ORTHOPHOTO FEATURE EXTRACTION AND CLUSTERING

Milos Kudelka[†], *Zdenek Horak*[†], *Vit Vozenilek*[*], *Vaclav Snasel*[†]

Abstract: In this article we use a combination of neural networks with other techniques for the analysis of orthophotos. Our goal is to obtain results that can serve as a useful groundwork for interactive exploration of the terrain in detail. In our approach we split an aerial photo into a regular grid of segments and for each segment we detect a set of features. These features depict the segment from the viewpoint of a general image analysis (color, tint, etc.) as well as from the viewpoint of the shapes in the segment. We perform clustering based on the Formal Concept Analysis (FCA) and Non-negative Matrix Factorization (NMF) methods and project the results using effective visualization techniques back to the aerial photo. The FCA as a tool allows users to be involved in the exploration of particular clusters by navigation in the space of clusters. In this article we also present two of our own computer systems that support the process of the validation of extracted features using a neural network and also the process of navigation in clusters. Despite the fact that in our approach we use only general properties of images, the results of our experiments demonstrate the usefulness of our approach and the potential for further development.

Key words: *Orthophotos, image analysis, neural network, clustering*

Received: December 1, 2011
Revised and accepted: April 6, 2012

1. Introduction

In the last two decades there have been many different methods proposed and implemented for the analysis of aerial photographs. Aerial data are one of the standard sources for the extraction of topographic objects. Classic applications include the detection and extraction of roads and buildings. Advances in the development of remote sensing devices increase the need timely and accurately to

[*]Vit Vozenilek
Department of Geography, Faculty of Science, Palacky University, Olomouc, tr. Svobody 26, 771 46 Olomouc, Czech Republic, E-mail: vit.vozenilek@upol.cz
[†]Milos Kudelka, Zdenek Horak, Vaclav Snasel
Department of Computer Science, FEI, VSB – Technical University of Ostrava, 17. listopadu 15, 708 33, Ostrava - Poruba, Czech Republic, E-mail: kudelka@inflex.cz, zdenek.horak.st4@vsb.cz, vaclav.snasel@vsb.cz

©ICS AS CR 2012

detect and extract various objects with increasing precision. It may also include other topographic objects such as forests and vegetation, agricultural use and parcel boundaries, hydrography, etc. Currently, this field of analysis falls under the paradigm of Object-based Image Analysis (OBIA), which is a sub-discipline of geoinformation science devoted to partitioning remote sensing imagery into meaningful image-objects with a focus on the generation, modeling and classification of these objects (see [4]).

Existing methods can be classified in different ways. One possibility is to divide them into two groups – automatic and semi-automatic. Semi-automatic methods – as opposed to the automatic ones – require human intervention, especially when tuning algorithms and judging results. Because of the many influences that contribute to the quality of aerial imagery, we usually cannot fully rely on automatic methods. This is expressed in a strong tendency to combine different methods and algorithms for solving various problems.

As will be described in detail in the following text, in our approach we proceed in the same way. For a description of images we use a set of general features (we do not use any knowledge base of known objects for extraction). These features are detected by a set of specific algorithms. We use a neural network for tuning these algorithms. The features are then detected automatically. To assess the quality of the detection we use our own application that incorporates the neural network again. This application visualizes how the system assesses a particular images. In the case of discovered inaccuracies, the user can retroactively affect the parameters of the automatic feature detection. Particular images are represented as vectors containing the values of individual features. The second automated step is to perform clustering and reflect the clusters back to the aerial image. After this automated procedure, it is the user's turn again. Here we have our other application, in which the user can explore a cluster of similar segments of an aerial image and search for objects of interests. The whole scheme is depicted in Fig. 1.

In the following section, we discuss related approaches. The third section contains a description of features in our detection system, while the fourth section recalls some basics of the tools and techniques used in the fifth section, which is focused on our experiment with orthophotos.

Remark The authors of this articles have participated in the development of a commercial Document Management System, which is used in several institutions in the Czech government. This system contains an image search module based on image features. There are thousands of still images stored in the DMS system (e.g. photos, scanned documents, plans, maps, logotypes, clip-art, etc.). After deploying the system and migrating the data, we discovered that from a user's point of view there are significant inaccuracies. Regardless of the parameters of the analysis having been tuned using a test dataset, we had to address the issue of detecting the bottlenecks in our settings. The result was the proposal of a process based on the use of the Formal Concept Analysis and Non-negative Matrix Factorization. This application of our image analysis system (using some of the below mentioned methods) is described in paper [9].

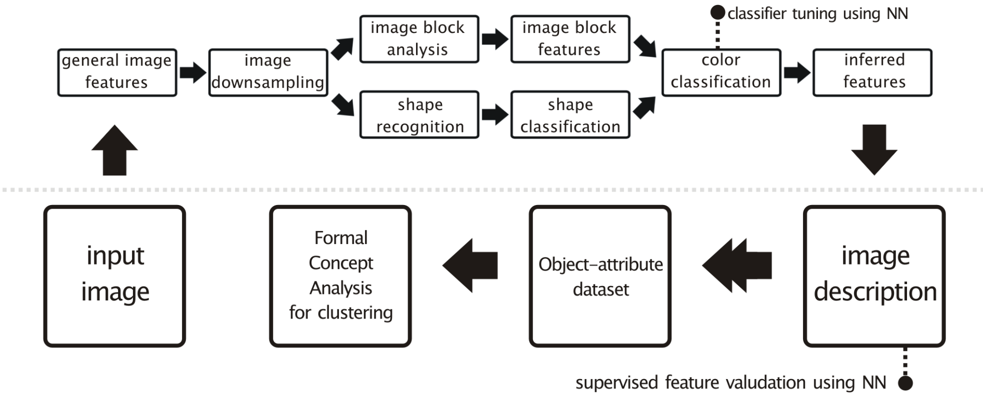

Fig. 1 *Image indexing scheme showing involvement of the neural networks and additional clustering using Formal Concept Analysis.*

2. Related Approaches

In this section we will describe approaches and methods that somehow correspond with our approach. Most of these methods perform image segmentation. But during the segment analysis phase, they utilize a certain level of terrain knowledge and also a knowledge of what topographic objects will be detected and extracted. Our approach is different mainly in that it is not focused on the detection of concrete topographic objects. Combining the analysis of general features and their use for the clustering and projection of these clusters back to the aerial photo, we can achieve interesting results. These results highly correspond with the users' expectations and are provided in a form that (with computer support) allows a high level of interactivity for a detailed image exploration.

The basis for all methods and algorithms for analyzing the orthophotos is digital image processing. Digital image processing is a set of technological approaches using computer algorithms to perform image processing on digital images [13, 18, 16]. Digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing [8]. Digital image processing may be modeled in the form of multidimensional systems rather than images that are defined over two or more dimensions. Some research deals with a new object-oriented classification method that integrates raster analysis and vector analysis, e.g. [12]. They combine the advantages of digital image processing (efficient improved CSC segmentation), geographical information systems (vector-based feature selection), and data mining (intelligent SVM classification) to interpret images from pixels to objects and thematic information.

Many different approaches dealing with the detection and extraction of man-made objects can be found in [2]. These are mainly methods focused on automatic road extraction and automatic building extraction. A summary and evaluation of methods and approaches from the field of automatic road extraction can be found in [15], while for the field of building extraction see [14]. For more recent approaches from the field of Object-Based Image Analysis (OBIA) you can see e.g. [4], a detailed summary of existing methods is described in [3].

3. Image Features

Our approach is to describe any image in terms of image contents and with concepts which are familiar to the users. In the following we present features we are capable of detecting. Some of the features are related to the whole image only, but many of them can also be used to describe various parts of the image.

3.1 Color features

According to an image's colors we are able to find out whether the image is gray-scaled, and if not, whether the image is toned into a specific hue. Also we can say if the image is light, dark or if the image is cool or warm.

- gray-scaled images
- color-toned images
- bright or dark images
- images with cool or warm color tones

The last group of features is color features. We want to describe the image in terms of colors in the same way as a human will, but it does not suffice only to count the ratio of one color in the image or in some areas of the image. A more complex histogram is also not enough. We should consider things like dithering, JPEG artifacts and the subjective perception of colors by people. Using color spatial distribution, color histograms and below mentioned shapes recognition, we are also able to detect the background color. The colors we are currently able to detect are:

- red, green, blue, yellow, turquoise, violet, orange, pink, brown, beige, black, white and gray
- background color

Remark The background color (in our approach) denotes a larger continuous unicolor part of the image which is adjacent to the border of the image and represents a significant portion of the image.

Color features detection Low-level color features were detected using a combination of their spatial distribution and a comparison to their prototypes. The first version of the system contained prototypes that were constructed manually using our subjective perception. However, this approach was not general enough, therefore we have created a set of training image patterns with manually annotated color features. To deal with human perception, we have averaged the annotation results among several annotators. Using this set, we have trained the artificial single-layer

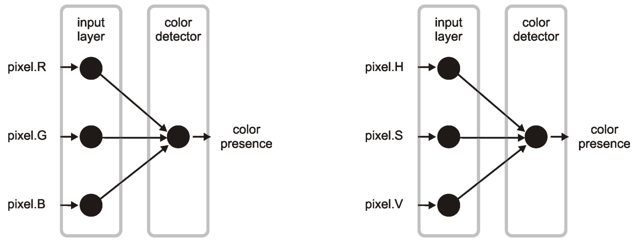

Fig. 2 *Illustration of simple neural network color detector.*

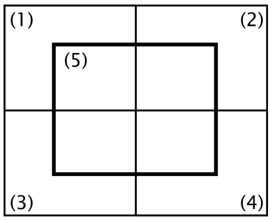

Fig. 3 *Image divided to regions.*

feed-forward neural network (see [19, 1]) confidently to identify the mentioned features. This network was very similar to the network used in the whole application, which is described below in detail.

We have used the pixels of particular patterns in different color models (as different models are suitable for different color features) as an input. Trained neurons (respectively their input weights and hidden threshold) were then transformed (using the most successful color model) into color feature prototypes (see Fig. 2). We detect all of the mentioned features as fuzzy degrees, but for selected applications we scale them down to the binary case.

3.2 Image regions

The mentioned features can be detected in the whole image. But, if we try to describe an image without knowing what exactly in the image is, we may end up with something like: *Bright photo with blue region in the upper part and a green region in the lower part.* Therefore, it makes sense to detect the features also in smaller areas of the image.

The question is how to choose these areas. The simplest choice is to make them fixed (of course, proportional to the size of the image). We divide the image into four proportional regions (as shown in Fig. 3). According to human perception and basic photography rules, we also add one central region (marked as number five) sized using the well-known Golden Ratio. This region emphasizes the most obvious parts and features perceived by people at first sight.

Clearly, this approach works poorly in some cases. For example, consider Fig. 8. The left image is a region in an orthophoto. If we split the image into several regions using different resolutions, we may end up with either detected quads (the middle image) or triangles (the right image). The system will report some similarities due to features computed from the whole image, but in large databases there will be a lots of images with the same, or even higher, similarity degrees.

Image segmentation For these reasons, we have used the previous approach only for basic features. To obtain more precise information about the processed image, we have decided to employee an image segmentation technique. Using the Flood fill algorithm (with eight directions, for details see [7]) we were able to separate regions with same, or almost the same, color. But to be able to index these shapes, we need to describe them. We have calculated the center of this shape and using this point and different angles we have sliced the shape into several regions (see 16 regions in Figs. 5, 6, and 7). For each region, we have computed the maximum distance from the center. Following the changes of this distance (peaks, regularities) we are able to distinguish between different basic shapes (rectangles, circles, triangles, etc.).

Of course, this approach is not general. We use it only for bigger shapes and we ignore possible holes within the shapes. Because we use mostly downsampled versions of source images, we can guarantee the effectiveness of processing. And because we use high quality downsampling, our results are similar to a person's first glance. The shapes we are able to detect are: line, rectangle, circle, triangle and quad.

At this moment, we detect shapes separately, but in the final description we keep only information whether at least one shape of such kind has been detected (i.e. the image contains one triangle) or whether there are multiple shapes of such kind (i.e. the image contains more triangles).

Currently we are thinking of using obtained distances not only for shape identification, but also for shape description. The same shape can be scaled, moved or rotated in different images, but the description using relative distance changes is still the same (up to the index rotation). The more different angles we use, the more precise description we obtain.

Anomalies Since the orthophotos are created from long distance, interesting objects are often relatively small and vaguely bound in the image. For this reason we have incorporated the concept of anomalies (see [5] for a recent survey). We consider an anomaly to be a shape, that:

- is formed by similar pixels,
- cannot be – due to its size – reliably classified as being one of the previously mentioned shapes,
- has other than background color.

As you will see in the experiment section, this concept became very important in our approach. Figure 4 contains highlighted samples of various shapes detected in the orthophotos.

Fig. 4 *Various shapes detected in orthophotos - rectangles, triangles, lines and anomalies.*

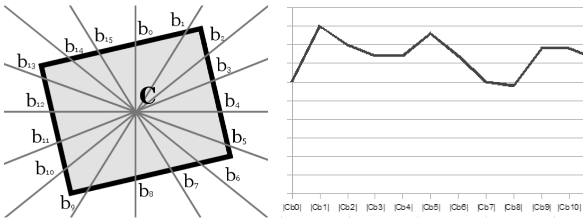

Fig. 5 *Rectangle detection.*

Remark Our system is also capable of detecting additional features, which are irrelevant to this particular application, such as: image size and aspect ratio features, human skin colors and overall properties of the image. Using the number of colors, their spatial relations, positions, local changes and histograms, we are able to distinguish between photos, clip-art, schemes and pencil-drawn images. Using the regularity of color changes and some statistics, we are able (with some probability) to say whether the image is artificially generated and if it contains text shapes. We can also calculate several image checksums to allow efficient duplicity detection.

4. Preliminaries

This section briefly discusses three mathematical methods which will be used in the experiment section. The first one – the neural network – is used both in feature detection and in the validation of this detection. The second one – the Formal

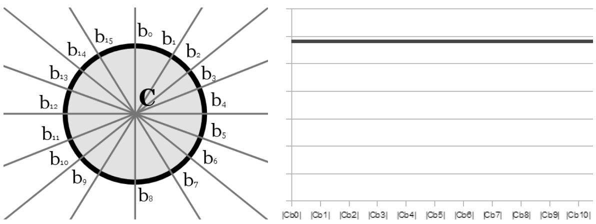

Fig. 6 *Circle detection.*

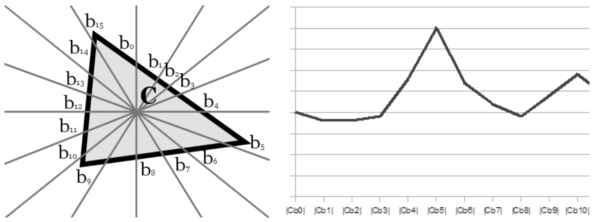

Fig. 7 *Triangle detection.*

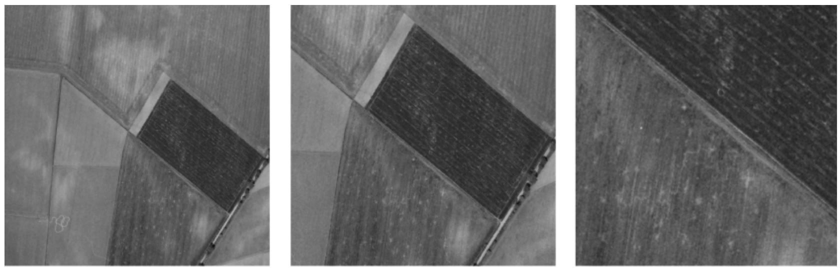

Fig. 8 *Ambiguity of shape detection caused by splitting the image using different resolutions.*

Concept Analysis – is used to cluster images using detected features. The last one – the Non-negative Matrix Factorization – helps us to handle the complexity problems which arise when dealing with large-scale data.

4.1 Neural network

The **Neural network** (or more precisely, "artificial neural network") is a computational model inspired by biological processes. This network consists of interconnected artificial neurons which transform excitation of input synapses to output excitation. Most of the neural networks can adapt themselves. There are many different variants of neural networks. Each variant is specific in its structure (whether the neurons are organized in layers, whether the neurons can be connected to themselves, etc.), learning method (the way the neural network is adapted) and neuron activation function (the way the neuron transforms input excitation to output excitation) and its parameters.

For our purposes we use a structure consisting of an input layer of neurons, several inner hidden neuron layers and one output layer of neurons. For the learning method we use **supervised learning**, where the network is presented repeatedly with specific samples, which are propagated towards the network output. This output is compared with expected results and the network is, using calculated error, adapted to minimize this error. The learning process finishes after a predefined number of learning epochs or if the error rate decreases under a predefined constant. After the learning phase, the neural network can be presented with another group of samples and provides its output. For more details on neural networks consult [19], [1] or see [17] for this particular case.

Our particular network is illustrated in Fig. 10. We have decided to use classic bipolar-sigmoid (because we needed to represent both positive and negative examples) as an **activation function** of neurons (having $\beta = 2$):

$$f(x) = \frac{2}{1 + e^{-\beta x}} - 1$$

Simple **backpropagation** has been used as a learning algorithm:

$$\triangle w_{ij}(t+1) = \eta \frac{\sigma E}{\sigma w_{ij}} + \alpha \triangle w_{ij}(t)$$

The basic idea of this algorithm is to calculate the total error E of the network (computed by comparing real outputs of the network with expected ones) and then change the weights $\triangle w_{ij}(t+1)$ of the network to minimize this error. The learning rate parameter η controls the speed of weight changes. To speed up learning, we use momentum α – which updates the weight in each step also with the value from the previous step $\triangle w_{ij}(t)$.

4.2 Formal concept analysis

Formal Concept Analysis (shortened to FCA, introduced by *Rudolf Wille* in 1980 [6]) is based on the understanding of the world in terms of objects and attributes. It is assumed that a relation exists to connect objects to the attributes they possess.

Formal context $C = (G, M, I)$ is a triplet consisting of two sets, G and M, with I in relation to G and M. The elements of G are defined as objects and the elements of M are defined as attributes of the context. In order to express that an object $g \in G$ is related to I with the attribute $m \in M$, we record it as gIm or $(g, m) \in I$ and read that object g has the attribute m.

For set $A \subseteq G$ of objects we define $A^{'} = \{m \in M \mid gIm \ for \ all \ g \in A\}$ (the set of attributes common to the objects in A). Correspondingly, for set $B \subseteq M$ of attributes we define $B^{'} = \{g \in G \mid gIm \ for \ all \ m \in B\}$ (the set of objects which have all attributes in B).

Formal concept of the context (G, M, I) is a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A^{'} = B$ and $B^{'} = A$. We call A the extent and B the intent of the concept (A, B). $\mathcal{B}(G, M, I)$ denotes the set of all concepts of context (G, M, I) and forms a complete lattice (the so-called **Galois lattice**).

Galois lattice may be visualized by the so-called Hasse diagram. In this diagram, every node represents one formal concept from the lattice. Nodes are usually labeled by attributes (above the node) and objects (below the node) possessed by a concept. For the sake of clarity, so-called reduced labeling is sometimes used (see Fig. 13 for illustration), which means that attributes are shown only at the first node (concept) they appear in. This holds reciprocally for objects. These two labelings are equivalent. For the purpose of this article we denote objects of the concept in the visualization only by their quantity.

Roughly speaking, Formal Concept Analysis can be used to find and visualize all meaningful groups of objects and their attributes that are present in a system. If the node in the lattice visualization is labeled by some particular attribute, that means that all nodes below (with respect to the edges) also possess this attribute. The same stands for objects – just in reverse order.

4.3 Non-negative matrix factorization (NMF)

Matrix factorization methods are heavily used methods in many different fields. Roughly speaking, their principle lies in decomposing one matrix into several smaller ones. After multiplying these matrices again, we get the same (or almost the same) matrix as the original one.

Non-negative matrix factorization differs from other rank reduction methods by the use of constraints that produce non-negative basis vectors, which make the concept of a parts-based representation possible. Reference [11] first introduced the notion of parts-based representations for problems in image analysis or text mining that occupy non-negative subspaces in a vector-space model. Basis vectors contain no negative entries. This allows only additive combinations of the vectors to reproduce the original.

Common approaches to NMF obtain an approximation of V by computing (W, H) pair to minimize the Frobenius norm of the difference $V - WH$. Let $V \in R^{m\times n}$ be a non-negative matrix and $W \in R^{m\times k}$ and $H \in R^{k\times n}$ for $0 < k \ll min(m, n)$. Then the minimization problem can be stated as $min\|V - WH\|^2$ with $W_{ij} > 0$ and $H_{ij} > 0$ for each i and j.

Having the approximation of V as WH, we can ignore the least important parts of these matrices and reconstruct the original matrix using the remaining parts only. Therefore, we obtain a simplified version of the original data. There are several methods for computing NMF. We have used the multiplicative method algorithm proposed by Lee and Seung [11, 10].

5. Experiment

5.1 Proposed model

Now we will describe our model in detail (depicted in Fig. 1). Our aim is automatically to identify similar image regions, which can be later semi-automatically explored. We start with an orthophoto image. This image has to be normalized to allow comparison with different orthophotos. Once normalized, we split the image into particular regions. The size of the regions in calculated using image resolution and sensing distance and expected size of particular objects. In certain situations we include also an overlap of regions to minimize the previously mentioned problem of splitting ambiguity.

Particular regions are then automatically processed using the described image classifier. This results in a dataset of particular regions and identified image features. Now we can verify the validity of detected image features using supervised training and our experimental application. This task can also be used to select a suitable set of image features for further exploration. This step is required because the visual similarity is very subjective and context-dependent. We need to capture the user's point of view.

In the next step, we perform an automatic clustering using Formal Concept Analysis (based on selected features). The result of this clustering is a navigational structure which is interconnected with the original orthophoto and allows the user to explore the whole region. Both of these crucial aspects of our approach are illustrated in the following sections.

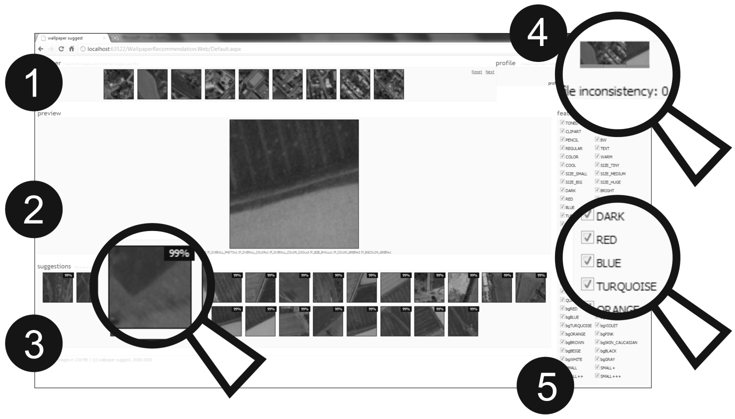

Fig. 9 *Screenshot of the validation application with highlighted and zoomed parts of its UI – (1) image gallery, (2) image preview, (3) suggested images, (4) user profile, (5) considered features.*

5.2 Dataset description

For our experiment we have used several datasets containing orthophotos from the regions of Poland and the Czech Republic containing heterogeneous landscapes such as green fields, agricultural land, valleys, softwood and hardwood forests, villages, towns, etc. We have normalized the color histogram of the photos and then we have split the original photos ($8,000 \times 8,000$ pixels) into a grid of square images (200 x 200 pixels), which resulted in 1,600 images per orthophoto. For our testing purposes we have also tried squares having 100 x 100 pixels, which had no significant influence on the result. Squares having 50 x 50 pixels have shown to be too small for our purposes.

In our approach we suppose that an orthophoto is the only available information about the landscape – therefore we do not consider any height data or additional images in a different light spectra. We also do not use any computationally demanding preprocessing methods, all decisions are based on local information only.

Remark We use binary vectors whose dimensions are user-dependent (the user can specify which features will be used during the experiment).

5.3 Feature validation

To verify that our set of features is capable of representing the user point of view on the images content, we have created a web application for image suggestions. In the first step, the user is presented with several random images from the data. He/she marks these images as interesting or not interesting. Using this process, the user search profile is created. In the second step, the application tries to understand this profile (a set of positive and negative examples) using an artificial multilayer feed-forward neural network. In the last step, the trained network is presented

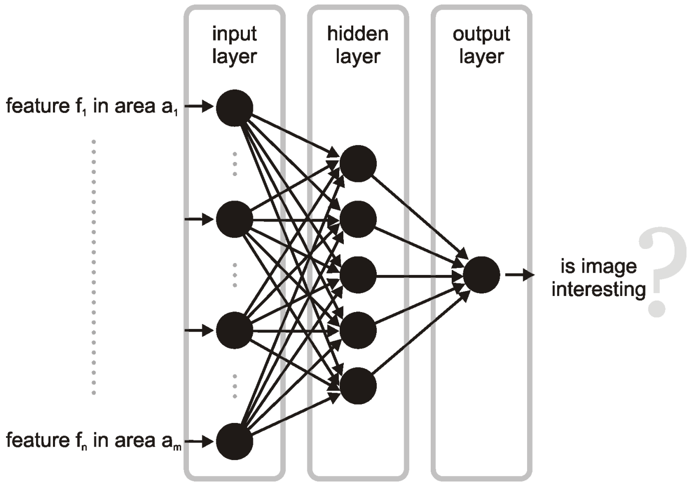

Fig. 10 *Illustration of neural network used in application.*

with the whole dataset and suggests images which may be potentially interesting to the user.

The user can clarify his/her profile by marking further images and the process is repeated. We have used part of the profile for training and the rest for the validation of the profile to verify the meaningfulness of this profile. The score of presented images is an indication for users to add more positive (if the overall score is too low) or negative (the overall score being too high) examples.

Network parameters Parameters of the neural network have been selected as follows (see Fig. 10): 610 input neurons (input activation represents the degree of individual feature presence), 5 hidden neurons and one output neuron (representing the degree of image acceptance). The learning rate was $\eta = 0.1$ and momentum $\alpha = 0.1$. The maximum number of iterations per learning epoch was set to 1,000. The number of input neurons corresponds to the number of features in different regions of the image. Remaining parameters have been selected after several attempts of being subjectively the best. A larger number of hidden neurons often caused the overtraining of the network (good performance on the training samples with very limited ability of generalization). A larger number of iterations produced no significant improvement. Lower values failed to comply with user judgments.

Application description This application (see Fig. 10) has been created as an ASP.NET Web application on the Microsoft .NET platform utilizing several other technologies such as CSS/JavaScript to improve user experience. Most of the computation time is used during the image dataset indexing, which is done only once and can be precomputed offline. The indexing of particular images takes on average 1.84 seconds and can be easily parallelized as the indexing of every particular image is completely independent.

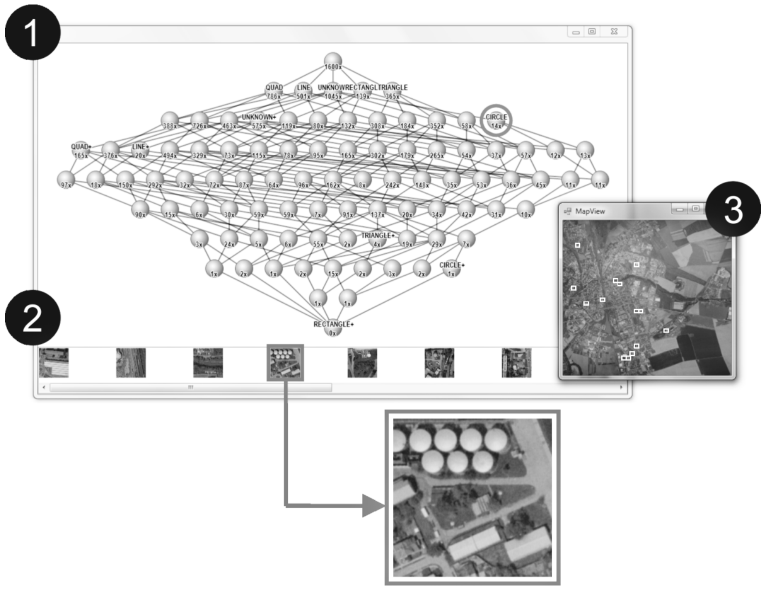

Fig. 11 *Screenshot of the lattice navigation application with highlighted parts of its UI – (1) computed lattice with "CIRCLE" concept selected, (2) objects from the selected concept, (3) objects from the selected concept highlighted in the original photo.*

The neural network is recreated with every request, but in a high-load environment can be stored between requests. Application memory contains indexed image signatures only, therefore the whole application is well scalable. In our testing environment we have been able to run this application easily on an Intel 2.13 GHz processor and 4 GB RAM with a dataset containing several thousand images. Clearly the process of running the neural network with particular image signatures in every step of recommendation has its computational limits, but these limits lie far beyond the boundaries of the purpose of our experiment.

We have performed several user testing sessions where we have selected the presented set of features as being the most suitable for our purposes. Using this process, we made sure that normal users can understand image analysis systems based on selected features and these users were normally able to find expected results after giving two or three positive and negative examples. The discrepancy between the user's expectations and the output of the system is a suggestion for another iteration of feature detection fine-tuning.

5.4 Feature lattice

Application description The application used in this experiment is illustrated in Fig. 11. It is a classic Windows Forms application written in C# on the Microsoft .NET platform. This application allows users to visualize the concept lattice of the image features and select interesting concepts. The objects from these con-

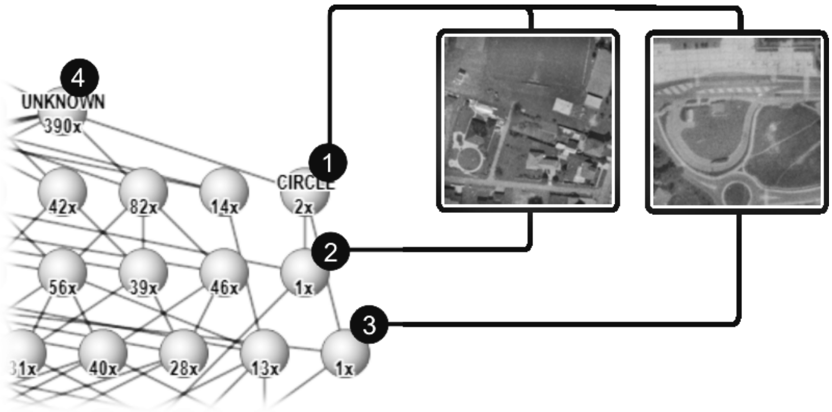

Fig. 12 *Navigation in the lattice - (1) concept of two segments containing unidentified shapes and circles, (2) concept of the first segment, (3) concept of the second segment, (4) concept of 390 segments containing unidentified shapes.*

cepts (particular images from the dataset) are then shown in the lower part of the application and also highlighted in the original photo. The user can navigate throughout the whole lattice and discover obtained concepts. If the concept lattice is too large (as seen from the user's perspective, as the computers can handle much larger lattices than humans), the user can show only a reduced version of the lattice.

In this experiment we neglect features specifically to some part of the image and consider only features which were detected to be relevant to the whole image.

Lattice navigation The principle of lattice navigation is explained in Fig. 12, which represents a portion of a concept lattice computed from shape features detected in an orthophoto of the border of a city. We can start with a concept denoted as "CIRCLE" / 2x (labeled by number 1 in the figure) which represents the concept of all segments (2) where circles have been detected. These segments are shown in the right part of the figure. Now we can navigate to concepts labeled by numbers two and three. Each one represents a concept with only one segment and a division of the concept labeled by number 1. From the user's viewpoint this division is natural, as one segment contains a traffic roundabout and the second one only the round part of a garden. The reasons why these concepts are divided in the lower level of the lattice can be found by investigating the links of these concepts to the left part (not shown) of the lattice. If we look above, we can see that all three mentioned concepts are linked to a concept called "UNKNOWN" / 390x. This information can be read as: in all segments where circles have been detected also unidentified shapes have been detected.

Shapes The upper part of Fig. 13 shows a concept lattice computed from shape features detected in an orthophoto of a larger city. We have selected the concept which contained all objects (image segments) where circles have been detected.

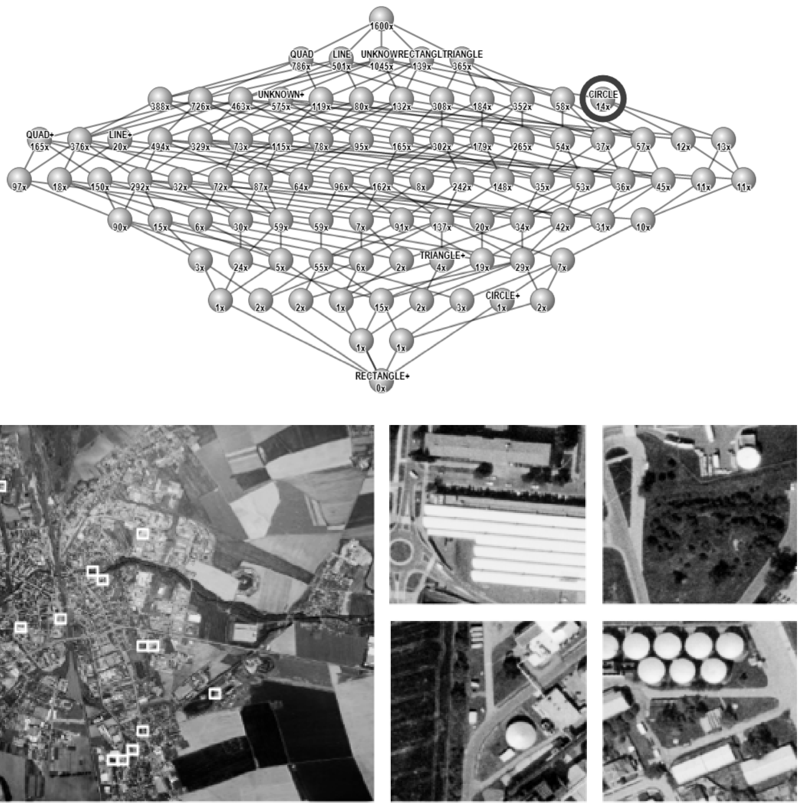

Fig. 13 *Circles - "circle" concept highlighted in the lattice (upper part), segments with detected circles highlighted in the photo (lower left part) and several samples of these segments.*

These segments have been highlighted in the original photo (the lower left part of the figure). Several samples of these segments are shown in the lower right part of the figure. We can see that the circles often represent traffic roundabouts and chemical tanks. Using additional logic describing the color of expected shapes or relative position to other shapes, such as lines representing roads, these objects can easily be identified and classified.

Regular shapes – especially circles – are rarely seen in orthophotos of true nature. Exceptions are often accompanied by specific geological conditions which might be expected a priori.

Anomalies As we can see in Fig. 14, anomalies represent a relatively reliable way to detect human settlements. We have scaled the amount of detected anomalies per segment to four degrees. A lattice constructed from these degrees can be seen on the left part of the figure. The whole image has been split into 1,600 segments, in which 759 contained at least one anomaly (concept labeled by "SMALL"), 354 of

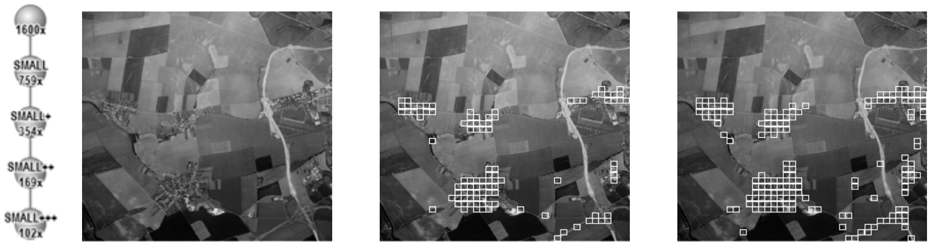

Fig. 14 *Anomalies - (i) lattice of anomalies features, (ii) original orthophoto, (iii) photo with highlighted segments containing high number of anomalies and (iv) slightly less number of anomalies.*

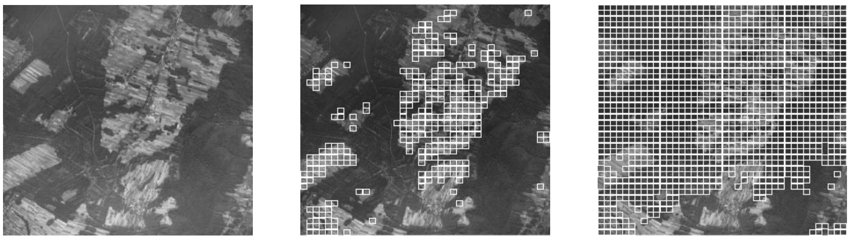

Fig. 15 *Colors and backgrounds - (i) original orthophoto, (ii) photo with highlighted segments containing the color yellow, (iii) photo with highlighted segments having a green background.*

them contained more anomalies (concept labeled by "SMALL+"), 169 segments contained a large number of anomalies (concept labeled by "SMALL++" containing segments highlighted in the midle part of the figure) and finally 102 segments containing many anomalies (labeled by "SMALL++" and highlighted in the right part of the figure). Comparing the last two concepts, we can identify peripheral areas of the settlements where buildings mix with fields and other countryside.

Colors and backgrounds Fig. 15 shows a typical country landscape, a mixture of forests with agricultural land and several villages. The middle part of the figure highlights segments where our system detected significant signs of the color yellow. We have manually verified these segments and they represented some specific kind of planted field. However, when considering more common colors, such as brown or beige, the interpretation of these segments is much more complex as weather conditions, time of day, season and camera optics should be considered. This can be illustrated in the right part of the figure which highlights photo segments where green has been identified as the background color. Notice the lower part of the photo where a blue and turquoise background has been detected instead of the green one. The human eye will probably ignore this fact (as there are only small differences between particular segments), but the overall toning of these segments have truly shifted to a blue tint because of the camera configuration.

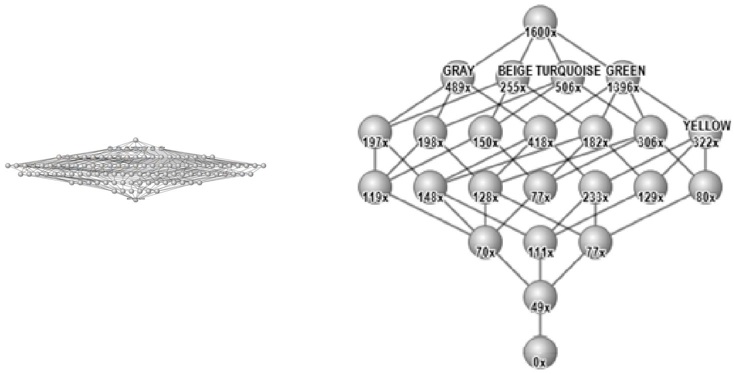

Fig. 16 *Concept lattice computed from color features – original and reduced to rank 5.*

Lattice reduction Sometimes (see left part of Fig. 16), when there is a large number of features and their combinations in the dataset, the lattice can become too complicated to be easily understood by the user. In this situation, we can employ the technique of matrix reduction (aforementioned NMF in this case) and reduce the original data into a lower dimension. In the right part of the figure, we can see the reduced version of the lattice which maintained the most important properties (with respect to the number of segments and features involved) of the original lattice. Interpretation of the reduced lattice is, therefore, imprecise, but gives the user an overview about the situation in the dataset. The lattice can be read as follows: in most images the system found the color green. There were also many segments containing the color yellow, but almost every single one contained the color green too. There were also many segments with the color gray and many of them were accompanied by the color green again. The color turquoise was identified in a third of the segments (as previously explained). Also there were around two hundred segments containing the color beige. There were only several dozen segments containing all the mentioned colors.

5.5 Experiment evaluation

As we have mentioned in the related approaches section, the existing approaches usually focus on detecting previously known specific elements in the orthophotos. Our approach differs in clustering self-similar parts of the orthophoto and further semiautomatic processing. Therefore, it is not easy to evaluate the effectiveness of our approach by a direct comparison with other approaches. However, there are other things that can prove the feasibility of our method.

One of the main goals of this paper was to identify the most suitable tasks where our approach can be used. Therefore, we have performed our experiments using different orthophotos without any knowledge of the terrain and location. Also the results of our experiments have been manually validated. Here are three main outcomes we have obtained:

- Ortophoto analysis systems cannot easily rely on photo colors as there are too many factors which can influence the resulting color in the image.
- The man-made objects in the orthophoto can be succesfully detected using shapes and anomalies detection.
- An appropriate combination of general features, such as colors and detected shapes of particular image regions, can be used. This is possible even without prior knowledge of the terrain or specific types of the searched topographic objects. It is possible – with high accuracy – to detect typical man-made objects, such as factory areas and facilities or rugged farm terrain, densely or sparsely populated areas, an open space in dense housing, isolated dwellings, meadows and clearings in the forest, etc.

6. Conclusions

Using several mathematical models and methods, such as neural networks, FCA and NMF, we have developed and described a system which can analyze orthophotos, detect user-oriented features in the photos, visualize the structure of the region and allow the user to effectively navigate through different types of segments of the region. We have investigated the similarities between different image segments and sources of these similarities in different kinds of maps. A promising area seems to be the study of how the computed characteristics change, for example by comparing orthophotos of the same region from different time periods.

References

[1] Arbib M. A.: The handbook of brain theory and neural networks, The MIT Press, 2003.

[2] Baltsavias E. P., Gruen A., Van Gool L.: Automatic extraction of man-made objects from aerial and space images (III), 2001.

[3] Blaschke T.: Object based image analysis for remote sensing, ISPRS Journal of Photogrammetry and Remote Sensing, Elsevier, **65**, 1, 2010, pp. 2–16.

[4] Blaschke T., Lang S., Hay G. J.: Object-based image analysis: spatial concepts for knowledge-driven remote sensing applications, Springer Verlag, 2008.

[5] Chandola V., Banerjee A., Kumar V.: Anomaly detection: A survey, ACM Computing Surveys (CSUR), **41**, 3, 2009, pp. 1–58.

[6] Ganter B., Wille R.: Formal Concept Analysis: Mathematical Foundations, Springer-Verlag, New York, 1997.

[7] Glassner A.: Fill'Er Up!, IEEE Computer Graphics and Applications, pp. 78–85, 2001.

[8] Gonzalez R. C., Woods R. E.: Digital image processing. London, Pearson Education, 954 pages, 2008.

[9] Horak Z., Kudelka M., Snasel V.: FCA as a Tool for Inaccuracy Detection in Content-Based Image Analysis, IEEE International Conference on Granular Computing, 2010, pp. 223–228.

[10] Lee D., Seung H.: Algorithms for Non-Negative Matrix Factorization, Advances in Neural Information Processing Systems, **13**, 2001, pp. 556–562.

[11] Lee D., Seung H.: Learning the parts of objects by non-negative matrix factorization, Nature, **401**, 1999, pp. 788–791.

[12] Li H. T., Gu H. Y., Han Y. S. et al.: Object-oriented classification of high-resolution remote sensing imagery based on an improved colour structure code and a support vector machine, International Journal of Remote Sensing, **31**, 6, 2010, pp. 1453–1470.

[13] Lillesand T., Kiefer R.: Remote Sensing and Image Interpretation, New York, John Wiley and Sons, 724 pages, 2000.

[14] Mayer H.: Automatic object extraction from aerial imagery – A survey focusing on buildings, Computer Vision and Image Understanding, Elsevier, **74** 2, 1999, pp. 138–149.

[15] Mena J. B.: State of the art on automatic road extraction for GIS update: a novel classification, Pattern Recognition Letters, **24**, 16, 2003, pp. 3037–3058.

[16] Pitas I.: Digital image processing algorithms and applications, New York, Wiley, 419 pages, 2000.

[17] Riedmiller M.: Advanced supervised learning in multi-layer perceptrons – From backpropagation to adaptive learning algorithms, Computer Standards & Interfaces, **16**, 3, 1994, pp. 265–278.

[18] Umbaugh S. E.: Computer imaging: digital image analysis and processing, London, Taylor & Francis, 659 pages, 2005.

[19] Yegnanarayana B.: Artificial neural networks, PHI Learning Pvt. Ltd., 2004.