FUSION OF DIGITAL TERRAIN MODELS AND TEXTURE FOR OBJECT EXTRACTION

W. Eckstein, C. Steger
FG BV, Informatik IX
Technische Universität München
80290 München
{eckstein,stegerc}@informatik.tu-muenchen.de

ABSTRACT

The paper introduces the extraction of objects that are higher than their surroundings, like houses, trees, or bridges, by combining the results of a segmentation of a DTM and a texture analysis of the gray image. We propose a new algorithm (called dual rank) to extract the ground in the DTM, which can be seen as an extension of the gray opening. The extraction of trees in the gray image is done by a texture transformation, which is a fast and stable method, for the distinction between houses/roads and trees. The combination of both results is done region-oriented, i.e., every high object is analysed if it is a house, a tree, or a combination of both and thus separated and classified. In addition shadows can be extracted using the DTM to support the interpretation process.

1 INTRODUCTION

The overall aim of our project (Steger$^+$,1995) is the interpretation of aerial images with the initial task of extracting objects like roads, houses and trees. The problem is the complex structure of the images, containing many different objects each varying in shape, color, and texture. This leads to a large search space for the matching process. To reduce the search space different sources of information have to be incorporated. This can be done by extending the model and by fusion of different input sources like color images, cartographic data, or laser scannings. This paper focuses on the fusion of the digital terrain model (DTM) with the aerial image. The DTM provides information about special classes of objects. One class is characterized by the relative height with respect to the surrounding ground. Typical representatives are houses, trees, trucks, and bridges. Another area is the shadow of high objects which cause additional edges in the gray image and thus complicate the interpretation process. On the other hand, the aerial image contains information about shape, intensity, and texture of objects. The combination of both can be used to extract high areas and separate these into different classes like houses, trees, or shadows. The distinction of objects in these classes allows a more stable and efficient matching in a subsequent model based interpretation process.

Different approaches to the extraction of objects in DTM have been proposed. The height information is either used directly (Haala, 1995) or via a stereo reconstruction process (Dang$^+$,1994). The advantage of the second approach is that the gray information of both images is used. However this can also be obtained by the first approach by calculating two orthophotos from the DTM. The images that have been used so far are very simple (Fritsch$^+$,1994): flat ground, simple houses, and little or no trees. In addition, the quality of the images is very good (high resolution and no errors in the DTM). Therefore, simple methods for the extraction of high objects have been applied. Two popular methods are the extraction by means of isolines, i.e., using shape and height of isolated areas, and the gray opening of the DTM, calculating an image representing the true ground as a reference for the comparison.

Our approach deals with realistic and thus more complex scenes compared to (Fritsch$^+$,1994): The DTM has a low resolution of 2 m and it contains errors. The environment is hilly and consists of a mixture of houses and trees standing close to each other. Due to these additional difficulties of the input data, more stable algorithms have to be developed.

The proposed extraction of the objects can be divided into the following steps:

Image material Using aerial images, an automatic DTM is calculated. An orthophoto based on this DTM is used. This paper will not deal with the calculation of DTMs or orthophotos.

Extraction of the ground Three different approaches are dicussed which are appropriate for hilly areas with objects of different size: Gray opening, lowpass filtering and a new operator called dual rank.

Extraction of relatively high objects This is a simple comparison between the DTM and the ground DTM.

Segmentation of trees The extraction of trees in the gray image is done by a texture transformation (Laws,1980) which can be calculated very efficiently and is thus an alternative to the texture analysis using gabor filters (Shao$^+$,1994).

Separation and classification of objects The combination of both segmentations is done by subtracting areas belonging to trees from high objects. This eliminates trees standing next to houses.

Extraction of shadows Knowing the position of the sun relative to the DTM, the extraction is a simulation of illumination with subsequent noise elimination.

2 HIGH OBJECTS

We start with an automatically generated DTM and a corresponding orthophoto. The DTM that is used has a resolution of 2 m. It is calculated with standard parameters in order to have realistic data for the automatic interpretation of different aerial photos. For the extraction of the ground, we assume that the relevant objects have a maximum length d between 20 m and 30 m. This size is large enough to detect most of the interesting objects (except dense forest and huge buildings) and allows a stable detection of the ground even in a hilly environment. In figure 1 we see two images with their corresponding DTMs. Both DTMs have complex structures, containing hills, ridges and steep slopes. In addition, some errors can be seen in the left example (peaks next to the road) and most of the houses are not as significant as expected.

The most popular operator for the extraction of the ground is the gray opening. It is a minimum filter followed by a maximum filter. A square of size $d \times d$ is often used as filter mask because rectangular masks can be separated to reduce the runtime from $O(d^2)$ down to $O(d)$ per pixel. The disadvantage of a square mask is that it is not rotationally invariant. Therefore, we suggest to use circular masks with diameter d.

If the DTM is noisy the use of an extremal value filter is unstable because the result can be significantly influenced by a single faulty pixel. Therefore, we propose a new filter which does not use the extremal-values but a rank value instead. To introduce this operator, we need some definitions (see also Haralick,1992):

Definition 2.1 *Let $\mathcal{N}^p$ be the gray values of an arbitrary neighborhood of pixel p and $n = |\mathcal{N}^p|$ the number of pixels in that neighborhood. The gray values $\mathcal{N}^p_i \in \mathcal{N}^p, i \in \{1 \dots n\}$ are sorted by a function s in such a way that*

$$\mathcal{N}^p_{s(1)} \leq \dots \leq \mathcal{N}^p_{s(n)}$$

The rank operator R with rank value $r \in \{1 \dots n\}$ is now defined by

$$R(p, r) \mapsto \mathcal{N}^p_{s^{-1}(r)}$$

The operator is quite simple: we sort all gray values in the neighborhood with a sort function (called s here), and select the one with a given rank value. If we choose $r = 1$ we get the minimum filter. In the case $r = n/2$ the filter is the well known median filter. If $r = n$ we get a maximum filter. If we choose r close to 1 or n we get a kind of minimum or maximum filter which is more stable with respect to noise. The neighborhood $\mathcal{N}$ can be of any shape. For this application we prefer circular masks, to make the operator is rotational invariant. Based on the rank operator we can now define the *dual rank*:

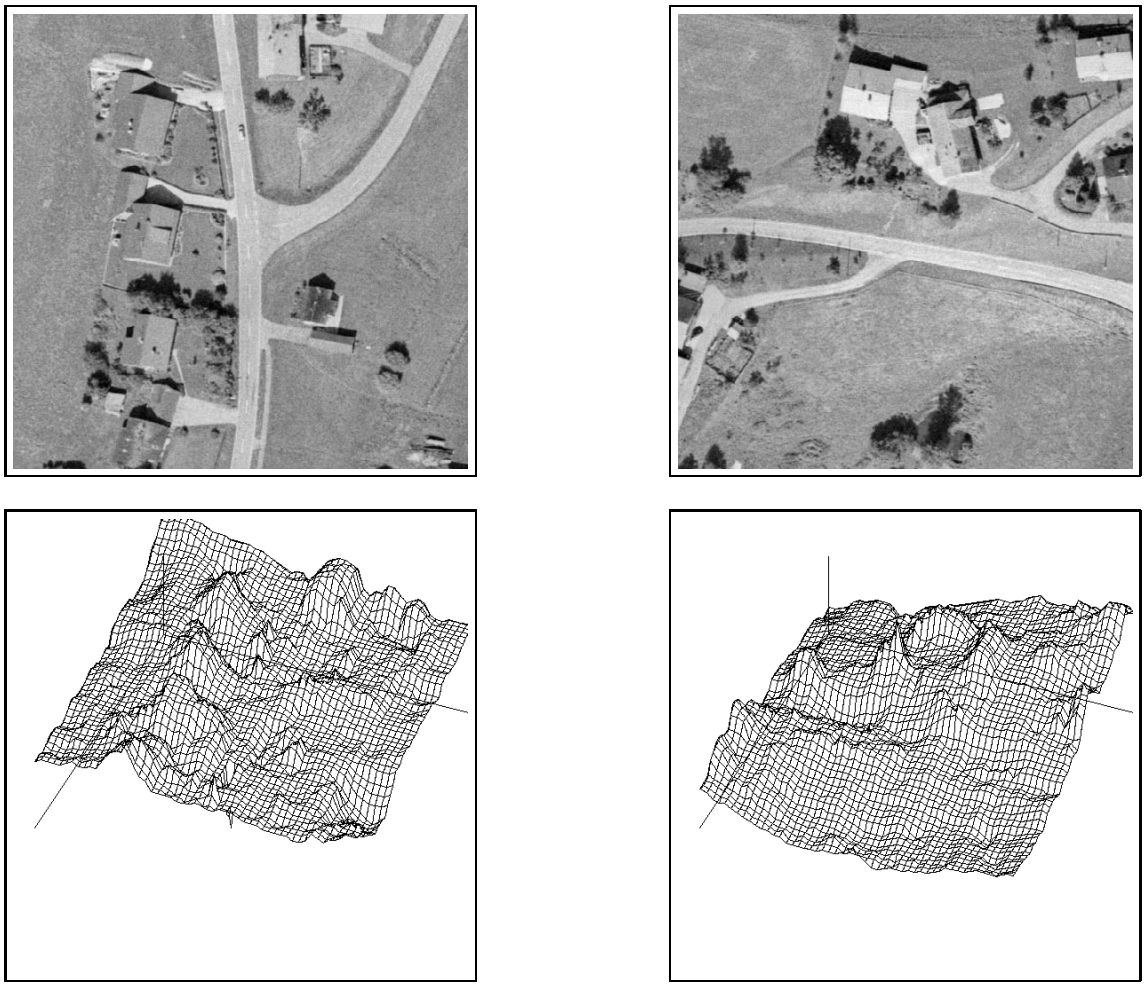

Figure 1. Two examples images with a resolution of 25 cm and the corresponding DTMs with a resolution of 2 m.

Definition 2.2 *Let p be a pixel, R be a rank operator for a given neighborhood, and r the rank value. The dual rank operator DR is defined by*

$$DR(p, r) \mapsto R(p, r) \circ R(p, n - r) \tag{1}$$

The dual rank consists of two successive rank operators. The first rank operator is applied with the given rank value while the second one uses the "complementary" value. The rank value 1 results in a gray opening, and the value n corresponds to a gray closing. In the case of $n/2$ we get two successive median filters. The rank value thus controls the selection of the operators.

The operator is defined in a very general manner and therefore flexible. This generality is achieved at the loss of efficiency. If we use a straight forward implementation we get the runtime complexity of

$$O(n \log(n)) \tag{2}$$

per pixel, where n is the number of pixels in the neighborhood. For our application, n is more than 1000, i.e., the algorithm would be fairly slow. If the accuracy of a byte image is sufficient to code (possibly part of) the DTM, the following algorithm can be used to calculate the rank and thus the dual rank very efficiently.

The main ideas for the implementation are:

1. A table with 256 entries is used containing the count of each gray value in the current neighborhood, i.e., the histogram of the gray values in $\mathcal{N}$. This table is a work-around for the sorting function.
2. The neighborhood $\mathcal{N}$ is coded using horizontal run lengths. With this type of coding the update of the table is simple because only the start and end points of the runs have to be considered. Going from one pixel to the next, all entries have to be decremented with the gray values of the start points and incremented with the gray values at the end points at the next position.
3. The search through the complete table for the rank gray value can be speeded up if the last rank value and the number of new gray values below and above the last rank value are used: we seek for the new rank value by starting from the old one in the direction of new entries in the table.

If we analyze the mean runtime complexity of this implementation we obtain

$$O(\sqrt{n}) \tag{3}$$

The square root results from the number of runs needed to code the neighborhood, which is convex and compact. Thus we have $2\sqrt{n}$ updates in the table. The search for the rank value is nearly constant, because the change of the rank value is equivalent to the gradient in the resulting image, which is very homogeneous. The worst case for this constant is 255 (image with horizontal black and white lines of width 1).

Runtime tests on a HP 712/60 with an image of size 512×512 are given in table 1. The times given are for the calculation of a dual rank, i.e., for two successive rank operations. The results show that the runtime is linear with the radius which is order $O(\sqrt{n})$, as stated above. Therefore we get the same complexity as the separated gray opening because $\sqrt{n} = d$.

Radius	n	Runtime (sec)
5	78	3.9
10	314	7.3
15	706	12.1
20	1256	14.6
30	2827	22
50	7854	47

Table 1. Runtime of dual rank with circular masks.

If the resolution of a byte image is not sufficient one normally would use float values to encode the height without a scaling. In this case the table used as an implementation for the sorting cannot be used. Therefore we have to insert every gray value at the ends of the runs and delete the gray values at the begins of the runs. Because insertion and deletion are of complexity $O(log(\sqrt{n}))$ the whole algorithm is of complexity

$$O(\sqrt{n}\log(\sqrt{n})) \tag{4}$$

The dual rank operator defined above can now be applied to for the extraction of the ground DTM. If we know the percentage e of erroneous pixels (e < 50) in the DTM we can simply choose the appropriate value for the rank r:

$$r = n\,\frac{e}{2 \cdot 100} \tag{5}$$

Using this rank value, we get an operator which is similar to the gray opening but more stable with respect to noise. Figure 2 shows the result of this operator applied to the DTMs in figure 1. We call the result *ground DTM*. Both examples show the steep slopes of the terrain in the image.

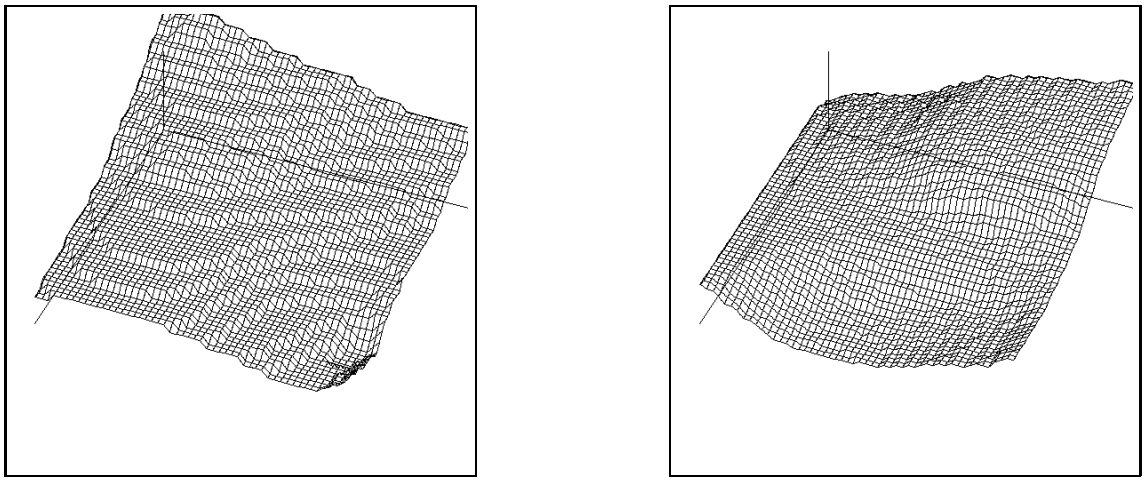

Figure 2. Ground DTM calculated by the dual rank.

Using this ground DTM, the extraction of high objects is very simple: It is just a comparison of the two DTMs. This can be calculated by a subtraction of both images, which is called *top-hat* in the case of gray opening. We call the result *normalized DTM*. Figure 3 shows the result of this operation. The left two images allow a comparison of the dual rank (left) and the gray opening (center). In the left image the ground is smoother with less "noise". The error in the DTM (next to the road in the front) results in an error (peak) using the gray opening. This peak is eliminated using the dual rank. In addition the "walls" of the houses are steeper (see the buildung at the back) compared with the gray opening.

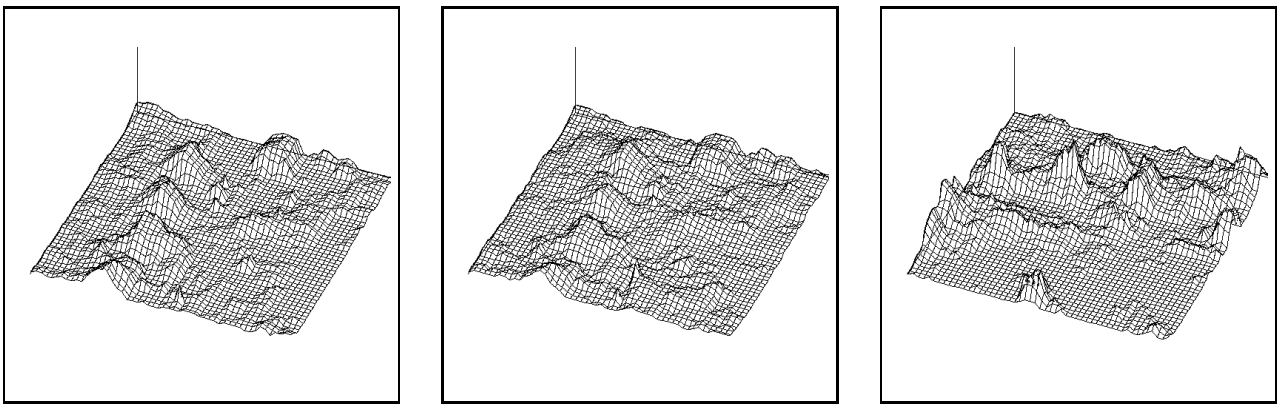

Figure 3. Normalized DTM of examples in figure 1. The outer images are calculated with the dual rank. The image in the middle is calculated with the gray opening.

The next step, the extraction of high objects, is just a threshold in the normalized DTM. Figure 4 shows the results of three different algorithms: The left image displays the results using the dual rank operation. The image in the middle shows the result of the gray opening. We see some errors in the front and at the junction of the road. The detection of the road is a bit surprising. But the road at the junction is actually higher than the surrounding meadows. Another problem is the house at the right, which is too small to be significant in the resolution of the DTM.

The right picture of figure 4 shows the results of the *dynamic threshold*. This is a very simple but highly efficient algorithm (Eckstein^{+},1991). The first step is to calculate a lowpass image. In this case we used a mean filter. The size

Figure 4. High objects calculated by dual rank, gray opening, and dynamic threshold (left to right).

of the filter mask is twice the size of the objects. The second step is a comparison between the lowpass image and the original image: all pixels of the original image which are "higher" than the pixels in the lowpass image are chosen. To make this procedure more stable, an offset is added to the lowpass image. The lowpass filtering in combination with the comparison can be interpreted as a threshold in a highpass image. The result is very similar to the other algorithms and can be used as a efficient alternative. Problems will arise if the houses/trees are standing very close to each other, e.g., if there are only some gaps between the trees in a forest. In this case the objects can no longer be separated or even detected.

3 TEXTURED OBJECTS

Once high objects have been found, they have to be classified. If they do not stand very close to each other a simple classification using texture information can be applied. If, for example, trees are standing next to a house (see the large objects in the lower half of figure 4) these objects have to be segmented. The proposed solution is a pixel classification in the gray image in order to find objects like trees. This is done independently from the extraction of high objects. The results are combined in a subsequent step.

There exist a lot of possible solutions to detect trees in images. Our approach uses linear filters which extract different frequency bands. One way to do this is by using gabor filters (Shao$^+$,1994). Because the distinction between the textures of roofs and trees is very simple we use the texture filter proposed by Laws (Laws,1980). This approach uses 5 convolution vectors l, e, s, r, and w.

$$
\begin{array}{lcr}
l & = & (\ 1 \quad 4 \quad 6 \quad 4 \quad 1) \\
e & = & (-1 \quad -2 \quad 0 \quad 2 \quad 1) \\
s & = & (-1 \quad 0 \quad 2 \quad 0 \quad -1) \\
r & = & (\ 1 \quad -4 \quad 6 \quad -4 \quad 1) \\
w & = & (-1 \quad 2 \quad 0 \quad -2 \quad 1)
\end{array}
$$

They can be combined (convolved) to 25 convolution mask of size 5×5, which are consequently separated. Each mask implements a different bandpass filter; we only use the symmetric versions, i.e., we have 5 texture filters. Three of these are shown in figure 5. At the left is filter ee, the next is filter ss, and the last is rr. Filter ee and ss are bandpass filters, rr is a highpass filter. These filters are symmetric but not rotational invariant. This disadvantage has to be accepted to improve the runtime (separated masks).

The pixel classification consists of the following steps:

1. Convolution of the image with the filter masks. To be invariant with respect to intensity, the filter ll is omitted (it is similar to a gauss filter).

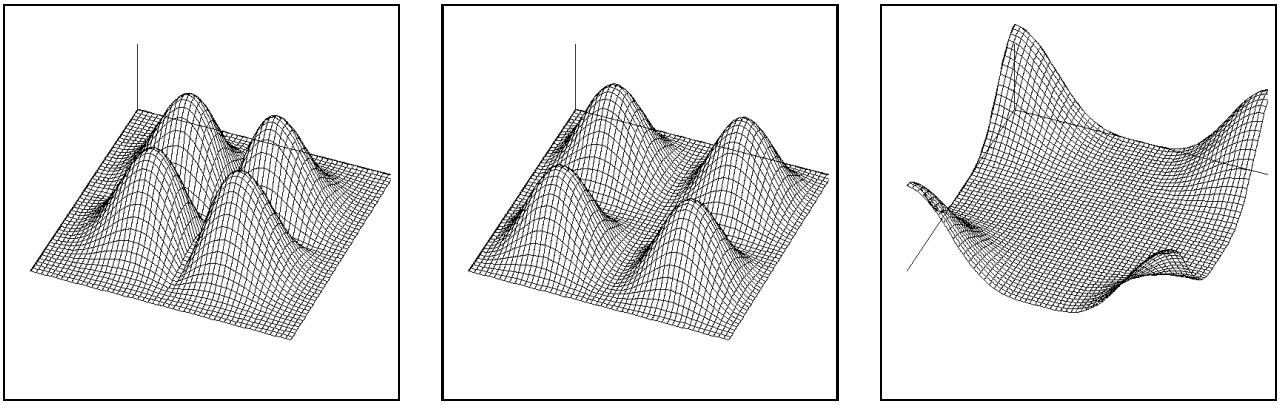

Figure 5. Three symmetric bandpass filters proposed by Laws.

2. In every filtered image the *texture energy* is calculated using a lowpass filter. Best results can be achieved with a median filter with a circular mask of diameter 10 m. Alternatives would be the mean and the gauss filter.
3. The classifier is trained using different sample regions. A classifier was used which approximates the 4-dimensional feature space by hyper spheres. Each cluster in the feature space is approximated by a set of spheres with a maximum diameter.
4. Every pixel is now classified using the 4 texture energy images.
5. To suppress noise, small regions were eliminated (minimum size $2\,m^2$) and a closing operation with circular mask (radius = 0.8m) is applied.

The result of this process can be seen in figure 6. The resulting classification is quite good, but there are some problems with pseudo-textures caused by shadows, roads, and houses standing close to each other. How to deal with these regions is discussed below.

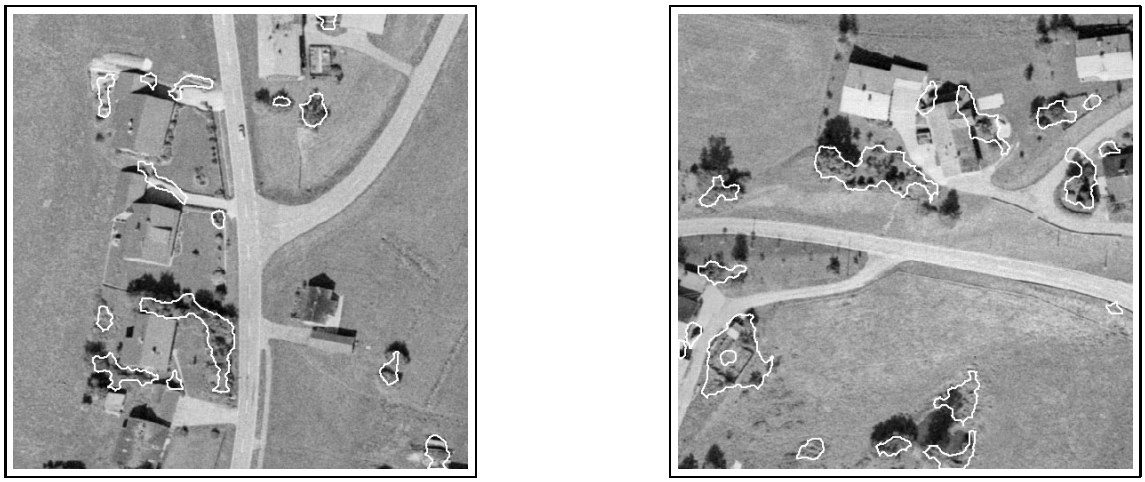

Figure 6. Classified pixels (class tree) after noise elimination.

4 COMBINATION OF HIGH AND TEXTURED OBJECTS

Using the results of section 2 and 3, it seems as if everything is done: we have high objects and areas with the texture of the trees. But the classification is actually more complicated:

- Objects of different classes often stand very close together (i.e., houses and trees). In this case the feature extraction has to be used not only for classification but also for a refinement of the segmentation. If we look at the object in figure 4 we see, for example, a house and trees next to it. These objects have to be partitioned into separate areas. This is done by calculation the difference between every high object and the pixels classified as trees.
- If the region is very hilly the extraction of the ground may fail. Therefore, a post classification has to be applied to this class of objects (mainly meadows and roads). In figure 7 at the right we see part of a road wrongly classified as a high object and thus as a house. This happens because the road lies on a ridge (see figure 3) and is therefore higher than the surrounding meadows. This problem can be solved by using a module for the extraction of roads (Eckstein^{+},1991) which allows a stable classification of this type of objects. The distinction between meadows and roofs using texture in an image with a resolution of 25 cm is impossible. In many cases color would be helpful.

After the refinement of the segmentation some noise cleaning is done: Every area smaller than 9 m^2 is eliminated and a closing operation with a circle (radius = 0.6m) is finally applied. The results of this process can be seen in figure 7. The outer contour of the detected objects is rather poor. Especially if we look at the right picture, the shape of

Figure 7. Final regions of interest for houses.

the houses does not fit quite well. This is mainly because the resolution of the DTM is low. An improvement of the resolution and an optimization of the parameters of the matching tool would yield better results. But this is not realistic if we want to process a large amount of images. The time for data aquisition is not acceptable. We propose to use the areas found as seed point for a refining segmentation or after a dilation as regions of interest to reduce the search space during matching.

5 EXTRACTION OF SHADOWS

Shadows cause a lot of problems during the interpretation of images because they change the gray values of objects drastically and add egdes or texture-like structures. In the image in figure 8, for example, the road is divided

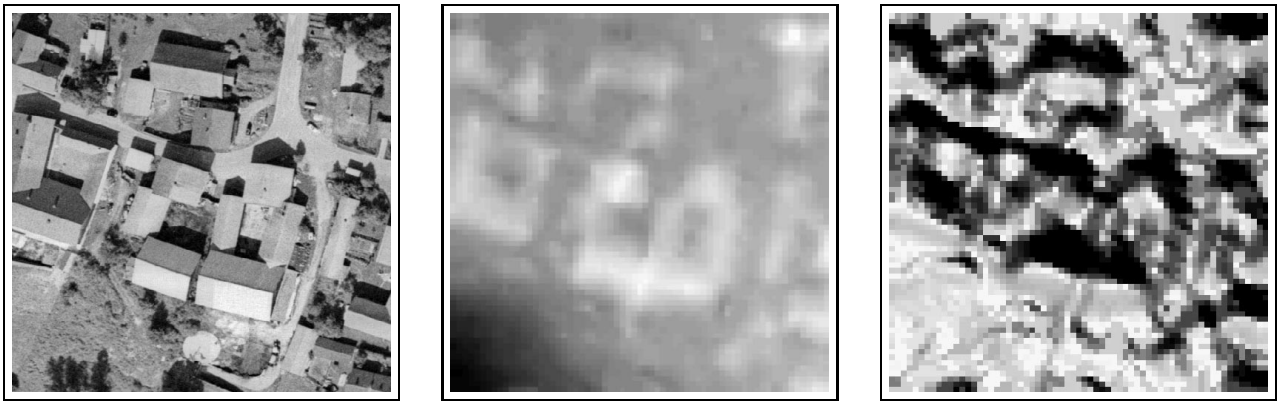

Figure 8. Original gray image (left) DTM (center) illumination of DTM (right).

into light and dark areas. To support the interpretation process the shadows have to be detected. This cannot simply be done by selecting all dark pixels because other dark objects may be present. Instead the illumination of the sun is simulated using the DTM (figure 8 center and right image). The segmentation of this image gives the raw shadows (figure 9 left). Due to the low resolution of the DTM the segmentation has to be improved:

Figure 9. Raw segmentation after illumination of DTM (left) and after noise cleaning (right).

1. Elimination of small areas.
2. All pixels of the remaining areas are selected which have the intensity of shadows (i.e., their gray values are inside a given range).
3. These pixels are used as seed areas for regiongrowing: Border pixels are added as long as the difference between their gray values and the mean value of the area is below a given threshold. In addition the number of iterations is limited.

The result of this post-processing can be seen in figure 9 right. These areas can now be used to support the interpretation e.g., to extend small areas of road hypotheses.

6 SUMMARY

To summarize, we developed an approach to aerial image segmentation which incorparates the DTM in combination with the corresponding orthophoto to gain a stable classification of objects even in complex scenes (hills, trees standing close to houses). The parameters used for preprocessing (size, height) are explicitly modelled or trained in the case of texture, and thus can easily be adapted.

The application has been realized using the image analysis tool $\mathcal{HORUS}$ (Eckstein$^+$1996). The runtime for a complete process, including dual rank, texture filters, median, classification, and post processing is about 40 seconds on an HP 712/60 with an image of size 512×512.

REFERENCES

[Dang$^+$,1994] T. Dang, O. Jamet, H. Maitre "Applying perceptual grouping and surface models to the detection and stereo reconstruction of building in aerial imagery" In *IntArchPhRS*, Vol 30, Part 3/1, 1994

[Eckstein$^+$,1991] W. Eckstein, W. Glock. "Development and Implementation of Methods for Segmentation of bond-wedges" In *Proceedings 4th Computer Analysis of Images and Patterns*, volume 5 of Research in Informatics, pages 153–161, R. Klette, editor, Dresden, 1991.

[Eckstein$^+$,1996] W. Eckstein, C. Steger. "Interactive Data Inspection and Program Development for Computer Vision" In *IntArchPhRS*, Vol 2656: Visual Data Exploration and Analysis III, 1996

[Fritsch$^+$,1994] D. Fritsch, M. Sester. "Test on image understanding" In *IntArchPhRS*, Vol 30, Part 3/1, 1994

[Haala,1994] N. Haala. "Detection of buildings by fusion of range and image data" In *IntArchPhRS*, Vol 30, Part 3/1, 1994

[Haralick,1992] R.M. Haralick, L.G. Shapiro. *Computer and Robot Vision* Addison-Wesley, Massachusetts, 1992

[Laws,1980] K.I. Laws. *Texture image segmentation* Ph.D. thesis, Dept. of Engineering, University of Southern California, 1980

[Shao$^+$,1994] J. Shao, W. Förstner. "Gabor wavelets for texture edge extraction" In *IntArchPhRS*, Vol 30, Part 3/2, 1994

[Steger$^+$,1995] C. Steger, C. Glock, W. Eckstein, H. Mayer, and B. Radig. "Model–based road extraction from images" In *Proceedings Automatic Extraction of Man-Made Objects from Aerial and Space Images*, Birkhaeuser, Ascona, 1995