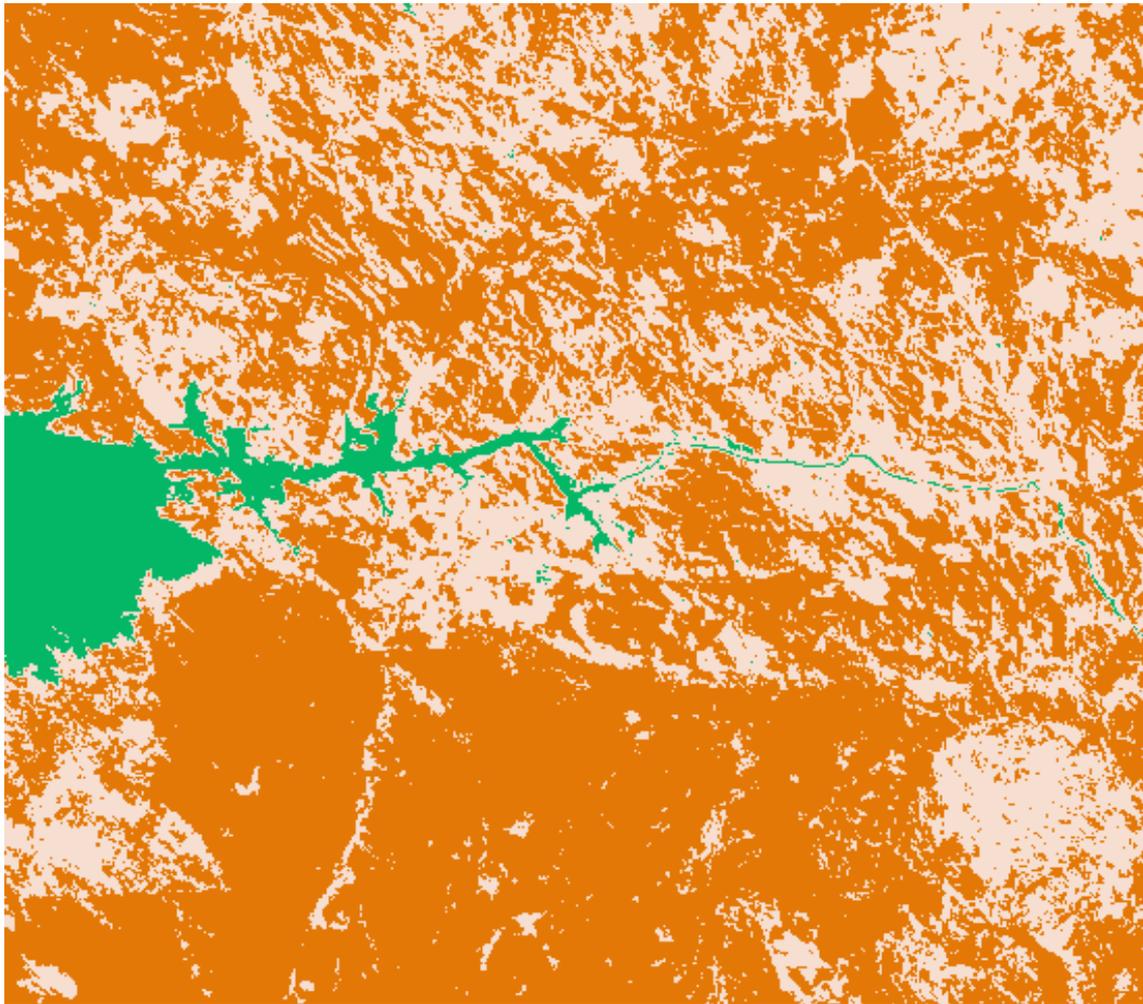


Figure 8.4: Water detection by classification

8.2.2 Classification methods

An iterative algorithm, the k-means, performs the classification. This algorithm assigns feature vectors into clusters on the basis of the nearest neighbour condition. The data are first transformed using Karhunen-Loeve transform. After this transform, the first images contain most of the relevant information. The grey level of the first two images are then used as feature vectors.

The classification result is given in Figure 8.4 with the tm840903 data. The water areas are detected.

8.3 Detection of bright spots

While developing the algorithms for water area detection, bright spots were found. We do not have a clear idea of what they are and whether they could be useful for minefield detection. In order to have a photo-interpreter tell us what they are, they have been detected.

In a first experimentation, the thresholding methods used to extract water areas provided us with a high threshold (i.e from 199 or 206 to 255) which has been used to extract borders of bright spots shown on Figure 8.5.

As a second method, coloured composition makes some small regions appearing distinctively; they are made of bright pixels in the third band and dark pixels in the fifth band. A thresholding method is used on the composition of image by taking the pixels that are the brightest in band three (900/1,000) and the darkest (200/1,000) in band 5. Results are shown in Figure 8.6.

This should however be checked by a photo-interpreter.

8.4 Conclusion

Several methods exist to detect water areas. They all work correctly when applied to find large areas such as lakes or seas. For rivers that are small for the image resolution, some errors may occur. However the methods provide a set of valuable regions of interest that may be studied further using other indicators and knowledge about these parts of the country.

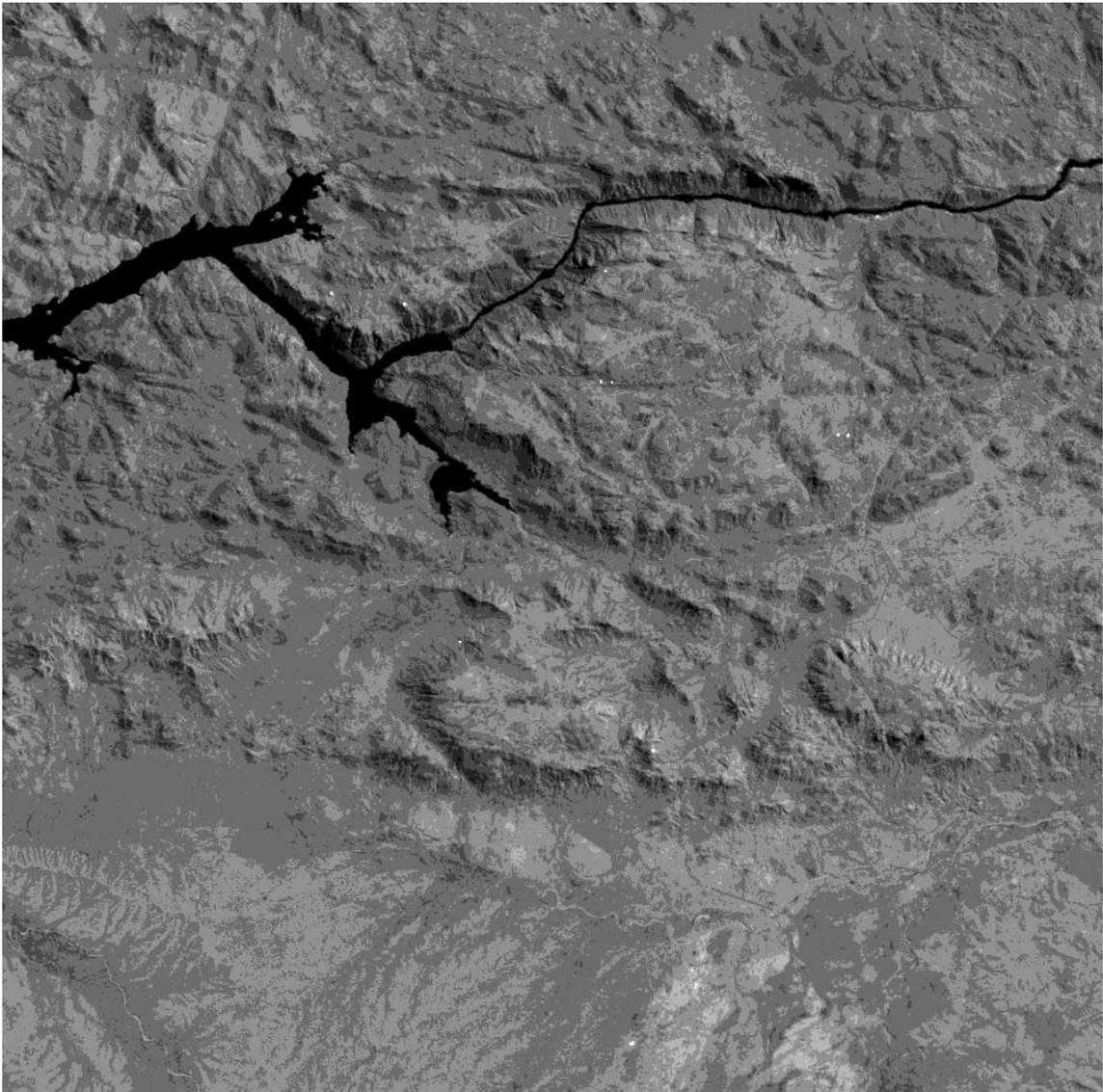


Figure 8.5: Bright spots

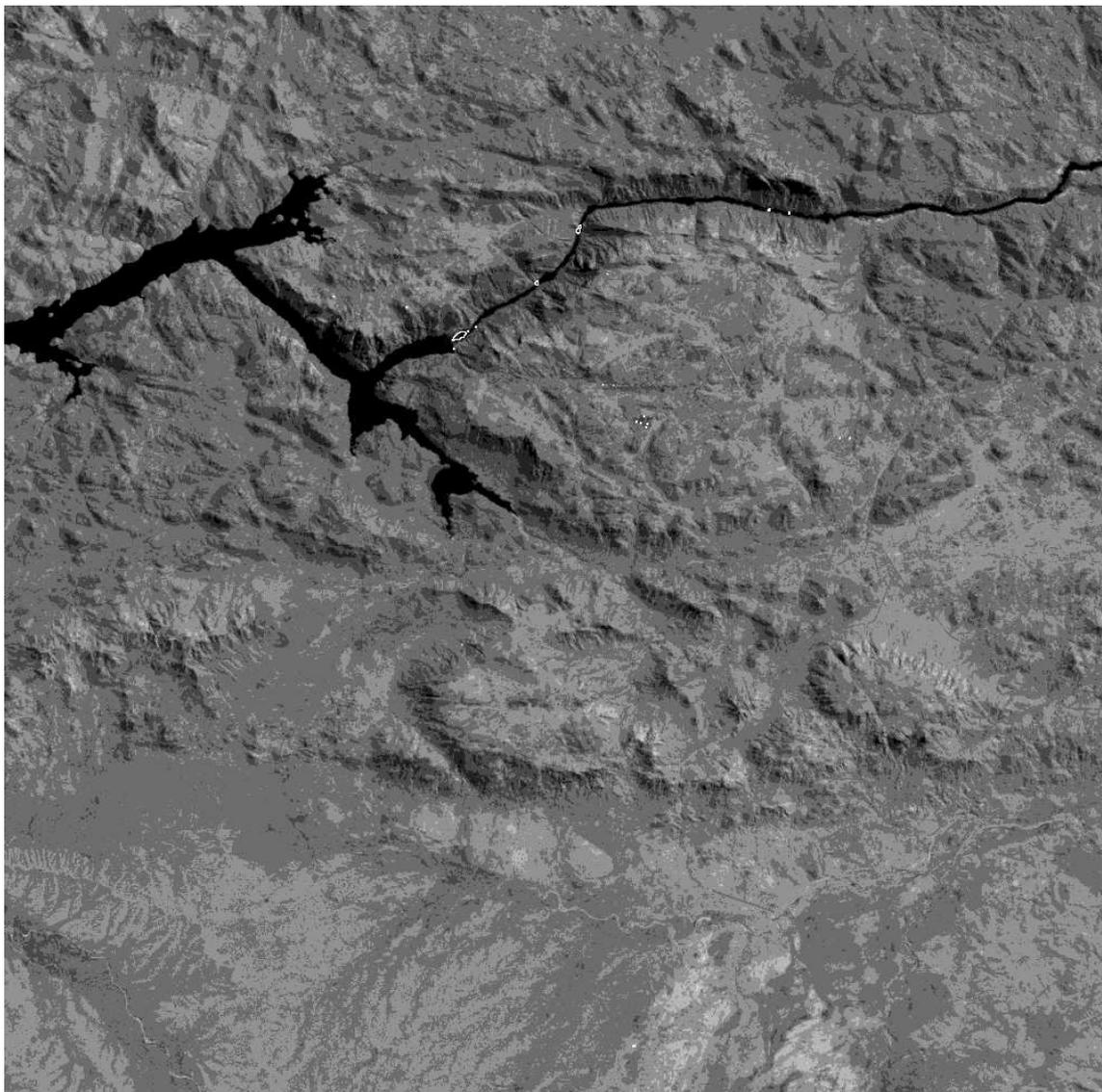


Figure 8.6: Combination

Chapter 9

Compression

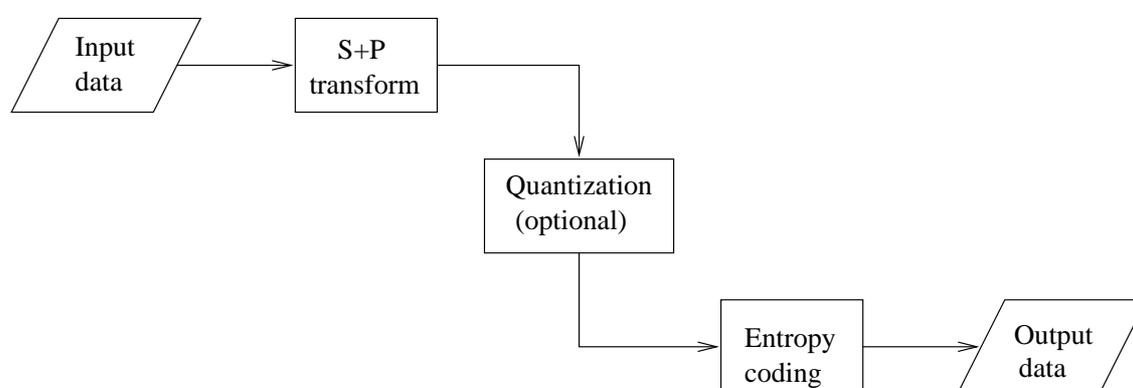


Figure 9.1: WT compression scheme

9.1 Motivation

Compared to human beings, image processing faces a particular difficulty: the large sizes of the images. That is due to the fact that, if image processing has to be used, imagery must be transferred into digital form. Because of the size of the images, disk storage may be a problem. However, it is possible to reduce the sizes of the images without drawbacks. Tests have been performed with an image of 350 megabytes (minefield of Buzi in colour infrared). With a loss-less wavelet-based algorithm, the compression ratio was 1.5. This low compression ratio may be due to the scanning of the images that was performed at a resolution near the grain size of the film. Therefore very little correlation in the grey levels of pixels in the same neighbourhood is present. And most of compression algorithms use this correlation to reduce the size of the data. In order to reach better compression ratio, lossy algorithms are needed. However the acceptable error between the original image and the compressed image depends on the application. If an image is to be used for human analysis, it depends on the size of the details to be searched. We found out that on the images we had, a compression ratio of 10 leads to a loss of quality not perceptible visually. If, on the contrary, the image is to be used for image processing, then a compression error acceptable is an error where the algorithm still

performs correctly. For the pole detection algorithm, we found out that even a compression ratio of 30 did not affect the detector.

The next section presents the principle of the method and the compression ratio that can be reached. The section 9.4 shows results of the pole detector applied to compressed images.

9.2 The method

Our wavelet-based compression algorithm was chosen because its performance in lossless compression is better than JPEG, it is competitive with other wavelet-based methods (same compression ratio and faster), and it allows also lossy compression. Subsection 9.2.1 describes the principle of the method. Subsection 9.2.2 lists the satellite images that were used for testing. Subsection 9.2.3 presents the entropy coding. The results of lossless and lossy compression can be found in subsection 9.2.4. However since this algorithm compress each band separately and the bands are usually correlated, the algorithm can be improved by de-correlation as described in subsection 9.2.5.

9.2.1 Principle

Our WT (Wavelet Transform) compression handles any bit depth, smaller or equal to 16-bit per pixel. The input image is first transformed in a scale-space representation (the S-transform), followed by a prediction stage (P) to further remove eventual remaining redundancies. Several iterations of the S+P transform may be applied recursively in order to get a finer scale-space representation. If a lossy compression is to be performed, the Wavelet Transform coefficients are quantized. The resulting coefficients are then entropy coded using an algorithm similar to the JPEG Variable Length Coding (VLC) followed by Arithmetic Coding (AC). Figure 9.1 shows the principle of the wavelet compression.

One level of the 2D S+P transform is obtained first, by performing the 1D S+P transform (the scale-space transform (S) followed by the prediction (P)) on all the lines of the block. Then by performing the same 1D S+P transform on all the columns of the resulting coefficient block. The transform process is recursive. The transformation is first applied to the whole block and produces four quadrants. It is then re-applied to the upper left quadrant of the resulting coefficients block. These operations will be executed in integer arithmetic, and hence do not require any form of approximation.

The following equations specify the functional definition of the direct scale-space (S) transform for a 1D integer sequence of length N :

$$\begin{cases} T_n & = \lfloor (O_{2n} + O_{2n+1})/2 \rfloor \\ T_{n+N/2} & = O_{2n} - O_{2n+1} \end{cases}$$

With $n = 0 \dots \frac{N}{2} - 1$

The downward truncation of a number x is denoted by $\lfloor x \rfloor$.

The S transform is based on the fact, that the least significant bit of the sum and the least significant bit of the difference are redundant. The transformed sequence is also an integer sequence. The lower part of the transformed sequence (from T_0 to $T_{\frac{N}{2}-1}$) is a low-pass filtered and dyadic down-sampled version of the original sequence; the S transform can be recursively applied to it. The second part (from $T_{\frac{N}{2}}$ to T_{N-1}) contains the high-frequency content of the original sequence and is also down-sampled.

The original sequence can be obtained from the transformed one using the following equations:

$$\begin{cases} O_{2n} &= T_n + \lfloor (T_{n+\frac{N}{2}} + 1)/2 \rfloor \\ O_{2n+1} &= O_{2n} - T_{n+\frac{N}{2}} \end{cases}$$

With $n = 0 \dots \frac{N}{2} - 1$

The prediction stage further removes redundancies in the second part of the transformed sequence using predictive coding. This predictive coding can be parameterized; our coder offers a choice of four different parameters set (no prediction, predictor A, predictor B, predictor C).

In order to make the S+P transform approximately a unitary transform, WT coefficients are multiplied by a factor according to the quadrant they belong to. These factors are positive power of two.

During coding quadrants are scanned from the top left to the bottom right. Within a quadrant, WT coefficients are coded using a left-to-right / right-to left scanning. Each coefficient is associated to a class (number of significant bits) and the class is passed to an arithmetic coder for entropy coding (contextual adaptive coding). The position of the coefficient within its respective class (and the sign of the coefficient) is then binary coded.

Our coder is able either to code the full image as a single block of pixels or either to subdivide it into sub-blocks of small sizes (16x16, 32x32, 64x64). The block-by-block coding offers several advantages:

- It allows to limit the computer resources needed to code the full image
- It allows the use of restart marker between blocks allowing performing error detection and recovery. The frequency of apparition of these markers is parameterized with a user-defined restart interval (RI).
- Finally, it allows to implement easily the coder on multi-processors architecture (tests performed on bi-pentium machines)

9.2.2 Test Images

Scenes	Band dimensions
tm840903	3313x2935x8bits
tm860909	3313x2935x8bits
tm900904	3313x2935x8bits
tm920909	3313x2935x8bits
tm950902	3313x2935x8bits

Table 9.1: LANDSAT TM scenes

Each scene contains 7 spectral bands: Blue (1), Green (2), Red (3), Near InfraRed (4), Near InfraRed (5), Thermal InfraRed (6), Near InfraRed (7).

All bands pixels are stored on 8 bit integer (256 grey levels). Each pixel is related to a square of 30m x 30m on ground, except for the thermal infrared band (6) where the ground resolution is 60m x 60m. That latter band is interpolated (zero order interpolator) in order to have the same dimensions as the other bands.

9.2.3 Zero Order Entropy

The entropy is a typical measure to estimate the information content of a sequence of events (here events are pixel values). The zero order entropy is expressed in bits per pixel (bpp). It is related to the theoretical maximum compression ratio that can be obtained using a zero order entropy coder (using probabilities of occurrences of events without taking into account any contextual information).

0^{th} Entropy	bits per pixel	Ratio
tm840903-b1	4.47	1.79
tm840903-b2	4.06	1.97
tm840903-b3	5.21	1.54
tm840903-b4	5.53	1.45
tm840903-b5	6.54	1.22
tm840903-b6	4.46	1.79
tm840903-b7	5.59	1.43
Mean	5.12	1.60
tm860909-b1	3.60	2.22
tm860909-b2	2.90	2.76
tm860909-b3	3.82	2.09
tm860909-b4	4.69	1.71
tm860909-b5	6.29	1.27
tm860909-b6	3.89	2.06
tm860909-b7	5.32	1.50
Mean	4.36	1.94
tm900904-b1	4.37	1.83
tm900904-b2	3.83	2.09
tm900904-b3	4.62	1.73
tm900904-b4	5.14	1.56
tm900904-b5	6.08	1.32
tm900904-b6	4.49	1.78
tm900904-b7	5.20	1.54
Mean	4.82	1.69
tm920909-b1	4.05	1.98
tm920909-b2	3.65	2.19
tm920909-b3	4.60	1.74
tm920909-b4	4.99	1.60
tm920909-b5	6.06	1.32
tm920909-b6	4.35	1.84
tm920909-b7	5.35	1.50
Mean	5.15	1.74
tm950902-b1	4.11	1.95
tm950902-b2	3.74	2.14
tm950902-b3	4.65	1.72
tm950902-b4	5.02	1.60
tm950902-b5	6.02	1.33
tm950902-b6	3.89	2.06
tm950902-b7	5.26	1.52
Mean	4.67	1.76
Total Mean	4.82	1.75

Table 9.2: Theoretical maximum compression ratio

9.2.4 Lossless and Lossy compression

Introduction

Using Wavelet coder: S+P(B), VLC, AC, 64x64 blocks, RI = 0

- L0: All bit planes are coded (Lossless compression)
- L1: 1 bit plane not coded (Lossy compression)
- L4: 2 bit planes not coded (Lossy compression)
- L9: 3 bit planes not coded (Lossy compression)
- L15: 4 bit planes not coded (Lossy compression)

Intermediate lossy steps are possible with the coder.

Compression times

The typical compression time for the test images ranges from 9 to 12 seconds (times excluding IO) for lossless compression. For lossy compression the computation time ranges from 4 seconds (high loss) to 12 seconds (quasi-lossless) on Pentium II 333 NT.

Compression ratios

See Table 9.3.

Maximum absolute error

The maximum absolute error (MAE) is the maximum absolute difference that is found between pixel values of the original image and their lossy compressed-decompressed counterparts. See Table 9.4.

Root mean square error

The root mean square error (RMSE) is a typical measure of the visual loss that occurs after compression. It is defined as:

$$\text{RMSE} = \sqrt{\frac{\sum_{i,j} (I_{i,j} - \tilde{I}_{i,j})^2}{N}}$$

Where $I_{i,j}$ and $\tilde{I}_{i,j}$ denote respectively the original and compressed-decompressed images and N denotes the number of pixels in the image. See table 9.5.

Ratio	L0	L1	L4	L9	L15
tm840903-b1	3.00	3.47	6.50	17.87	54.37
tm840903-b2	3.48	4.13	8.63	22.61	62.71
tm840903-b3	2.63	2.98	4.90	10.42	24.37
tm840903-b4	2.47	2.77	4.33	8.45	18.75
tm840903-b5	1.94	2.10	2.88	4.73	9.25
tm840903-b6	5.39	6.89	18.77	42.67	80.07
tm840903-b7	2.37	2.64	4.05	7.92	17.57
Mean	3.04	3.57	7.15	16.38	38.16
tm860909-b1	3.28	3.83	8.02	30.19	156.16
tm860909-b2	4.33	5.41	15.23	57.24	179.08
tm860909-b3	3.51	4.16	8.99	25.96	78.32
tm860909-b4	3.08	3.59	6.63	15.45	37.91
tm860909-b5	2.02	2.20	3.09	5.26	10.71
tm860909-b6	5.73	7.38	21.16	52.01	99.17
tm860909-b7	2.44	2.73	4.27	8.59	19.14
Mean	3.48	4.19	9.63	27.81	82.93
tm900904-b1	2.84	3.14	5.05	15.67	72.78
tm900904-b2	3.43	3.93	7.73	27.15	85.62
tm900904-b3	2.69	2.96	4.58	11.00	33.15
tm900904-b4	2.45	2.67	3.86	7.86	21.32
tm900904-b5	1.88	1.99	2.50	3.73	7.36
tm900904-b6	5.69	7.12	17.66	36.47	66.49
tm900904-b7	2.23	2.40	3.28	5.96	15.38
Mean	3.03	3.46	6.38	15.41	43.16
tm920909-b1	2.82	3.12	4.99	15.45	77.34
tm920909-b2	3.51	4.03	8.17	29.47	91.96
tm920909-b3	2.75	3.03	4.77	11.92	36.68
tm920909-b4	2.61	2.86	4.32	9.57	26.93
tm920909-b5	1.89	2.02	2.53	3.80	7.59
tm920909-b6	5.64	7.08	17.63	37.01	71.54
tm920909-b7	2.17	2.32	3.12	5.45	13.55
Mean	3.06	3.49	6.50	16.09	46.51
tm950902-b1	2.78	3.06	4.83	14.23	70.58
tm950902-b2	3.43	3.93	7.73	27.13	85.17
tm950902-b3	2.73	3.00	4.70	11.50	34.46
tm950902-b4	2.53	2.76	4.08	8.59	23.46
tm950902-b5	1.91	2.03	2.57	3.91	7.97
tm950902-b6	6.01	7.55	19.37	41.86	76.69
tm950902-b7	2.22	2.39	3.25	5.90	15.35
Mean	3.09	3.53	6.65	16.16	44.81
Total Mean	3.14	3.65	7.26	18.37	51.11

Table 9.3: Compression ratios

MAE	L0	L1	L4	L9	L15
tm840903-b1	0	1	5	8	13
tm840903-b2	0	1	4	8	13
tm840903-b3	0	1	4	9	17
tm840903-b4	0	1	5	9	15
tm840903-b5	0	1	5	9	18
tm840903-b6	0	1	4	5	8
tm840903-b7	0	1	5	9	16
Mean	0	1	4.57	8.14	14.29
tm860909-b1	0	1	4	7	9
tm860909-b2	0	1	4	7	8
tm860909-b3	0	1	4	8	13
tm860909-b4	0	1	4	8	15
tm860909-b5	0	1	5	10	18
tm860909-b6	0	1	3	5	8
tm860909-b7	0	1	5	9	17
Mean	0	1	4.14	7.71	12.57
tm900904-b1	0	1	5	8	15
tm900904-b2	0	1	4	9	14
tm900904-b3	0	1	4	8	15
tm900904-b4	0	1	4	10	16
tm900904-b5	0	1	4	10	20
tm900904-b6	0	1	4	6	12
tm900904-b7	0	1	5	9	16
Mean	0	1	4.29	8.57	15.43
tm920909-b1	0	1	5	9	15
tm920909-b2	0	1	4	8	15
tm920909-b3	0	1	5	9	15
tm920909-b4	0	1	5	9	17
tm920909-b5	0	1	5	9	21
tm920909-b6	0	1	4	8	13
tm920909-b7	0	1	5	9	17
Mean	0	1	4.71	8.71	16.14
tm950902-b1	0	1	5	10	14
tm950902-b2	0	1	5	7	13
tm950902-b3	0	1	5	9	15
tm950902-b4	0	1	5	9	16
tm950902-b5	0	1	5	9	18
tm950902-b6	0	1	3	6	12
tm950902-b7	0	1	5	10	18
Mean	0	1	4.71	8.57	15.14
Total Mean	0	1	4.48	8.34	14.71

Table 9.4: Maximum absolute error

RMSE	L0	L1	L4	L9	L15
tm840903-b1	0	0.41	0.93	1.48	1.97
tm840903-b2	0	0.39	0.87	1.31	1.75
tm840903-b3	0	0.41	0.96	1.63	2.38
tm840903-b4	0	0.41	0.97	1.68	2.56
tm840903-b5	0	0.42	1.02	1.88	3.22
tm840903-b6	0	0.31	0.62	0.88	1.19
tm840903-b7	0	0.41	0.98	1.74	2.65
Mean	0	0.39	0.91	1.51	2.25
tm860909-b1	0	0.40	0.92	1.42	1.83
tm860909-b2	0	0.38	0.79	1.13	1.45
tm860909-b3	0	0.40	0.87	1.30	1.69
tm860909-b4	0	0.40	0.90	1.44	2.01
tm860909-b5	0	0.42	1.02	1.86	3.10
tm860909-b6	0	0.30	0.59	0.84	1.11
tm860909-b7	0	0.41	0.98	1.70	2.55
Mean	0	0.39	0.87	1.38	1.96
tm900904-b1	0	0.38	0.91	1.60	2.01
tm900904-b2	0	0.38	0.87	1.32	1.68
tm900904-b3	0	0.38	0.92	1.63	2.29
tm900904-b4	0	0.38	0.93	1.71	2.63
tm900904-b5	0	0.38	0.93	1.80	3.42
tm900904-b6	0	0.29	0.62	0.93	1.25
tm900904-b7	0	0.38	0.93	1.79	2.93
Mean	0	0.37	0.87	1.54	2.32
tm920909-b1	0	0.38	0.91	1.61	2.02
tm920909-b2	0	0.38	0.87	1.30	1.64
tm920909-b3	0	0.38	0.84	1.61	2.22
tm920909-b4	0	0.38	0.92	1.66	2.43
tm920909-b5	0	0.38	0.93	1.80	3.41
tm920909-b6	0	0.29	0.62	0.93	1.27
tm920909-b7	0	0.38	0.93	1.80	3.03
Mean	0	0.37	0.86	1.53	2.29
tm950902-b1	0	0.38	0.91	1.63	2.07
tm950902-b2	0	0.38	0.87	1.32	1.68
tm950902-b3	0	0.38	0.92	1.61	2.26
tm950902-b4	0	0.38	0.93	1.69	2.54
tm950902-b5	0	0.38	0.93	1.80	3.38
tm950902-b6	0	0.27	0.60	0.89	1.20
tm950902-b7	0	0.38	0.93	1.79	2.94
Mean	0	0.36	0.87	1.53	2.30
Total Mean	0	0.38	0.88	1.50	2.22

Table 9.5: Root mean square error

9.2.5 Improving lossless compression using spectral de-correlation

The wavelet coder reduces the entropy of the original image by means of the wavelet transform that is a good tool to spatially de-correlate pixel values.

The aim of the spectral de-correlation is to try to take into account the information content of a previously coded image (reference) to code the current image.

Linear prediction

This was done using linear prediction:

- Predict the pixels of the image to compress using pixels of the reference image using a linear function

$$I_{\text{LPC}} = a_{\text{LPC}}R + b_{\text{LPC}}$$

- Code the difference between current pixel (I) and prediction (In fact, code the residual image)

The linear predictor coefficients (LPC A and LPC B) are estimated using residual minimum variance criteria (least square minimization).

$$\begin{aligned} \text{CRIT} &= \sum_{i,j} (I_{i,j} - I_{i,j}^{\text{LP}})^2 \\ \text{CRIT} &= \sum_{i,j} (I_{i,j} - (a_{\text{LPC}}R_{i,j} + b_{\text{LPC}}))^2 \\ \text{CRIT} &= \sum_{i,j} I_{i,j}^2 \\ &\quad - 2a_{\text{LPC}} \sum_{i,j} I_{i,j}R_{i,j} \\ &\quad - 2b_{\text{LPC}} \sum_{i,j} I_{i,j} \\ &\quad + a_{\text{LPC}}^2 \sum_{i,j} R_{i,j}^2 \\ &\quad + 2a_{\text{LPC}}b_{\text{LPC}} \sum_{i,j} R_{i,j} \\ &\quad + b_{\text{LPC}}^2 \end{aligned}$$

Using partial derivative and solving for a_{LPC} and b_{LPC} , one can easily find the linear predictor coefficients that minimize the criteria.

Correlation factors

A good estimation the correlation between bands can be obtained by computing the covariance and correlation matrixes. The elements of the covariance matrix are computed as follow:

$$\text{CVAR}_{i,j} = \frac{\sum_{k=1}^M (x_i^k x_j^k)}{N}$$

Where x_i^k and x_j^k are the k^{th} pixel values (mean normalized) of respectively image i and j , and N is the number of image pixels.

The correlation matrix is just computed using a proper normalization of the covariance matrix; this normalization scales the axis to deal with unit variances:

$$\text{COR}_{i,j} = \frac{\text{CVAR}_{i,j}}{\sqrt{\text{CVAR}_{i,i}\text{CVAR}_{j,j}}}$$

During computation of the linear predictor coefficients and correlation factors, zero pixels are excluded (images mask due to geometric rectification). This allows avoiding a statistical bias due to the large number of zero pixels.

Linear prediction & Scene tm840903

See Table 9.6 to Table 9.12.

Image	Mean				Std	O^{th} Entropy	Ratio
tm840903-b1	96.17				5.43	4.47	3.00
					Residual image		
Image	Reference	Cor.	LPC A	LPC B.	Std	O^{th} Entropy	Ratio
tm840903-b1	tm840903-b2	0.92	1.22	45.48	2.14	3.15	2.99
tm840903-b1	tm840903-b3	0.59	0.35	76.62	4.38	3.88	3.17
tm840903-b1	tm840903-b4	0.24	0.09	90.91	5.27	4.36	3.03
tm840903-b1	tm840903-b5	0.26	0.05	91.13	5.25	4.34	3.03
tm840903-b1	tm840903-b6	0.00	0.00	97.25	5.43	4.47	3.00
tm840903-b1	tm840903-b7	0.25	0.09	91.33	5.25	4.35	3.02

Table 9.6: tm840903 (1)

Image	Mean				Std	O^{th} Entropy	Ratio
tm840903-b2	41.57				4.09	4.06	3.48
					Residual image		
Image	Reference	Cor.	LPC A	LPC B.	Std	O^{th} Entropy	Ratio
tm840903-b2	tm840903-b1	0.92	0.69	-25.09	1.61	2.76	3.36
tm840903-b2	tm840903-b3	0.74	0.33	23.17	2.76	3.02	3.86
tm840903-b2	tm840903-b4	0.42	0.12	34.66	3.72	3.81	3.56
tm840903-b2	tm840903-b5	0.39	0.05	35.78	3.77	3.84	3.53
tm840903-b2	tm840903-b6	0.04	0.02	37.77	4.09	4.03	3.46
tm840903-b2	tm840903-b7	0.36	0.10	36.42	3.82	3.89	3.49

Table 9.7: tm840903 (2)

Image	Mean				Std	O^{th} Entropy	Ratio
tm840903-b3	55.93				9.18	5.21	2.63
					Residual image		
Image	Reference	Cor.	LPC A	LPC B.	Std	O^{th} Entropy	Ratio
tm840903-b3	tm840903-b1	0.59	1.00	-40.15	7.41	4.60	2.69
tm840903-b3	tm840903-b2	0.74	1.65	-12.83	6.20	3.98	2.83
tm840903-b3	tm840903-b4	0.87	0.56	23.63	4.52	4.13	3.05
tm840903-b3	tm840903-b5	0.85	0.25	27.62	4.76	4.24	2.92
tm840903-b3	tm840903-b6	0.46	0.61	-39.54	8.15	5.04	2.60
tm840903-b3	tm840903-b7	0.78	0.47	30.75	5.73	4.55	2.78

Table 9.8: tm840903 (3)

Image	Mean				Std	O^{th} Entropy	Ratio
tm840903-b4	57.79				14.30	5.53	2.47
					Residual image		
Image	Reference	Cor.	LPC A	LPC B.	Std	O^{th} Entropy	Ratio
tm840903-b4	tm840903-b1	0.24	0.63	-2.87	13.88	5.39	2.48
tm840903-b4	tm840903-b2	0.42	1.46	-2.87	12.99	5.09	2.54
tm840903-b4	tm840903-b3	0.87	1.36	-18.05	7.04	4.56	2.65
tm840903-b4	tm840903-b5	0.93	0.43	10.03	5.41	4.41	2.65
tm840903-b4	tm840903-b6	0.51	1.06	-107.42	12.28	5.51	2.43
tm840903-b4	tm840903-b7	0.80	0.74	17.73	8.62	5.10	2.43

Table 9.9: tm840903 (4)

Image	Mean				Std	O^{th} Entropy	Ratio
tm840903-b5	111.84				31.00	6.54	1.94
					Residual image		
Image	Reference	Cor.	LPC A	LPC B.	Std	O^{th} Entropy	Ratio
tm840903-b5	tm840903-b1	0.26	1.47	-29.29	29.96	6.29	1.94
tm840903-b5	tm840903-b2	0.39	2.97	-11.59	28.51	6.01	1.96
tm840903-b5	tm840903-b3	0.85	2.89	-49.63	16.08	5.73	1.93
tm840903-b5	tm840903-b4	0.93	2.01	-4.14	11.72	5.52	2.00
tm840903-b5	tm840903-b6	0.68	3.04	-360.89	22.84	6.40	1.89
tm840903-b5	tm840903-b7	0.94	1.90	9.20	10.33	5.30	2.09

Table 9.10: tm840903 (5)

Image	Mean				Std	O^{th} Entropy	Ratio
tm840903-b6	155.40				6.89	4.46	5.39
					Residual image		
Image	Reference	Cor.	LPC A	LPC B.	Std	O^{th} Entropy	Ratio
tm840903-b6	tm840903-b1	0.00	-0.01	156.48	6.89	4.47	5.27
tm840903-b6	tm840903-b2	0.04	0.07	152.52	6.88	4.44	4.94
tm840903-b6	tm840903-b3	0.46	0.35	136.05	6.11	4.44	3.68
tm840903-b6	tm840903-b4	0.51	0.25	141.14	5.92	4.48	3.84
tm840903-b6	tm840903-b5	0.68	0.15	138.60	5.08	4.20	3.68
tm840903-b6	tm840903-b7	0.79	0.35	136.31	4.23	4.01	3.41

Table 9.11: tm840903 (6)

Image	Mean				Std	O^{th} Entropy	Ratio
tm840903-b7	53.92				15.35	5.59	2.37
					Residual image		
Image	Reference	Cor.	LPC A	LPC B.	Std	O^{th} Entropy	Ratio
tm840903-b7	tm840903-b1	0.25	0.72	-15.04	14.85	5.34	2.38
tm840903-b7	tm840903-b2	0.36	1.34	-1.98	14.33	5.18	2.40
tm840903-b7	tm840903-b3	0.78	1.31	-19.15	9.59	5.15	2.36
tm840903-b7	tm840903-b4	0.80	0.86	4.41	9.26	5.20	2.36
tm840903-b7	tm840903-b5	0.94	0.47	1.69	5.12	4.24	2.73
tm840903-b7	tm840903-b6	0.79	1.76	-219.48	9.43	5.25	2.31

Table 9.12: tm840903 (7)

Linear prediction & Scene tm950902

See Table 9.13 to Table 9.19.

Image	Mean				Std	O^{th} Entropy	Ratio
tm950902-b1	90.36				4.15	4.11	2.78
					Residual image		
Image	Reference	Cor.	LPC A	LPC B.	Std	O^{th} Entropy	Ratio
tm950902-b1	tm950902-b2	0.83	1.10	49.42	2.30	3.12	2.77
tm950902-b1	tm950902-b3	0.78	0.50	65.84	2.60	3.28	2.85
tm950902-b1	tm950902-b4	0.67	0.32	75.31	3.08	3.51	2.83
tm950902-b1	tm950902-b5	0.63	0.13	77.93	3.23	3.58	2.84
tm950902-b1	tm950902-b6	0.16	0.19	64.23	4.09	3.92	2.78
tm950902-b1	tm950902-b7	0.53	0.20	80.45	3.51	3.70	2.81

Table 9.13: tm950902 (1)

Image	Mean				Std	O^{th} Entropy	Ratio
tm950902-b2	37.07				3.12	3.74	3.43
					Residual image		
Image	Reference	Cor.	LPC A	LPC B.	Std	O^{th} Entropy	Ratio
tm950902-b2	tm950902-b1	0.83	0.63	-19.61	1.73	2.73	3.16
tm950902-b2	tm950902-b3	0.92	0.45	15.17	1.21	2.23	3.70
tm950902-b2	tm950902-b4	0.82	0.29	23.16	1.78	2.75	3.61
tm950902-b2	tm950902-b5	0.78	0.12	25.39	1.95	2.88	3.57
tm950902-b2	tm950902-b6	0.23	0.20	9.36	3.04	3.49	3.42
tm950902-b2	tm950902-b7	0.68	0.19	27.57	2.30	3.12	3.48

Table 9.14: tm950902 (2)

Image	Mean				Std	O^{th} Entropy	Ratio
tm950902-b3	48.80				6.42	4.65	2.73
					Residual image		
Image	Reference	Cor.	LPC A	LPC B.	Std	O^{th} Entropy	Ratio
tm950902-b3	tm950902-b1	0.78	1.21	-60.16	4.03	3.87	2.49
tm950902-b3	tm950902-b2	0.92	1.90	-21.47	2.48	3.20	2.76
tm950902-b3	tm950902-b4	0.92	0.67	16.93	2.58	3.21	3.12
tm950902-b3	tm950902-b5	0.87	0.28	22.13	3.20	3.56	2.95
tm950902-b3	tm950902-b6	0.25	0.45	-13.71	6.22	4.48	2.72
tm950902-b3	tm950902-b7	0.73	0.43	27.77	4.41	4.00	2.81

Table 9.15: tm950902 (3)

Image	Mean				Std	O^{th} Entropy	Ratio
tm950902-b4	47.63				8.73	5.02	2.53
					Residual image		
Image	Reference	Cor.	LPC A	LPC B.	Std	O^{th} Entropy	Ratio
tm950902-b4	tm950902-b1	0.67	1.42	-80.83	6.52	4.51	2.29
tm950902-b4	tm950902-b2	0.82	2.31	-38.08	5.01	4.16	2.45
tm950902-b4	tm950902-b3	0.92	1.25	-13.57	3.53	3.66	2.76
tm950902-b4	tm950902-b5	0.88	0.39	10.56	4.17	3.91	2.68
tm950902-b4	tm950902-b6	0.16	0.39	-6.56	8.68	4.89	2.52
tm950902-b4	tm950902-b7	0.66	0.53	21.46	6.60	4.52	2.52

Table 9.16: tm950902 (4)

Image	Mean				Std	O^{th} Entropy	Ratio
tm950902-b5	95.52				19.95	6.02	1.91
					Residual image		
Image	Reference	Cor.	LPC A	LPC B.	Std	O^{th} Entropy	Ratio
tm950902-b5	tm950902-b1	0.63	3.01	-176.89	15.55	5.63	1.78
tm950902-b5	tm950902-b2	0.78	4.98	-89.14	12.48	5.35	1.86
tm950902-b5	tm950902-b3	0.87	2.69	-35.93	9.93	5.08	1.98
tm950902-b5	tm950902-b4	0.88	2.00	0.39	9.46	5.02	2.01
tm950902-b5	tm950902-b6	0.37	2.04	-189.14	18.57	5.91	1.91
tm950902-b5	tm950902-b7	0.89	1.63	15.55	9.08	4.94	2.06

Table 9.17: tm950902 (5)

Image	Mean				Std	O^{th} Entropy	Ratio
tm950902-b6	139.79				3.58	3.89	6.01
					Residual image		
Image	Reference	Cor.	LPC A	LPC B.	Std	O^{th} Entropy	Ratio
tm950902-b6	tm950902-b1	0.16	0.14	127.22	3.53	3.66	4.80
tm950902-b6	tm950902-b2	0.23	0.26	130.16	3.48	3.65	4.71
tm950902-b6	tm950902-b3	0.25	0.14	133.02	3.46	3.64	4.81
tm950902-b6	tm950902-b4	0.16	0.06	136.73	3.53	3.67	5.21
tm950902-b6	tm950902-b5	0.37	0.07	133.54	3.33	3.59	4.63
tm950902-b6	tm950902-b7	0.55	0.18	130.90	2.98	3.45	4.11

Table 9.18: tm950902 (6)

Image	Mean				Std	O^{th} Entropy	Ratio
tm950902-b7	49.118				10.92	5.26	2.22
					Residual image		
Image	Reference	Cor.	LPC A	LPC B.	Std	O^{th} Entropy	Ratio
tm950902-b7	tm950902-b1	0.53	1.40	-77.22	9.25	4.96	2.12
tm950902-b7	tm950902-b2	0.68	2.36	-38.31	8.05	4.79	2.19
tm950902-b7	tm950902-b3	0.73	1.24	-11.15	7.50	4.70	2.28
tm950902-b7	tm950902-b4	0.66	0.82	10.08	8.19	4.81	2.27
tm950902-b7	tm950902-b5	0.89	0.49	2.62	4.97	4.09	2.51
tm950902-b7	tm950902-b6	0.55	1.69	-186.41	9.11	4.96	2.22

Table 9.19: tm950902 (7)

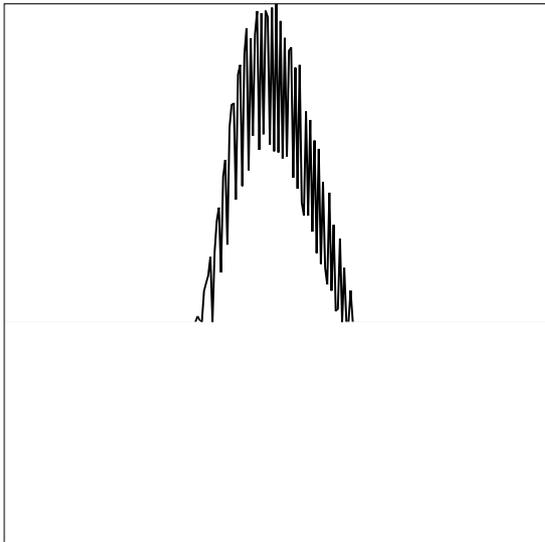


Figure 9.2: Histogram of a CIR image

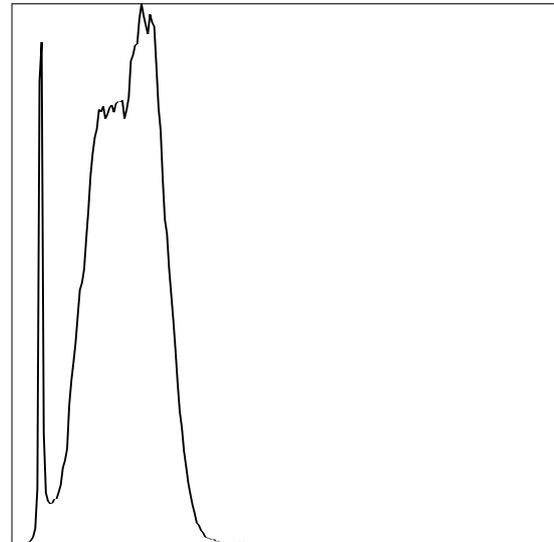


Figure 9.3: Histogram of satellite image

9.3 Influence of scanning on compression

Figure 9.2 shows the histogram of a part of the infrared band of a colour infrared airborne image. Figure 9.3 shows the histogram on a part of the infrared band of a TM image. The CIR histogram shows a jitter, unlike the TM histogram.

Figure 9.4 presents a row of the colour infrared image, and Figure 9.5 a row of the Thematic mapper image.

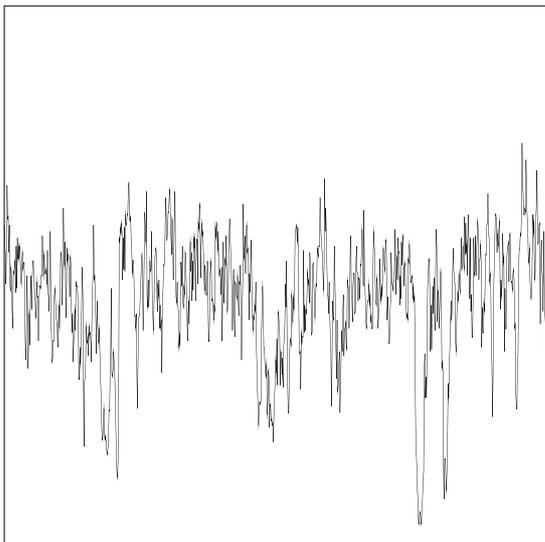


Figure 9.4: One row in CIR image

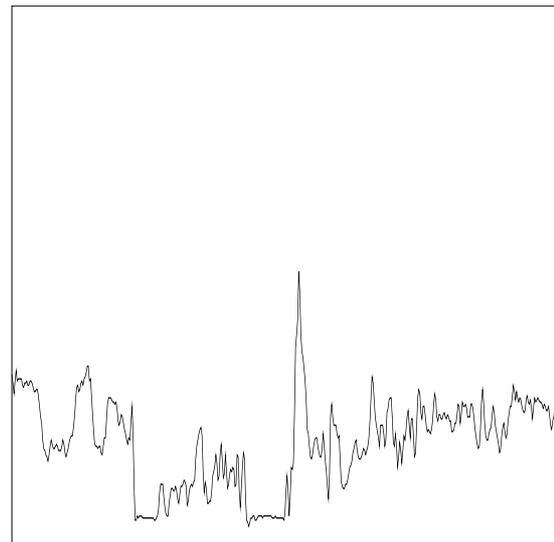


Figure 9.5: One row in TM image

Finally, Figure 9.6 shows the Fourier transform of the colour infrared image, and Figure 9.7 the Fourier transform of the Thematic Mapper image. There are far more high frequencies in the airborne image than in the satellite image. These high frequencies make the compression

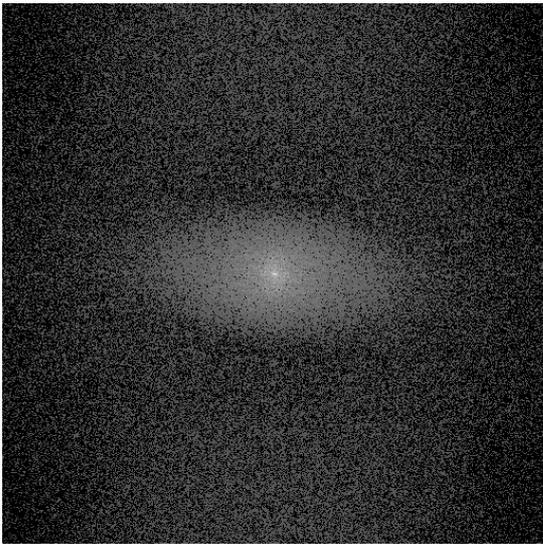


Figure 9.6: FFT of CIR image

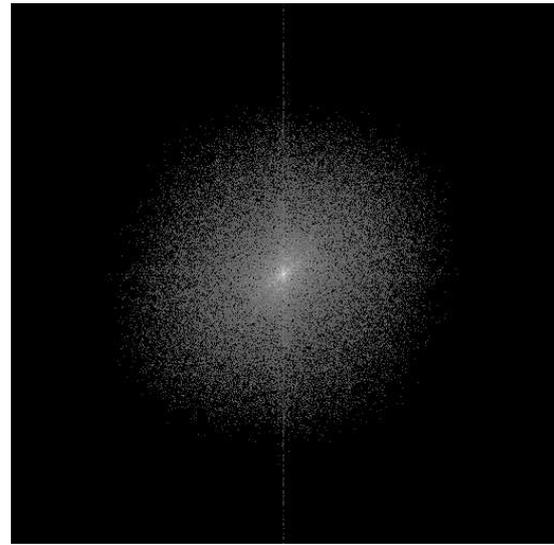


Figure 9.7: FFT of TM image

difficult. They come partly from the high resolution of the scanning of the image.

A way to solve this problem in the case of lossy compression is to prefilter the image before the compression. The compression ratio becomes then similar to the ratio of TM images. Figure 9.8 shows the Fourier transform of a prefiltered colour infrared image. High frequencies have been reduced.

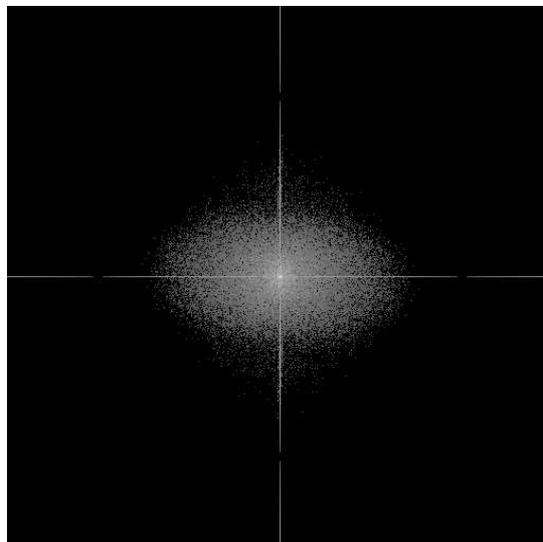


Figure 9.8: FFT of a prefiltered CIR image



Figure 9.9: Original image

9.4 Pole detector on compressed image

The pole detection algorithm described in section 5.2 works on the luminance image. To test the effect of compression artifacts on the performance of that detector, a typical part of the Buzi minefield presenting four poles has been selected. The corresponding image is presented in Figure 9.9. The luminance image has then been computed and compressed with various compression ratios. A compression ratio of 1.41 was reached with the lossless method.

Table 9.20 presents the compression ratio, the maximum error and the RMS error for 6 levels of lossy compression (polesLossyN). The number N represents the number of bit planes that were discarded.

The results obtained when a prefiltering is applied are presented in Table 9.21. The results are worse in term of MAE and RMSE, but visually, the compressed image is much closer to the original. This is due to the inadequacy of RMSE to measure the visual quality of an image.

Scenes	Compression Ratio	MAE	RMSE
poles.rma	1.41	0	0
polesLossy1	1.48	1	0.41
polesLossy2	1.77	4	0.98
polesLossy3	2.34	9	1.82
polesLossy4	3.69	17	3.78
polesLossy5	8.72	34	7.49
polesLossy5_6	18.10	47	9.20
polesLossy6	31.74	60	10.45

Table 9.20: Compression ratios

Scenes	Compression Ratio	MAE	RMSE
poles.rma	2.86	0	0
polesLossy1	3.30	222	15.25
polesLossy2	5.26	222	15.34
polesLossy3	9.76	222	15.54
polesLossy4	15.22	222	15.70
polesLossy5	24.55	220	16.03
polesLossy6	38.31	222	16.56

Table 9.21: Compression ratios with prefiltering

The pole detector has been launched on all the compressed/decompressed images. Even with the highest compression ratio (38.31) all poles were still detected. We may thus conclude that the pole detector is highly robust to compression artifacts.

Figure 9.10 presents the results of the pole detector when applied to most compressed image with lossless compression. On Figure 9.11, the results on the most compressed image with lossy compression without prefiltering (polesLossy6). Results with prefiltered images are similar.

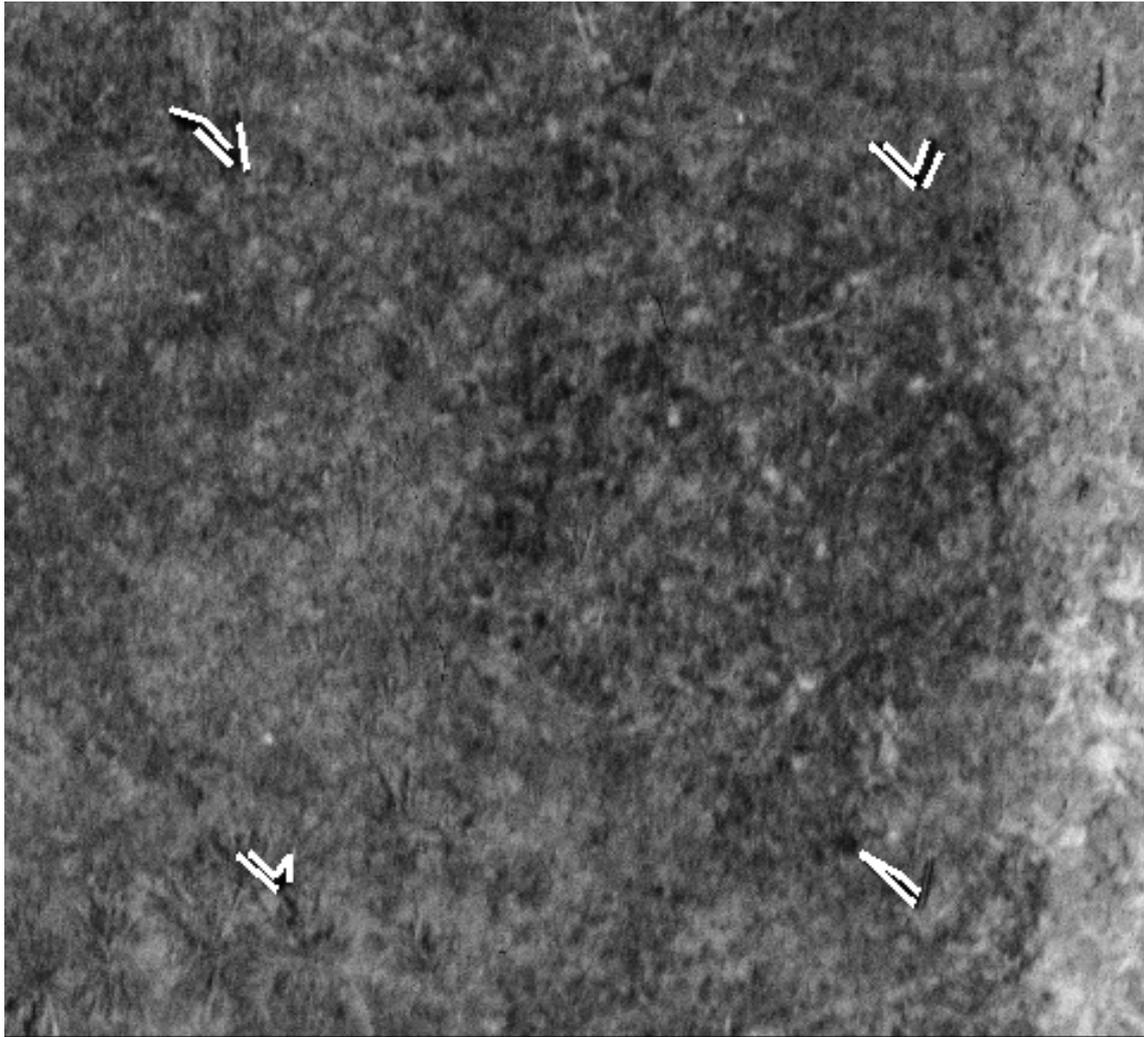


Figure 9.10: Results on compressed image (lossless)

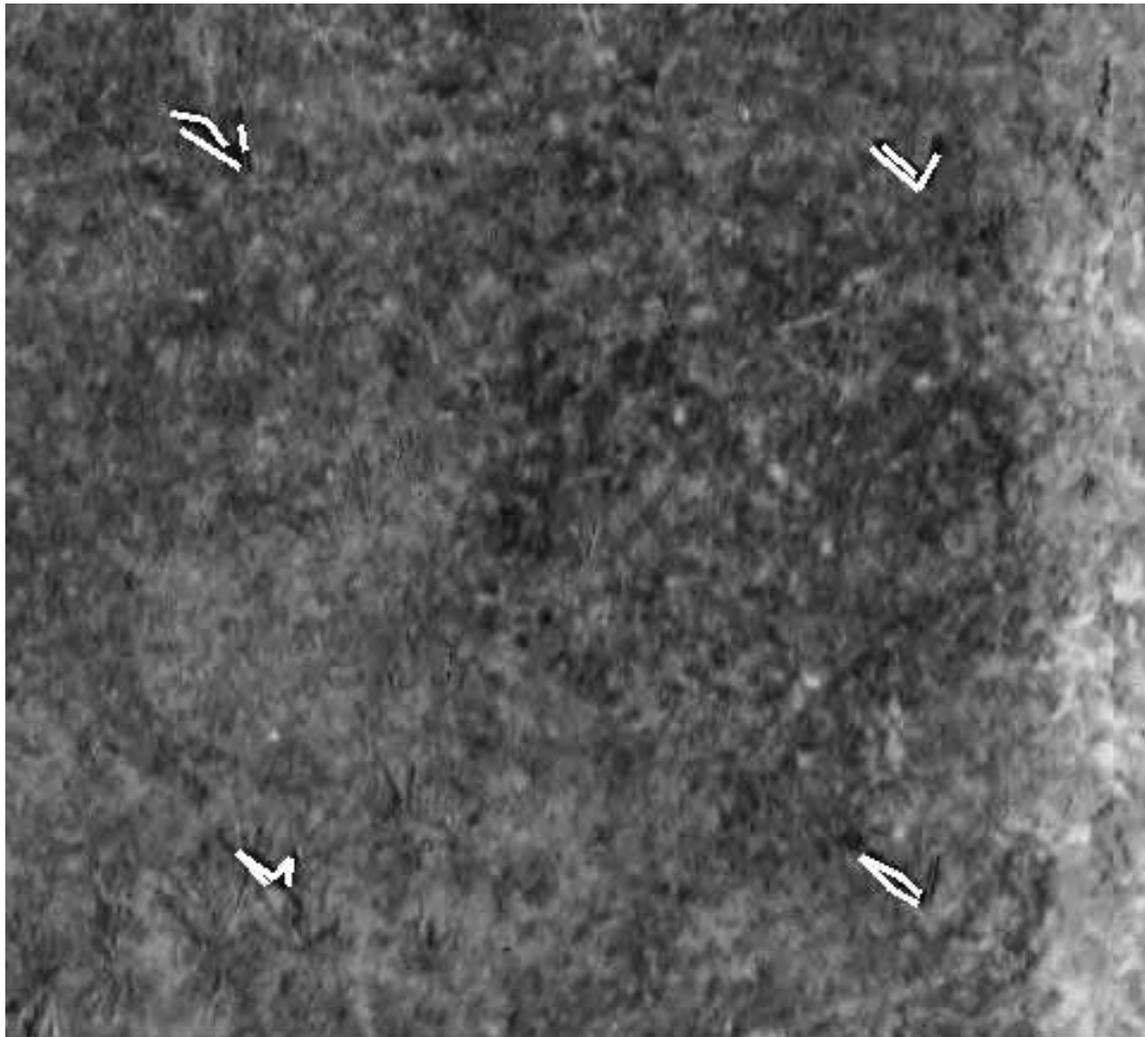


Figure 9.11: Results on the compressed image (lossy)

9.5 Conclusion

When comparing the compression ratio obtained using lossless compression on Landsat and airborne colour infrared images, it appears, that the compression ratio is much lower for airborne colour infrared (1.4) than for Landsat images (3.14). This may be explained by the higher textural content of the airborne images. This high textural content is partly due to the high resolution used for scanning in order to retain the fine details of the images but that scanning resolution is also at the limit of the grain and the grain texture begins to appear.

For those image, lossy compression should thus be used to reach a significant reduction of data volume. Typically, up to a compression ratio of ten, the degradation of the image is hardly visible. Of course, the precise degradation limit is function of the kind of details that are used by the photo interpreter as mine indicators.

When semi-automatic detection of minefield indicators are considered, the degradation limit and the corresponding compression ratio are function of the indicator and the algorithm used for the detector. Typically, for the pole detector presented in section 5.2 even compression ratio as high as 40 did not hinder the detector.

For the multi-spectral images (Landsat or colour), spectral correlation may be used to increase the compression ratio. This approach has been evaluated on the Landsat images. Even though the correlation between the bands is rather high, the increase of the compression ratio remains rather limited. Typically, for lossless compression the compression ratio is increase by 10 %. This may be explained by the fact that even if the prediction is good, the variance of the residual errors is small but the spatial correlation is much lower than the original image (the criterion of the compression does not take into account the spatial correlation) and the gain of spatial decorrelation is thus reduced.

Chapter 10

Graphical user interface

It is the final stage of a semi-automatic image processing tool to be integrated into a complete environment where the user can launch tools and study the results in the most interactive way. Graphical User Interfaces (GUI) and Geographical Information Systems (GIS) are such tools. In this chapter, the dedicated features of our GUI that were used for the visual inspection of the database are presented.

The RMA/SIC visualization tools is able to display *virtual* images such as image mosaics. These *virtual* images are made of the composition of several sub-images stored on the disk. The advantage of our tool is that the co-registration of the sub-images and the mosaic is performed in real-time without using temporary disk space.

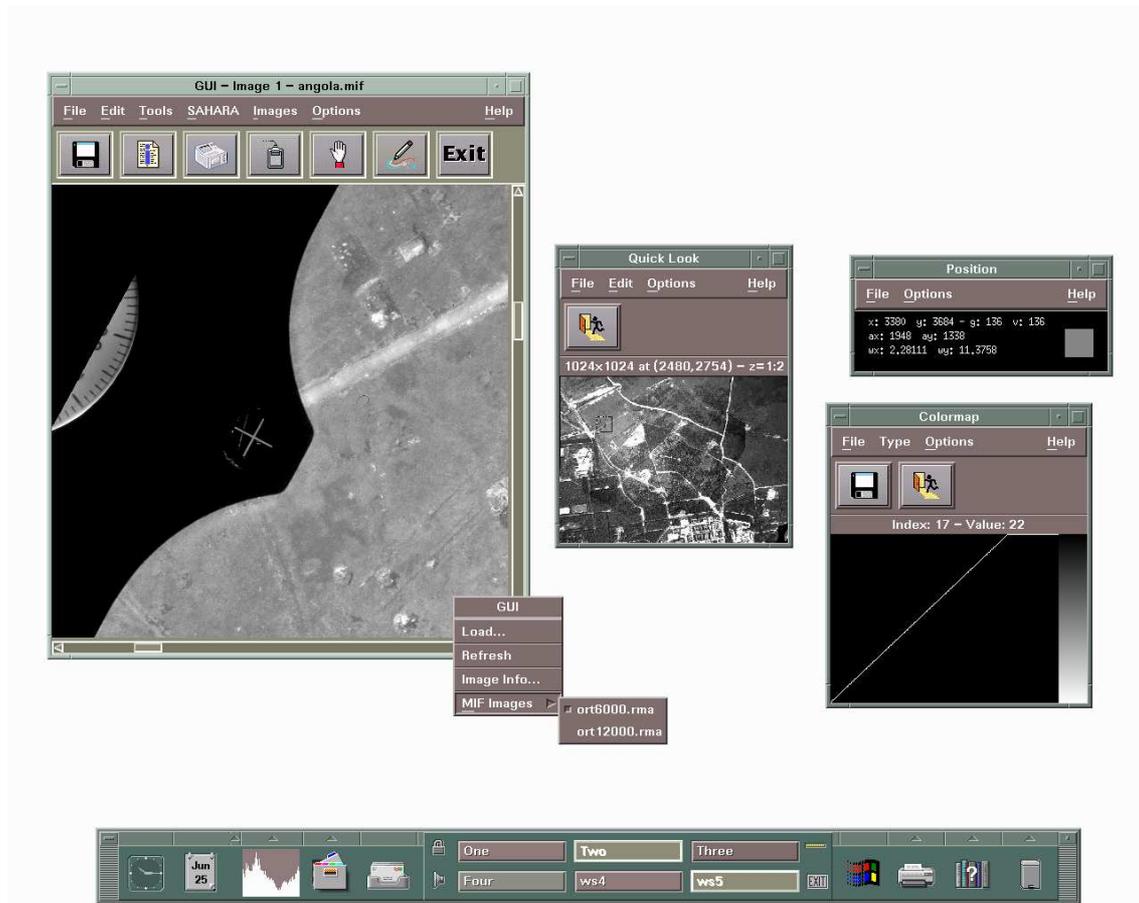


Figure 10.1: Part of a mosaic image

Figure 10.1 shows part of a *virtual* image made of *ort6000* and *ort12000*, two images from Aerodata. Only the *ort6000* part of the mosaic is activated.



Figure 10.2: Mosaic image with all sub-images activated

In figure 10.2, the *ort12000* part of the mosaic is also activated. The *orth6000* part is in front of the *ort12000* one.

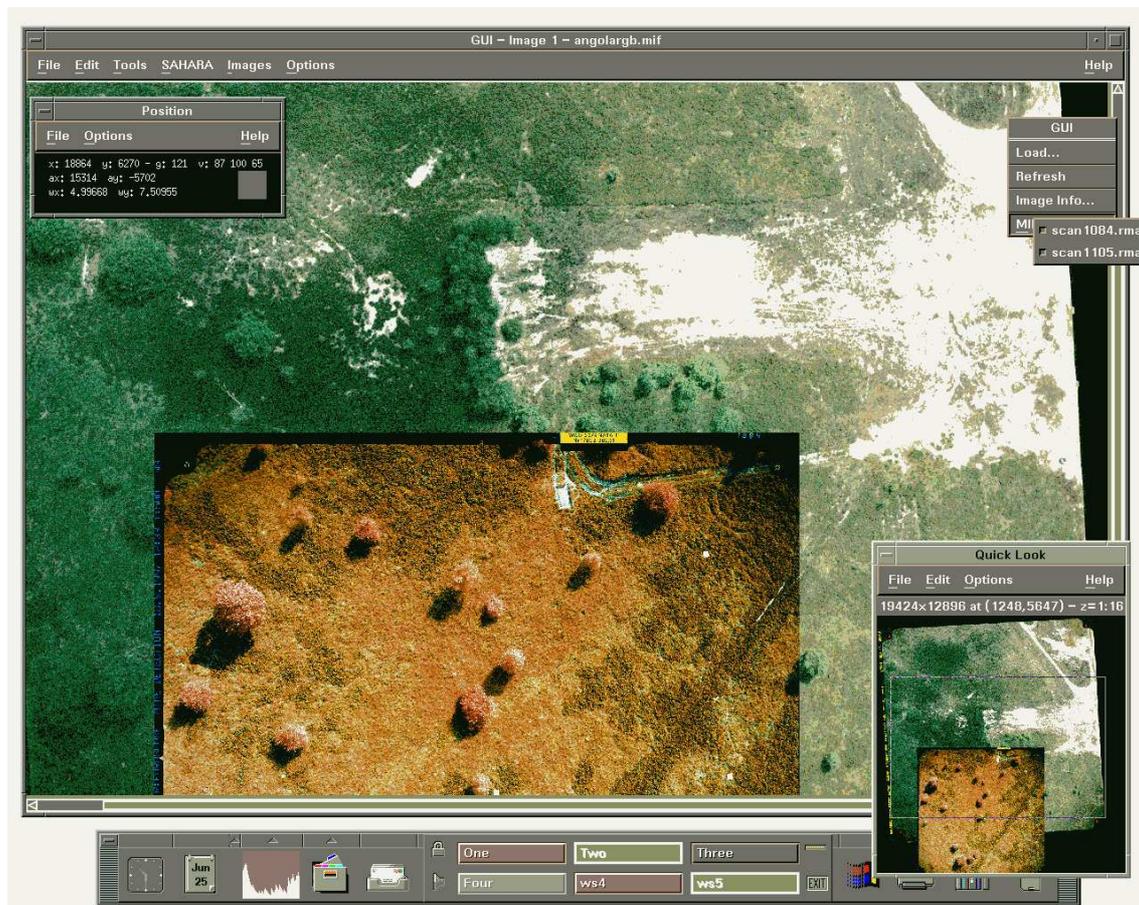


Figure 10.3: Large mosaic image

In figure 10.3, a large image mosaic made of *scan1084* and *scan1105*, two images from Eurosense is shown.

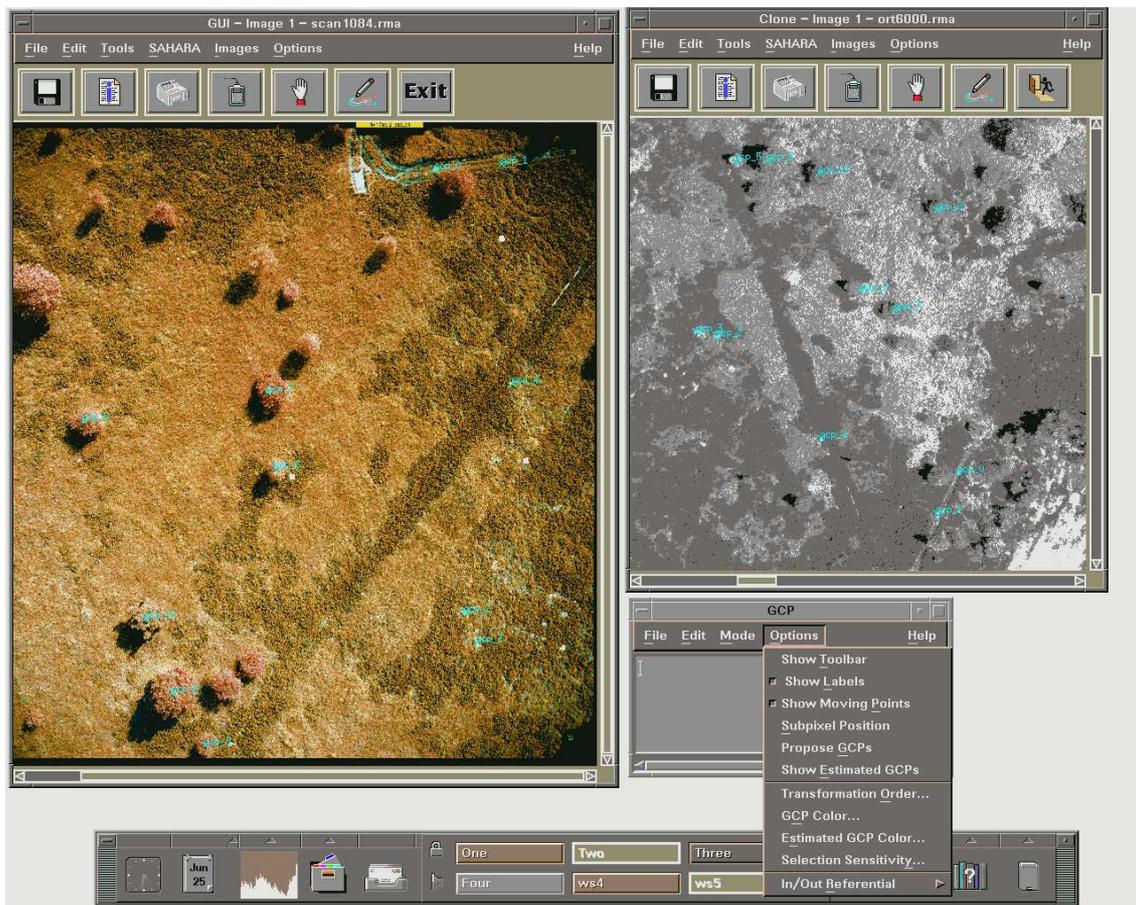


Figure 10.4: Snapshot of the registration GCP selection process

Our visualization tool also allows the user to geocode / register one or more images by means of a Ground Control Point (GCP) tool. These GCP are used to compute the parameters of a polynomial transformation. These parameters are then stored in an ASCII header file that is associated with the original image. The pair made of the original image file and the header file allows the visualization tool to register interactively the image to any other image projection. Figure 10.4 shows a snapshot of the registration GCP selection process.

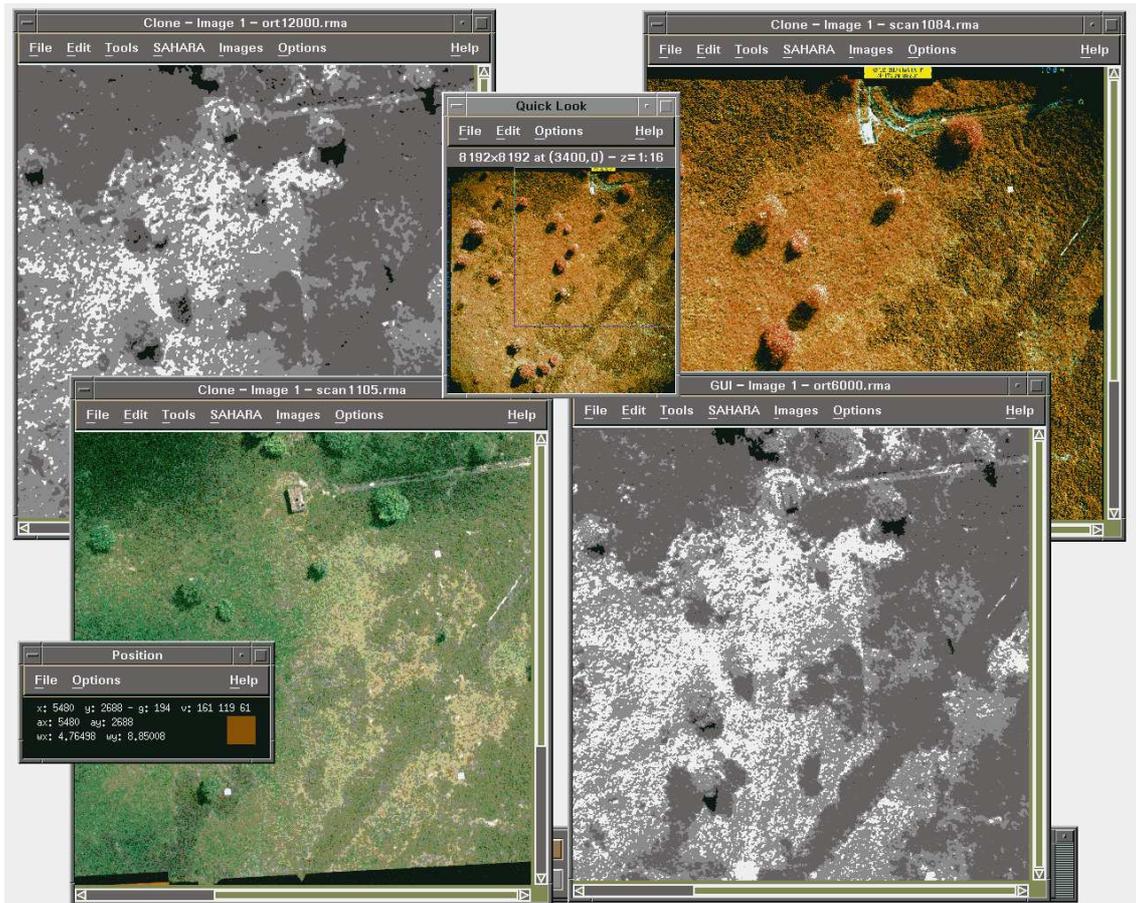


Figure 10.5: Interactive registration process

Figure 10.5 shows the interactive registration process; *ort6000*, *ort12000* and *scan1105* images are co-registered in real-time to the projection of the *scan1084* image.



Figure 10.6: Interactive registration process on *virtual* images

Figure 10.6 shows that the same operation can be performed on *virtual* images. Here the image mosaic made of the Aerodata images (*ort6000* and *ort12000*) is co-registered with the image mosaic made of the *scan1084* and *scan1105* RGB images.

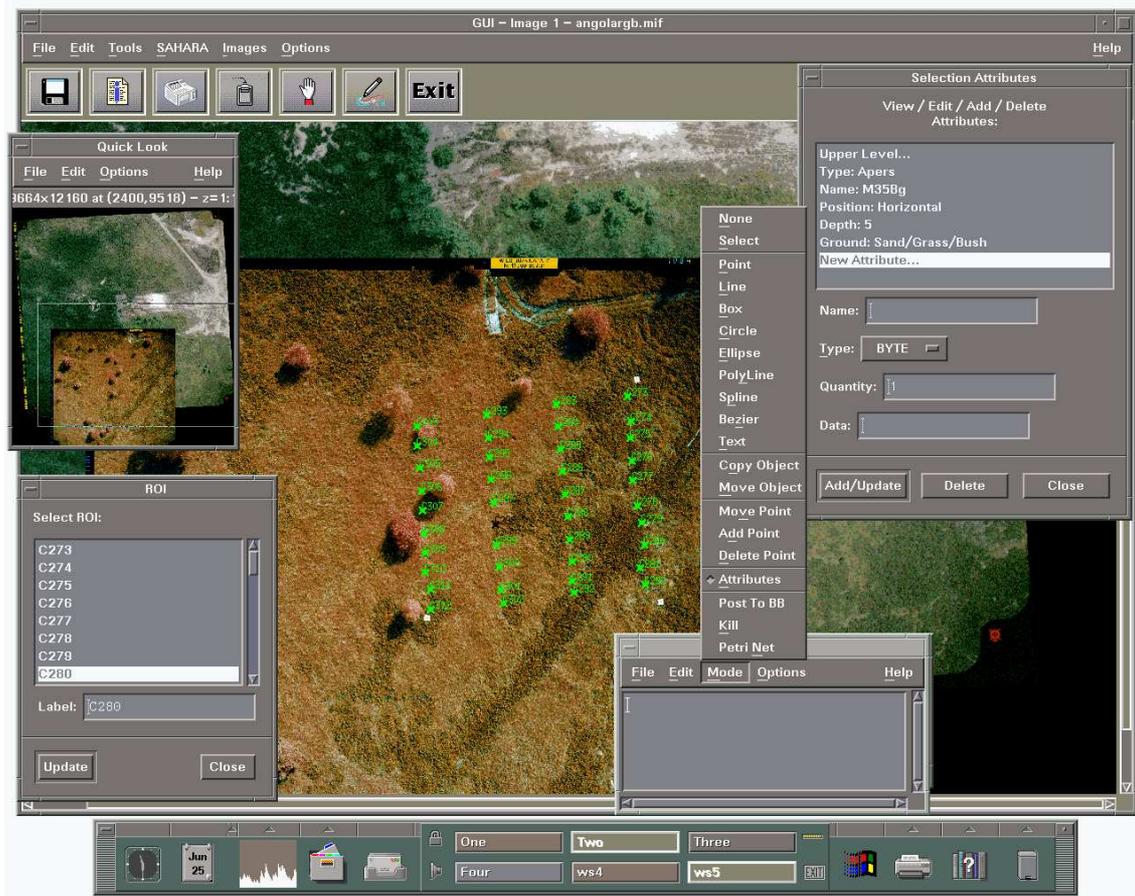


Figure 10.7: Vectorial information

Our visualization tool is able to display vectorial information on top of images such as GIS data. The user can edit these vectorial objects, define object attributes, export the data, etc. Some of these objects may define a region of interest; the user has the ability to move around these regions of interest.

Figure 10.7 shows the mines present in the test field C. Mines characteristics are encoded as vectorial object attributes.

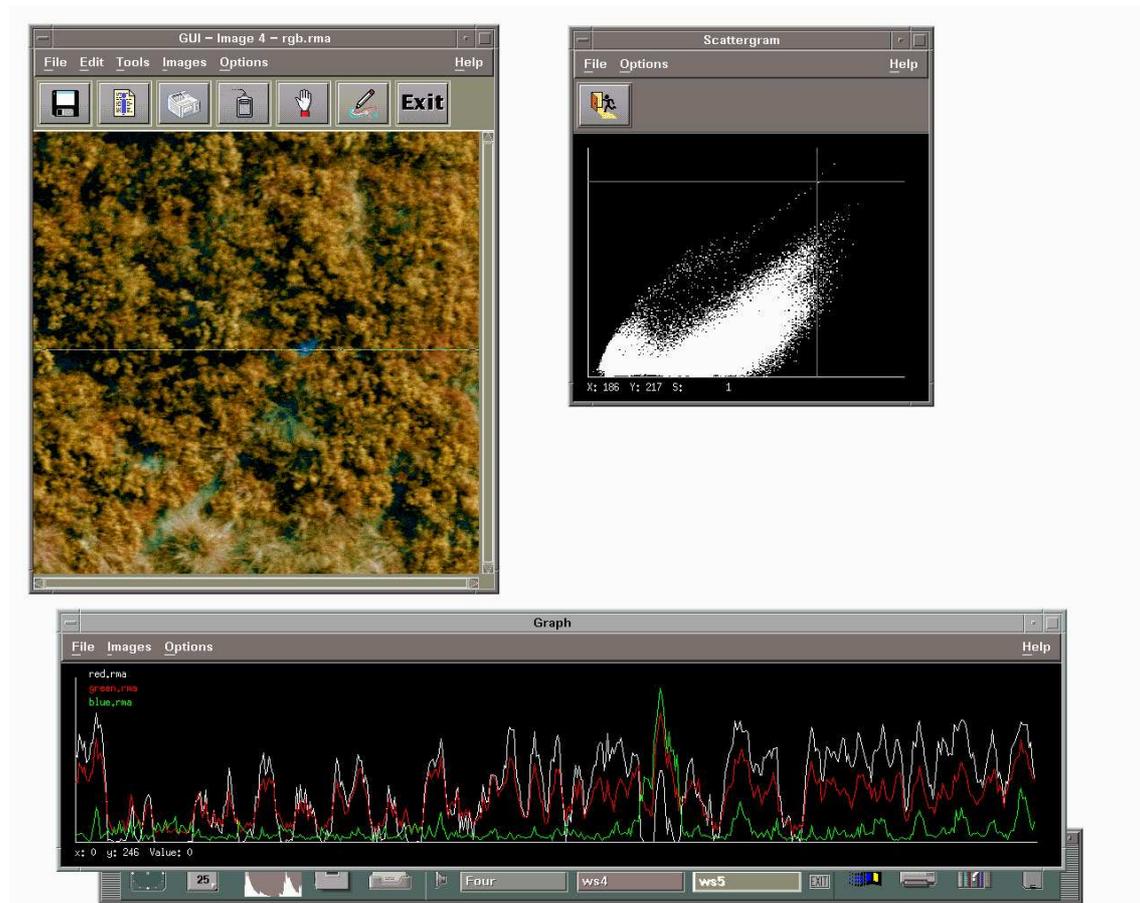


Figure 10.8: Scattergram and graph tool

Several analysis tools are available in our environment. The Scattergram tool displays a 2D histogram of occurrences of grey level pairs using the data from two different spectral bands. This tool is useful for multi-spectral image analysis. The Graph tool displays a grey level cut (profiles) of different images. Figure 10.8 shows part of the colour infrared scan1084 images and the scattergram and graph computed on it. The scattergram is computed using the green and blue band of the colour infrared RGB images.

Chapter 11

Recommendations

Based on the experience obtained in this project, it is possible to draw some conclusion on how to choose good minefield indicators, build a good detector and collect the data, to improve the image processing work.

11.1 How to choose the indicators

As already mentioned, indicators must fulfill two conditions: be valid indicators for a photo-interpreter and be simple enough to allow an automatic detection. The first constraint means that the presence of the indicator must really improve the confidence in the presence of a minefield. The second means that a detector, as defined in the next paragraph, can be implemented. It is also a good idea, but not a necessity, to choose indicators which would take a long time to be detected by a human interpretation, because of their small sizes for instance. That way, image processing could really be a valuable addition to the tools of the photo-interpreters.

11.2 How to build the detectors

Once the indicators have been chosen, the detectors must be built. If the detector is to be launched prior to any human investigation, such as a pole detector presented previously, then it must be able to work on very large images. In the examples presented here, the pole detector and the anti-tank mine detector divide large images into several smaller overlapping ones, perform the detection on all these sub-images and then merge the results.

The speed of the detector is also important. If the detector is to speed up the work of a photo-interpreter, then it must run on a reasonable time using reasonable computation power for economical reasons. This limit is only a qualitative guideline and it is not possible to precisely define it, since the computing power at a given cost is increasing fast. Any algorithm could probably be used if the computation power is made available, but if the method used must be economically viable, then the complexity of the algorithm must remain limited.

11.3 How to collect the data

Obviously, the data must be collected with their use kept in mind. Once the indicators have been chosen, then the most adequate sensors can be known, both in terms of principles and

resolution. The detectors can also provide some constraints. For instance, the pole detector presented here uses the shadow of the poles. Images taken when the shadows are clearly visible will help the detection, typically the flights should not take place at night or at midday. For the use of stereovision, a delicate phase is the rectification, whether it is performed manually or automatically. If the two stereoscopic images could be taken in such a way that the images were originally rectified then again the quality of the result of the detector would be improved.

Bibliography

- [1] J A Craib. Mine detection and demining from an operator's point of view. In *Workshop of anti-personnel mine detection and removal*, Lauzanne, 1995.
- [2] Pascal Druyts, Yann Yvinec, and Marc Acheroy. Usefulness of semi-automatic tools for airborne minefield detection. In *Clawar'98, First International Symposium*, pages 241–248, Brussels, Belgium, 1998.
- [3] Olivier Faugeras, B Hotz, H Mathieu, T Vieville, Zhengyou Zhang, P.Fua, E Theron, M Laurent, G Berry, J Vuillemin, P Bertin, and C Proy. Real time correlation based stereo: algorithm implementations and applications. Technical Report RR-2013, 2004 route des Lucioles B.P. 93 F-06902 Sophia-Antipolis France, 1993.
- [4] P J Huber. *Robust Statistics*. John Wiley & Sons, New York, 1981.
- [5] D E Lake, B Sadler, and S Casey. Detecting regularity in minefields using collinearity and a modified Euclidean algorithm. In *SPIE, Detection and Remediation Technologies for Mines and Mine like Targets II*, Abinash C. Dubey; Robert L. Barnard; Eds, volume 3079, July 1997.
- [6] K Scheerer. Airborne multi-sensor system for the autonomous detection of landmines. In *SPIE, Detection and Remediation Technologies for Mines and Mine like Targets II*, Abinash C. Dubey; Robert L. Barnard; Eds, volume 3079, July 1997.
- [7] J L van Genderen and B H P Maathuis. Airborne detection of landmines: a review of the techniques and some practical results. In *DGP session, Disarmament and international security*, Regensburg, Germany, March 1998.
- [8] Yann Yvinec and Bernard Thiesse. Estimating the fundamental matrix from phase based 2D disparity. In *Second Asian Conference on Computer Vision ACCV'95*, Singapore, December 1995.