Technical Report

Segmentation and Classification of Landcover Areas using a Polymorphic Feature Extraction

December 2, 1999

TB-ipb-99/4

Stephan Heuel, Annett Faber
Institute of Photogrammetry
Nussallee 15
53115 Bonn

Abstract

This report demonstrates the use of FEX, cf. **?** for segmenting land-use units from remote-sensed images and their classification to meaningful clusters(?!).

Two approaches for segmenting land-use units are proposed, one is based on symbolic data and one is based on iconic data. Advantages and disadvantages of both methods are discussed. Problems of the method and the output of FEX, which appeared during this work are discussed.

The classification is based on a linear classifer, which supplies classified areas according to their agricultural use. Results, demonstrating the the feasibility of the process, are shown and discussed.

Contents

Chapter 1

Introduction

This report discusses the segmentation and classification of land-use units from remote-sensed images. An image of land-use units consists of a set of homogeneous areas surrounded by either a street or borders of other homogeneous areas. The homogeneity criterion can be based on the grey values or on the multi-spectral properties of the image. Since we assume the landscape to be approximately planar, we do not have to deal with 3D information.

Previous work at the IPB, see example **?**, use structural modeling of land-use fields based on an hierarchical polygonal model. The structural information is used to guide the segmentation process using the MDL principle and guarantees a consistent and complete output.

Now we want to apply a segmentation based on a more generic segmentation model, which was introduced by **?** and is motivated by an object model consisting of regular faces that are surrounded by lines and points either being singular or formed by intersecting lines. This way we are not restricted to a polygonal segmentation procedure.

Then the classification scheme is realized by using a linear classifier, especially the Nearst-Neighbor-Classifier with rejection class. The result of this process are classified areas, according to their agricultural use.

Chapter 2

Feature-Based Segmentation

Our segmentation approach for land-use units does not directly work with the image but uses a generic feature extraction procedure which does not include any specific knowledge about the structure of the scene. Instead, we want to analyze the output of the feature extraction procedure. We use a polymorphic feature extraction as proposed in ?.

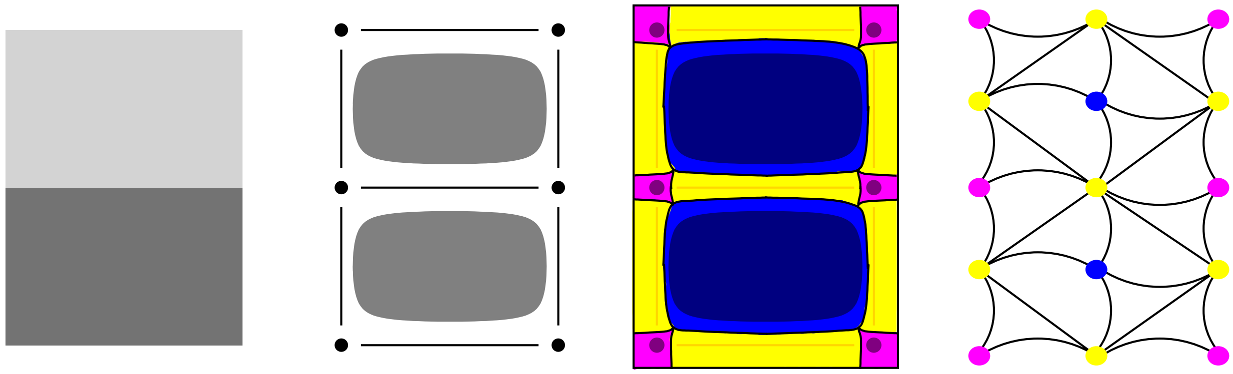
2.1 Polymorphic Feature Extraction

The Feature Extraction FEX replaces a digital image with a first symbolic description, namely a list of points, a list of lines and a list of blobs. Because we have geometric objects of different dimensions, we call the extraction scheme polymorphic. Each geometric object in a list contains several attributes describing its properties, e.g. the covariance matrix of a point or the mean grey level of a blob. Note that since the features are extracted simultaneously, they are non-overlapping.

We can also directly obtain the neighborhood relations between the features by analyzing the voronoi diagram of all image areas that are not classified by features, see figure ??(c). Now it is possible to define the output of FEX as a graph with nodes as features and arcs as neighborhood relations between features, we call this graph feature neighborhood graph, FNG[1].

The input of the FEX system consists of a one- or multi-channel image and a set of parameters, where the most dominant are three scale parameters σ_R, $sigma_L$ and $sigma_P$: σ_R defines the overall image resolution and mainly reduces the noise. σ_L and $sigma_P$ define the significance of a line resp. point. note that because of the algorithm used in FEX the following rule holds: large σ_L/σ_P imply large exoregions.

[1]Note that in previous publications it was called feature adjacency graph, FAG



(a) (b) (c) (d)

Figure 2.1: Artificial example for FEX. Figure (a) depicts the input image of two rectangular areas; (b) shows the extracted features; (c) displays the exoskeleton (or voronoi-diagram of the background) in red; so-called exoregions of blobs are blue, exoregions of lines are yellow and exoregions of points are magenta. Neighboring exoregions define the neighborhood relationship of its features; (d) shows the feature neighborhood graph FNG

2.2 Contextual Interpretation of the Segmentation

Now we want to interpret the results of FEX in our context of extracting land-use units. Clearly these units correspond to blobs within FNG, the symbolic description of FEX. But just the correspondence of blobs to units does not give the full solution to our problem: the boundaries of the blobs generally are ragged and do not correspond to the exact boundaries of the land-use unit. This is because a boundary consists of lines and points which are explicitly represented in the FNG. Fortunately we directly can extract the boundary of a blob via the relations defined in the FNG.

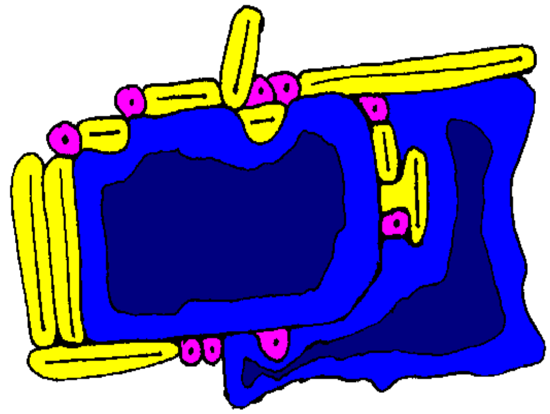
So the main approach is to identify blobs corresponding to land-use units and to extract the boundaries of these blobs. Using this approach we may not be able to obtain a complete description of the scene, since we have not modeled streets etc. Therefore it is possible to have gaps between to units which could—but not necessarily have to—correspond to streets.

The identification of blobs can be done using the knowledge of a minimum size of a land unit, let's say 100qm. We can convert this number to pixel-units and take it as a threshold for rejecting blobs that are too small to be land-use units.

The extraction of the boundaries is more involving, we suggest two approaches: the first deals only with the symbolic description, that is output by FEX, this mainly includes the FAG and an ordered sequence of neighbors for each blob. The latter sequence can be acquired by tracing the boundary of the exoregion of a blob. The second approach uses the exoskeleton image itself and extend every blob to the boundary of its exoregion.

It cannot be expected that the features and their neighborhood relations from FEX are consistent to an ideal image description. For a structured analysis of relational errors see cf. ? and ?. As a motivating example and to illustrate the problems for grouping blob-boundaries, we give an artificial example of a typical result of FEX for low quality images (i.e. images that do not fit to the image model of fex, for example having textures), cf. figure ??. Some features that are neighbored to the upper left blob apparently do not belong to the boundary, some feature types are neighbored which should not be, e.g. two points or two blobs.

(a)



(b)

Figure 2.2: (a) Artificial example of a possible result of FEX. Note that there are neighborhood relationships, that do not occur in the ideal image model, see ?; (b) the exoskeleton regions of the features in (a) defining neighborhood relationships.

2.2.1 Symbolic Approach

This approach is motivated by the grouping strategy FAGANA (feature adjacency graph analysis), which is described in ?. FAGANA claims to be a topological grouping method, i.e. that main parts of the reasoning is done using topological aspects. It uses the ideal image model and knowledge about the algorithm and used parameters of FEX.

We adopt this idea, but concentrate on a blob-focused approach: separately for every blob we group only those points and lines, which are adjacent to this blob. This corresponds to the blob-induced grouping mentioned in ?, p. 119f. But now our goal is to guarantee a closed sequence of points and lines for every blob, while the blob-induced grouping does an independent grouping of pairs of features and does not necessarily result in a closed sequence.

First we have to obtain an ordered sequence of features that are neighbored to one blob. This cannot be retrieved by the FNG since the graph-arcs are not ordered[2]. This sequence

[2]it would be interesting to incorporate a graph-arc order which preserves the topological order of features in the image

of features is computed in a newer version of FEX (> 1.52) and is part of the symbolic description. In figure ??(a) the ordering is shown by the numbers from 1 to 18.

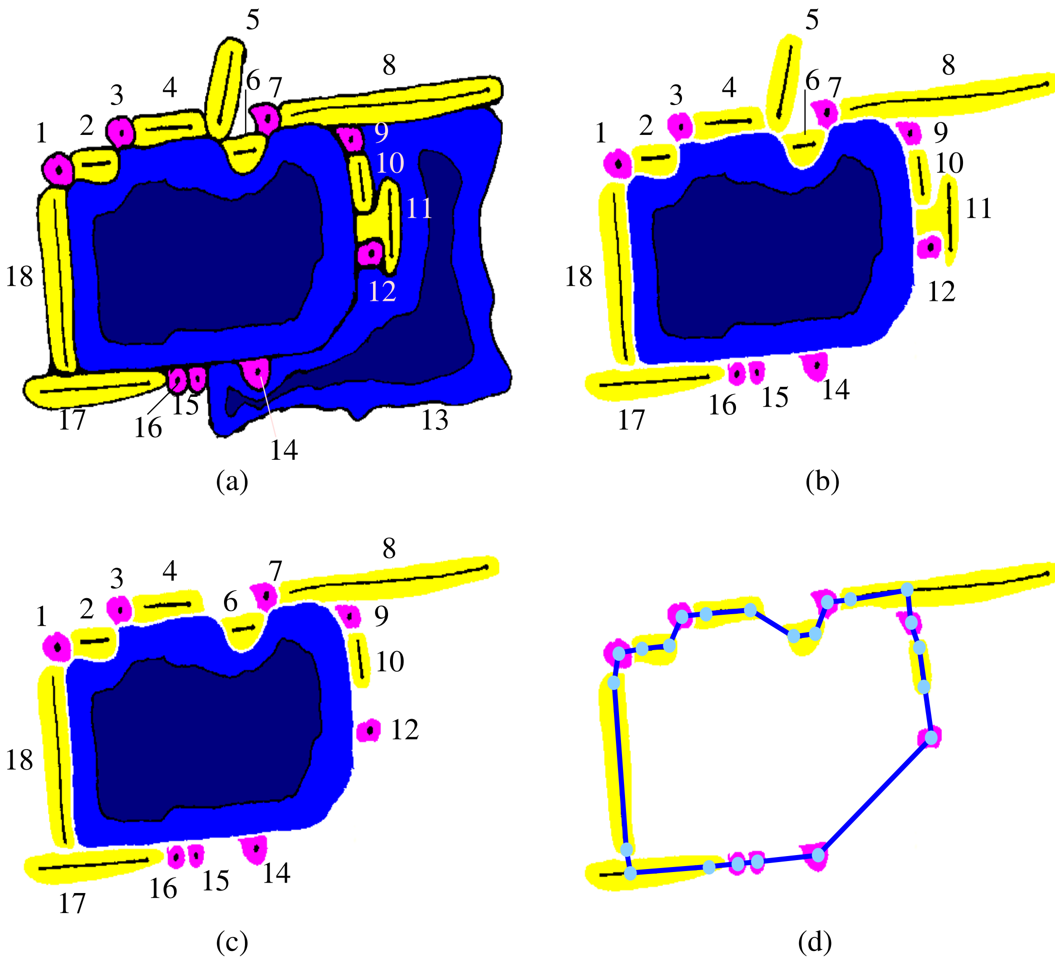


Figure 2.3: (a) Exoregions of set of neighbored features to upper left blob; (b) only points and lines neighbors; (c) as figure (b), but removed lines with insufficient neighborhood; (d) polygon computed using lines and points from figure (c)

Elimination of sequence elements. Since we are only interested in points and lines, we will eliminate blob neighbors from the sequence, as blob 13 in figure ??(b). If the common exoskeleton line of the two neighbored regions is long, we may loose a significant part of the blob boundary. On the other hand, this part of the blob boundary is only available by the exoskeleton, which is not represented in a symbolic description format but only as an image, see also section ??. Furthermore the image boundary is not specifically included in the exoskeleton and therefore not included in the FNG. This may also lead to gaps in the blob-description. For the rest of the paper, we want to exclude blobs neighbored to the image boundary.

Then we might have lines that are adjacent to the blobs, but are not related to the blob boundary, because it shares only a small fraction of its exoregion with the blob exoregion, see figure ??(b), lines 5 and 11. We can eliminate these lines by setting a threshold T that depends on the input parameters image resolution σ_R and the line scale of σ_L, for example $T = (\sigma_R + \sigma_L)\pi$. If we'd know the position of the line inside the exoregion and the projections of the endpoints onto the common exoskeleton line, we could get an even better classification of boundary lines.

Pairwise grouping. After we have cleaned out the ordered sequence of point and line neighbors, we can do a pairwise check for the connectivity between a pair $C(f_1, f_2)$. Neglecting the order, we have to distinguish three cases. Note that we do not want to test on geometric classifications of pairs, like mentioned in ?, p. 173f. where one can find a detailed analysis of cliques of features according to their relative geometrical positions. We assume that we don't need this since the features are expected to be aligned to the blob boundary and some of the cases can not occur[3].

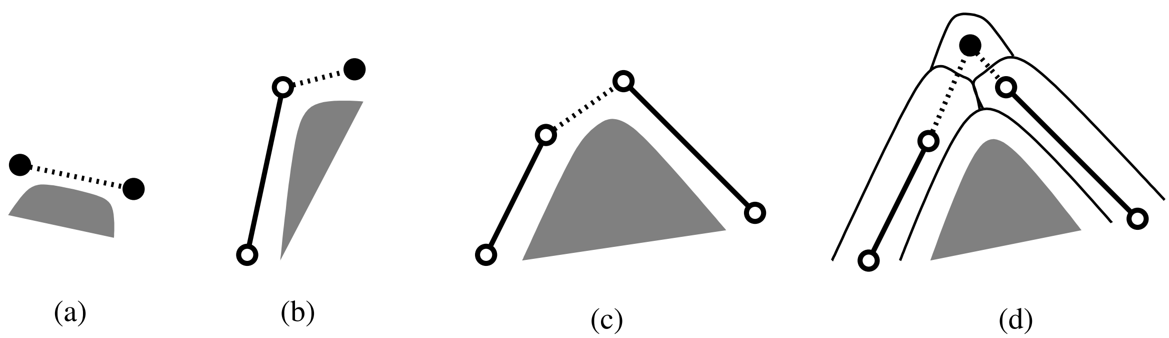


Figure 2.4: Examples of point-line pairs. (a) are two points, connected by a virtual (dotted) line, (b) shows one line and one point, (c) displays two lines, that are directly connected by a virtual line and in figure (d) the two lines have a common neighbor point, which is connected to each of the lines by a virtual line. Note that we neglect the classification of the relative geometric positions of the two features, see text.

- *(Point, Point).* If a point is followed by a point and both are neighbored to a blob, one can create a so-called virtual line between the points.
- *(Point, Line).* If a point is followed by a line, one has to determine the closest line-point to the extracted point and create a virtual line between them. Note that this requires a geometric test for closeness and not a topological test. One could think of constraints from the exoskeleton image in order to find the closest point; the FNG arc between the point and the line could indicate the position of the point regarding to the line using the projection of the common exoskeleton line onto the given line.

[3]also that FAGANA does not completely take advantage out of this analysis

At the moment we do not have this information in the symbolic description and therefore we can only use the mentioned geometric test. We may get problems, when the line-point is not the one we wanted to have, like in figure **??**(c) line 8 and point 9. We'll try to resolve this problem in the next step.

- *(Line, Line).* Two lines can be connected by a virtual line in the same way as above using a geometric test for closeness, with all its difficulties, this is case (c) in figure **??**.

  Sometimes it is possible that the exoregions of the two subsequent lines had displaced an exoregion of a point so that the neighborhood between this point and the blob was not found, like in figure **??**(d). Here one can look for a common point of both lines, which is collinear to each of them. If such a point eas found, one can create two virtual lines from each line to the found point.

Applying the above scheme to all subsequent pairs in the ordered sequence yield a first approximate polygon describing the boundary of a blob. We further represent the line-points explicitly as virtual points, so that the elements of new sequence are either a point followed by a line or a line followed by a point.

Shortening Lines. Problem may still occur because some of the lines that are neighbored to the blob are longer than their boundary parts, like lines 8 and 17 in the figure **??**.

Therefore we have incorporated a correction step which tries to compensate these cases by computing the foot of perpendicular of the neighboring point (either a real or a virtual point). Note that we would not need this when we'd have an improved description of the exoskeleton.

The correction step is based on geometric relations and might not work in some situations, in our experience this observation holds for all purely geometric tests during grouping. An example: if one has computed the foot of perpendicular p_F of a point p onto a line with $[p_S, p_E]$, what is now the new line, $[p_S, p_F]$ or $[p_F, p_E]$? It is clear that this depends on the sequence of boundary features, i.e. the ordering of the neighbors.

We do not want to describe the used geometric algorithm in detail, since we think that it is much more reasonable to use the exoskeleton lines to determine the part of the line which belongs to the blob boundary. As mentioned before, this analysis of exoskeleton lines has not been done.

2.2.2 Iconic Approach

For some images the above approach might fail for blobs, where too many neighboring lines and points are missing. This can happen when the edge contrast between two blobs are too low to determine an edge, but high enough to distinguish two blobs; another reason could be that the blob sizes are too small and because of resolution problems one cannot retrieve edges or points. An example of such image data is the multichannel image in figure **??** on page **??**, an enlarged patch of the feature image and the exoskeleton is shown in figure **??**. The symbolic approach would not succeed on this kind of data.

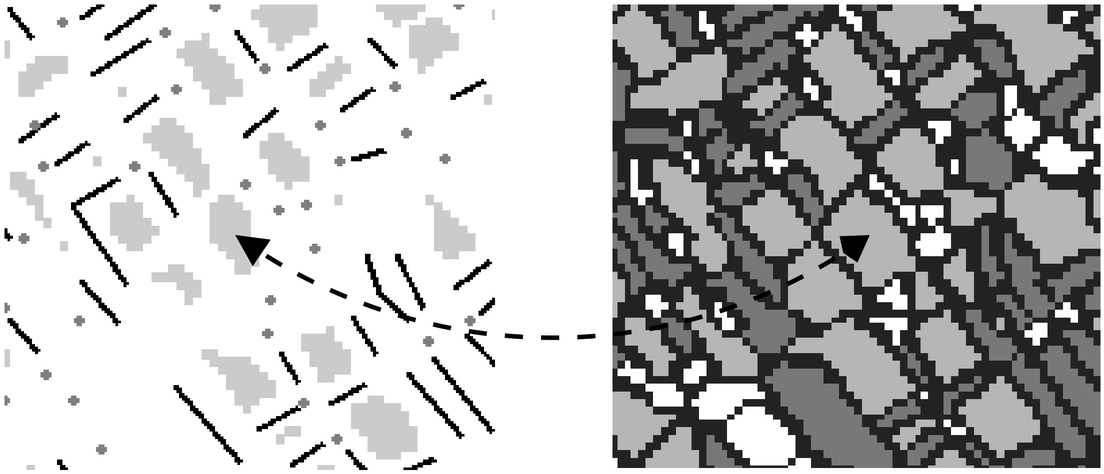


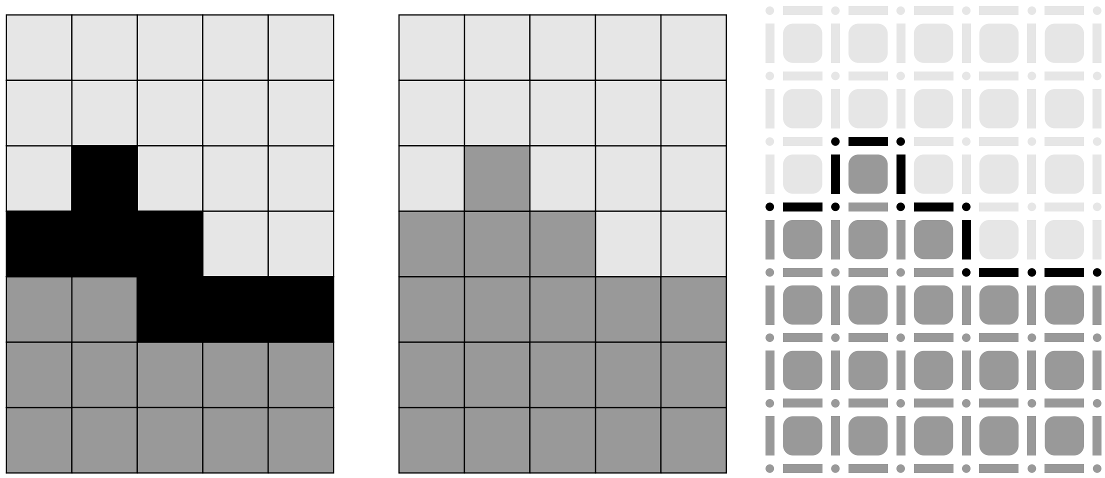
Figure 2.5: Problems of symbolic approach when too few features are extracted; on the left the extracted features are displayed, on the right the associated exoskeletons, where light grey regions correspond to exoregions of blobs.

Instead, we propose a purely iconical approach which could be described as a steered region growing scheme: starting from the extracted blobs we extend the borders to the exolines of its exoregion. This is reasonable since the exolines do have some geometric meaning: either an exoline also belong to an exoregion of an feature line, then we assume that the exoline is parallel to the extracted line (we neglect the uncertainty around the endpoints of lines); or they belong to a neighbored region, then the exoline gives us the best possible geometric description of the boundary of this blob.

One problem arises because of the nature of the exoskeleton image from FEX: the exoskeleton lines are black pixel in the image, see left picture in figure **??**. We want to to extend the exoregions to also cover these pixels, see figure **??**(a) and (b), so that the exoskeleton is actually represented as crack-edges between the two blobs. At this point we transform the image to hyper-raster format, see **?**. Then we can directly represent the exoskeleton as objects in hyper-raster format and use a connected components algorithm to identify and colorize each extended blob.

Remark: Coverage Measure of Points and Lines for a Blob-boundary. One could measure the coverage-percentage of points and lines for the boundary of blob by labeling their common exoskeleton line and divide the number of labeled line pixels with the total number of exoline pixels of the blob-exoregion.

Furthermore one can measure the overall distribution of features by projecting the closed boundary curve of the exoregion, i.e. the whole exoline, to a circle and compute the gravity point of the labeled pixels. If the gravity point is close to the center of the circle, the features are well distributed, see also **?**.



(a) (b) (c)

Figure 2.6:

2.3 Discussion

During this work we realized that we need an enhanced description of the topology in order to apply a reasonable grouping process.

2.3.1 Limitations of FEX

- For certain applications one notices insufficient resolution of FEX, see for example fig. ??. A solution would be to use the hyper-raster already during at the segmentation of points, lines and blobs.
- The exoskeleton is not analyzed to be stored in a symbolic description. Furthermore its iconical representation is not available as a hyper-raster image and therefore lacks topological consistence.
- The boundary of images is not modeled. This can cause problems in the description of an image: the relationship of a feature and an image boundary is not explicitly stored.

2.3.2 Possible Improvements of FEX

Besides maintaining and debugging the current implementation, there are some theoretical issues, that should be looked at when a re-implementation of the system is considered.

- Using a hyper-raster, at least for the extraction of the feature topology.

- Analysis of the relation of exoskeleton lines to the features of the exoregions
  - line endpoints onto exoskeleton lines
  - given two features, the projection of the common exoskeleton line onto both features could give useful information
- Analysis of the feature position inside its exoregion.
- Influence of the gradient image on the exoskeleton line, see **?**.

Chapter 3

Classification

In a classification step it concerns now to assign the regions, segmented before, special land use units. In addition each region is checked regarding its similarity with all possible land use units (classes) and assigned according to one class or rejected however, if no allocation is possible.

3.1 Classification techniques

Different techniques exist, with which a classification can be implemented ?. Such a technique is called *classificator*, a algorithm, which is use this technique is called *classification*. A subset of objects, which is interpreted connecting for numeric, heuristic or subjective reasons is called *class*. The criteria, by which special characteristics of objects are more specified and on the basis those one comes to a decision of allocation, are the *features*. One differentiates between *linear*, *logical* and *statistical* classifiers.

Linear classification One assumes objects are certain by M-dimensional feature tuple. This tuple are so called feature vectors. The M-dimensional Euclidean space E^M, that means the space of all M-dimensional feature vectors, is called *feature space*. To obtain a classifier, the objective is to divide the feature space into a number of disjoint regions C_k with respect to the number of existing classes K. The simplest case is the diviation of the E^2 in two regions by a straight line, which is or of the Euclidean space by a hyper level. But uses one a linear function, which is called *discriminance function*.

$$D(x \mid a) = D(x|a_0, ..., a_M) = -a_0 + \sum_{m=1}^{M} a_m \, x_m$$

The diviation can be achieved by the following decision:

$$\begin{aligned} x \in C_1 \text{ if } D(x \mid a) &< 0 \\ x \in C_2 \text{ if } D(x \mid a) &> 0 \end{aligned}$$

An example of such type of classifier is the so-called **M**_inimum_-**D**_istance-classifier_, short MD-classifier. Klassifikator (s.o. fig.**??**). The idea of a MD-classifier is to charakterize a class by a representative. So a object belongs to the class, to which it posesses the smallest distance d . For the definition of the distance a euclidian (f.e. the euclidian metric) is given:

$$d(x, S^{(k)}) \;=\; \sum_{m=1}^{M}(x_m - s_m^{(k)})^2 \;=\; \sum_{m=1}^{M} x^2 m - 2\sum_{m=1}^{M} x_m s_m^{(k)} + \sum_{m=1}^{M}(s_m^{(k)})^2$$

Because that is an extreme value problem, the right term is without influence and so $d(k)$ is:

$$d_k(x) = \sum_{m=1}^{M}(s_m^{(k)})^2 - 2\sum_{m=1}^{M} x_m s_m^{(k)}$$

Therefore the feature space K is devided in disjoint regions R_k by

$$R_k \;=\; \{x \mid d_k(x) < d_l(x), \;\; l \neq k\}$$

In order to reduce the probability of an incorrect correspondence, also a rejection class is introduced. So a object belongs to the class, to which it posesses the smallest distance d again, but the distance d may not exceed a characteristic value, otherwise the object will be assign to the rejection class. Such a non-linear classifier one calls **M**_inimum_-**D**_istance-classifier_ with rejection (e.g. Abb.: **??**).

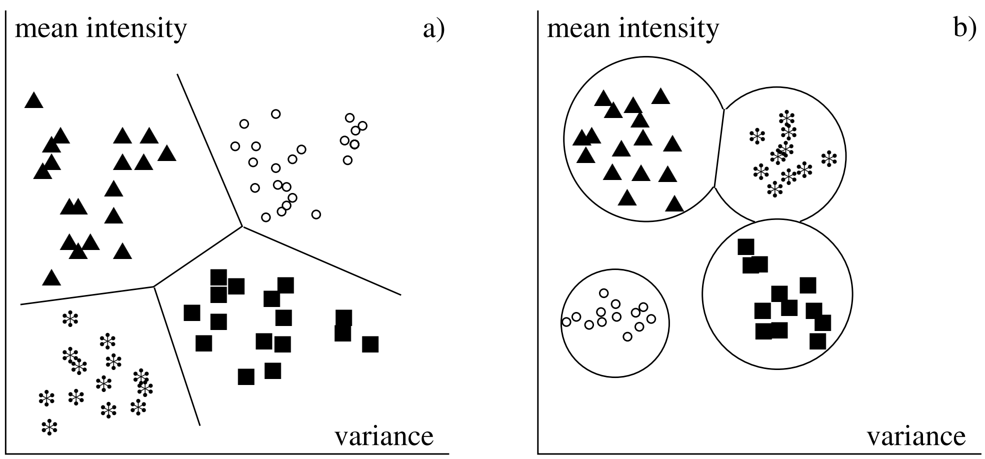


Figure 3.1: a) MD-classifier b) MD-classifier with rejection class

To use only one representative for one class often leads to an uncorrect deviation of the feature space. So it is better to use several representatives. For that case first one determines the smallest distances of an unknown object x to all representatives $s^{k,i}$ of one class C_k. The next step is then the determination of a minimum of all of these distances.

$$d_{min}^{(k)}(x) \;=\; \min_{i\in(1,\ldots,n_k)}(d(x, s(k,i)))$$

Such type of classifier is called **N**_earest_-**N**_eighbor-classifier_ (e.g. Abb.: **??**).

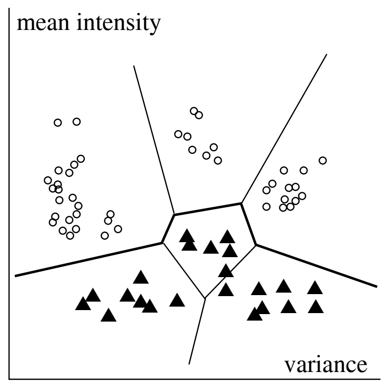


Figure 3.2: kNN classifier

Logical classification There is an other possibility to devide the feature space in characteristical regions. Each region can be described by an parallel epiped. For the m-dimensional feature space these description leads to so called *hyperquader* for each class (e.g. Abb.: **??**). Thereby a class C_k is decribed by the extremal points
$p^{(k)} = (p_1^{(k)}, p_2^{(k)}, \ldots, p_M^{(k)})$ and $q^{(k)} = (q_1^{(k)}, q_2^{(k)}, \ldots, q_M^{(k)})$.
It applies $x = (x_1, x_2, \ldots, x_M) \in C_k$ exactly if $\forall_{m=1}^{M} \, [p_m^{(k)} \leq x_m \leq q_m^{(k)}]$

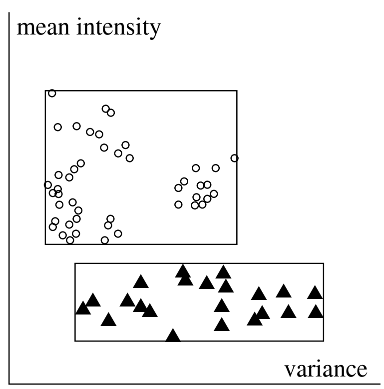


Figure 3.3: parallel epiped classifier

If an object is assign to more than one class on that way, there exist three possible decision rules:

- The object belongs to no class.
- The object belongs to one of the possible classes only.
- The object belongs to more than one class.

Statistical classification In the practice it is possible that one object belongs to the class C_k but it is assigned to the class C_l, because with respect to its geometry it is situated in the region R_l. For that reason the classes C_k are not exactly represented by the regions R_k. Until the typical objects of a class C_k are situated in the interior of a region with a high certainty, untypical objects are situated nearly the border of these region. Thus the possibility of an error classification is higher. To solve this problem statistical considerations seems to be useful. Therefore one assumes the classes $C_1, \ldots, C_k$ as strongly different from the regions $R_1, \ldots, R_k$ (even the number of classes k and the number of regions l could be different). For uncertain decisions a rejection class can use again. To use statistical classification techniques one assumes the classes C_k as integer value k and the propability p_k of that event "the object x belongs to the class C_k" can be decribed by the random value Y with $p_k = P(Y = k)$. p_k is called *A-priori-class-propability.* The feature space is described by the random vector X. So is a $M + 1$-dimensional random vector given by $Z = (Y, X)$. The propability that an object x belongs to the class k and that it is situated in the region l is then given by

$$P(Y = k, X \in R_l) \;=\; P(Y = k) \int\limits_{R_l} f(x \mid Y = k) dx.$$

The function $f(x \mid Y = k)$ is called *causes density* of the class k. The propability $p_k(x)$ that an object x is situated in any arbitrary region is given by

$$p_k(x) \;=\; p(Y = k \mid x) \;=\; \frac{p(Y = k)\; f(x \mid Y = k)}{\sum\limits_{k'=1}^{K} p(Y = k')\; f(x \mid Y = k')}.$$

The propability $p_k(x)$ is called *A-posteriori-class-propability.* A typical representative of such type of classification is the *bayes classifier*(e.g. Abb.: **??**).

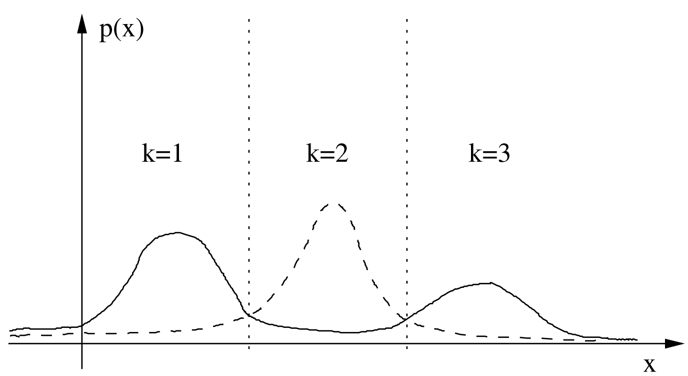


Figure 3.4: bayes classifier

3.2 Algorithm

In this section the algorithm for classification land-use units is described. Results for classification are shown and discussed.

3.2.1 Input

As input of the classification algorithm the symbolic description of FEX is used. These descriptions contains the extracted blobs (as symbols for land-use-units), which are characterized by their

- number
- mean intensity
- mean variance of the grey levels.

The feature space of the objects, which have to be clasified, is given by these characteristical features.

3.2.2 Teach of the classifier

For teaching the classifier an unsuperwised process is used. Therefore the histogramms of the features (mean intensity, mean variance of the grey levels) of the whole feature space are computed. The result of analyzing these histogramms are the borders of possible classes. Within these borders the features characterizing the special classes are determined. Therefore the following equations are used:
for the mean variance σ_k of a class k

$$\sigma_k = \frac{1}{n} \sum \sigma_i$$

for the mean intensity g_k of a class k

$$g_k = \frac{1}{n} \sum g_i$$

In these equations the parameter n is the number of blobs within the class borders, determined before. The number of classes K and the characteristical smallest distance d of the classifier are given by analyzing the feature histogramms too.

3.2.3 Classification with kNN-classifier

Because of simplicity of realization and good results for the classification the *kNN-classifier* is used (e.g. sec. **??**). The first step of classification is the determination of the smallest distances of an object x (blob) to all features (representatives) of one class C_k.

$$g_i = d(g_i, g_k) \forall k \in [1, K]$$
$$\sigma_i = d(\sigma_i, \sigma_k) \forall k \in [1, K]$$

letter/color	interpretation
A/dark blue	wheat
B/red	maize
C/brown	rye
D/cyan	potato
E/violett	wood
F/yellow	barley
G/lightblue	pure ground

Table 3.1: Possible assignment for classification

The next step then is the determination of a minimum of all of these distances by

$$d_{min} = \min_k d(g_i, \sigma_i; g_k, \sigma_k) \; \forall k \in [1, K]$$

Then, an object belongs to the class C_k, for which this smallest distance d_{min} was calculated.

3.2.4 Output

The output contains the ID number of each object and it assigned class number in form of a list. The result can be represented graphically by coloring the aereas (blobs) according to its classification. A possible result is indicated in tabel **??** .

3.3 Discussion

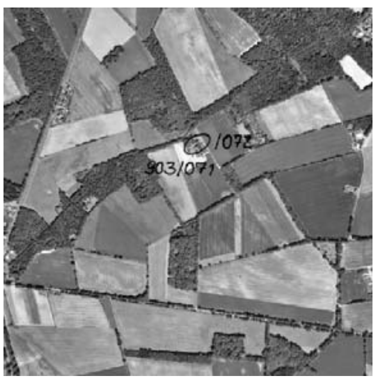
The section **??** demonstrates results for the classification of land-use-units for three different arial images. For examples seven classes were differentiated. The classes could be interpreted as follows:

For each example the initialization files are given here. They contain the parameter of the unsuperwised teached classifers. The files `acker1.knn.ini` and `acker2.knn.ini` are used for classification the aerial images I and II (e.g. **??**, **??**), the file `flevo.knn.ini` was used for classification a multichannel image (e.g. **??**).
If one compares the original image data with the classification result, it can be fixed-placed that most assignments took place correctly. That means, that diffenrent agricultural aeras interpreted also different. This statement is given without statistical investigations of the results. Classification problems occurred with wooded aereas, because of the texture of such aereas. This is to be seen clearly in the example **??**, where the classifiaction result is partly wrong or unpossible. The problem of other uncorrecly assignments could solved by further training the classifier.

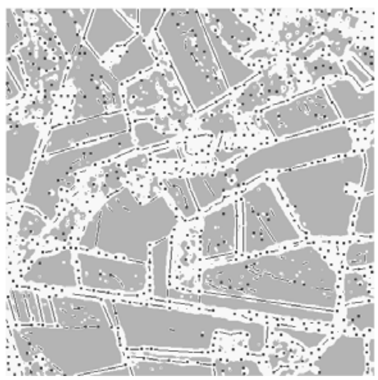
```
 acker1.knn.ini
% ******************************************************************
%dim_feat_vec 2 (dimension of feature vector)
%num_of_classes 7 (number of classes)
%return_value 3000 (value for a distance)
%class_boundaries_(g(k),S(k)) (characterizing features for class k)
%9.0 7.0
%55.0 0.0
%75 0.0
%90 0.0
%120 35.5
%160 0.0
%173.16 0.0
%194 0.0
% ******************************************************************
 acker2.knn.ini
% ******************************************************************
%dim_feat_vec 2 (dimension of feature vector)
%num_of_classes 7 (number of classes)
%return_value 3000(value for a distance)
%class_boundaries_(g(k),S(k)) (characterizing features for class k)
%100 1.0
%120 0.0
%135.0 0.0
%156.8 0.0
%173.16 0.0
%194 0.0
%200 0.0
% ******************************************************************
 flevo.knn.ini
% ******************************************************************
%dim_feat_vec 2 (dimension of feature vector)
%num_of_classes 7 (number of classes)
%return_value 100(value for a distance)
%class_boundaries_(g(k),S(k)) (characterizing features for class k)
%0 0.19
%21.11 0.0
%24.11 0.0
%30.5 0.0
%35.8 0.0
%45.36 0.0
%55.2 0.0
% ******************************************************************
```

Chapter 4

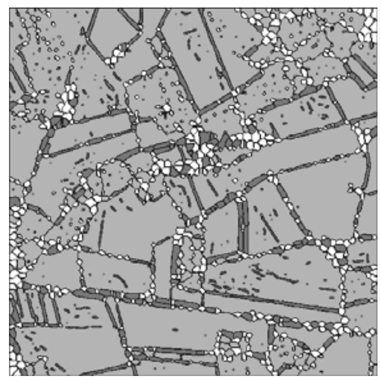
Results



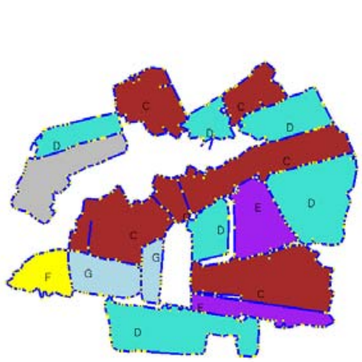
(a) Original



(b) Features



(c) Exoskeleton

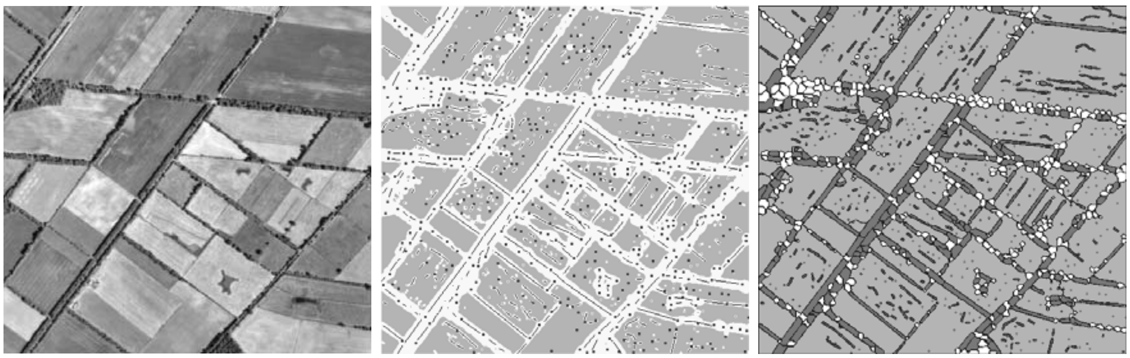


(d) Blob-classes

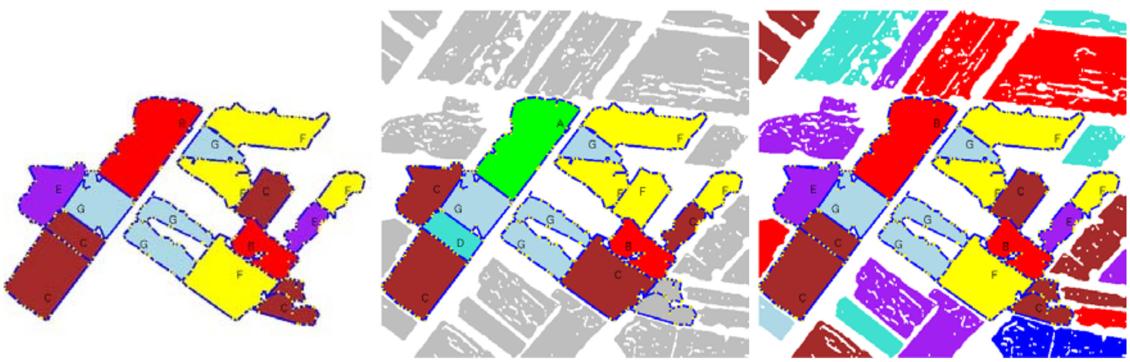
(e) Blob-classes with non-classified boundary blobs

(f) Blob-classes with classified boundary blobs

Figure 4.1: Segmentation and classification of an aerial image I



(a) Original (b) Features (c) Exoskeleton

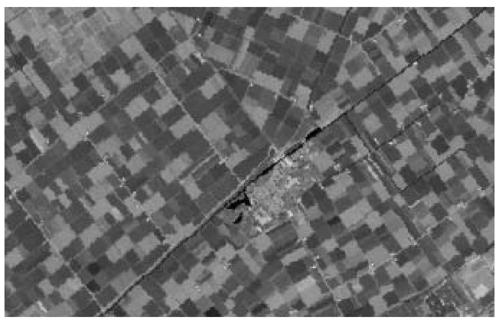


(d) Blob-classes (e) Blob-classes with non-classified boundary blobs (f) Blob-classes with classified boundary blobs

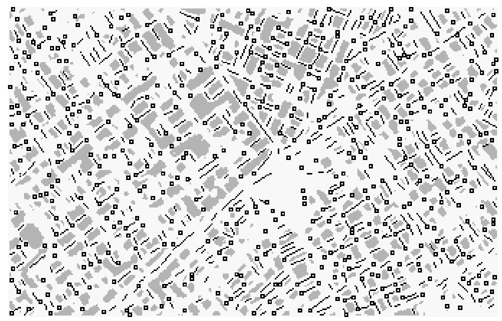
Figure 4.2: Segmentation and classification of an aerial image II

(a) Original, first channel (normalized)

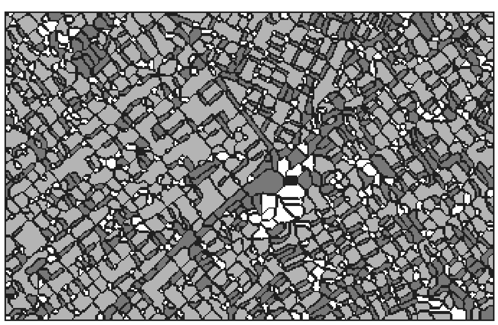
(b) Original, second channel (normalized)



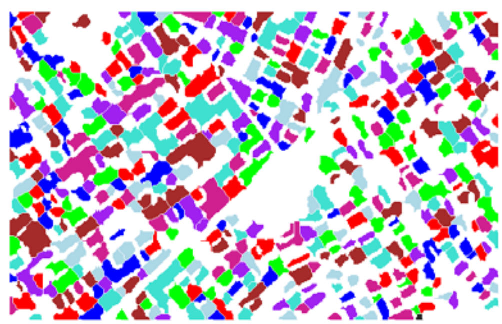
(c) Original, third channel (normalized)



(d) Features



(e) Exoskeleton



(f) Blob-classes

Figure 4.3: Segmentation and classification of a multichannel image (courtesy from ...)

References

FÖRSTNER, W. (1994): A Framework for Low Level Feature Extraction. In: EKLUNDH, J. O. (Ed.), *Computer Vision - ECCV 94, Vol. II*, Band 802 der Reihe LNCS, pages 383–394. Springer, 1994.

FÖRSTNER, W. (September 1999): Tensor-based Distance Transform. Note, September 1999.

FUCHS, C.; LANG, F.; FÖRSTNER, W. (1994): On the Noise and Scale Behaviour of Relational Descriptions. In: EBNER, HEIPKE, EDER (Ed.), *Int. Arch. f. Photogr. and Remote Sensing, Vol. 30, 3/2*, Band 3, pages 257–267, 1994.

FUCHS, C. (1998): *Extraktion polymorpher Bildstrukturen und ihre topologische und geometrische Gruppierung*. PhD thesis, Universität Bonn, 1998.

MAROUANI, S.; HUERTAS, A.; MEDIONI, G. (1995): Model-Based Aircraft Recognition in Perspective Aerial Imagery. In: *Symposium on Computer Vision '95*, pages 371–376, 1995.

PAN, H.; FÖRSTNER, W. (1994): Segmentation of Remotely Sensed Images by MDL-principled Polygon Map Grammar. In: *Int. Arch. f. Photogr. and Remote Sensing*, 1994.

VOSS, K.; SÜSSE, H. (1991): *Praktische Bildverarbeitung*. Carl Hanser Verlag München, Wien, 1991.

WINTER, STEPHAN (1995): Topological Relations between Discrete Regions. In: EGENHOFER, MAX J.; HERRING, JOHN R. (Eds.), *Advances in Spatial Databases*, pages 310–327. Springer, LNCS 951, 1995.