UNIVERSITY OF JOENSUU
COMPUTER SCIENCE
DISSERTATIONS ??

ALEXEY PODLASOV

PROCESSING OF MAP IMAGES FOR IMPROVING QUALITY AND COMPRESSION

ACADEMIC DISSERTATION

To be presented, with the permission of the Faculty of Science of the University of Joensuu, for public criticism in Louhela Auditorium of the Science Park, Länsikatu 15, Joensuu, on December ??th, 2007, at 12 noon.

UNIVERSITY OF JOENSUU
2007

Supervisor Professor Pasi Fränti
Department of Computer Science and Statistics
University of Joensuu
Joensuu, Finland

Reviewers Professor Jukka Teuhola
Department of Information Technology
University of Turku
Turku, Finland

Associate professor Su Yang
Department of Computer Science and Engineering
Fudan University
Shanghai, China

Opponent to appear

ISSN 1238-6944
ISBN 952-458-349-6
Computing Reviews (1998) Classification: I.4.2, I.4.3, I.4.10, C.3
Yliopistopaino
Joensuu 2007

Processing of map images for improving quality and compression

Alexey Podlasov

Department of Computer Science and Statistics
University of Joensuu
P.O.Box 111, FIN-80101 Joensuu FINLAND
apodla@cs.joensuu.fi

University of Joensuu, Computer Science, Dissertations 7
Joensuu, 2007, 204 pages
ISSN 1238-6944, ISBN 952-458-349-6

Abstract

The thesis is dedicated to processing of map images for improved filtering and compression. The main purpose of the study is the developing of optimized algorithms taking the properties of map imagery into account for improving performance of map processing techniques. The research consists of two major areas.

The first topic is layer-wise enhancement and compression of map images. Firstly we propose binary morphological restoration technique for semantic layers of the map. The proposed technique reconstructs layers corrupted by color overlapping and allows achieving better map image compression. Secondly, we study bit plane separation, predictive modeling and highly optimized context modeling for compression of natural and palette images. Extensive evaluation of standard and novel compression techniques is presented.

The second part of the thesis is dedicated to context tree modeling for filtering and compression of map images. Firstly, we propose generalized context tree based statistical filter for map images. The filter uses variable-size local probability estimator for effective detection of statistical inconsistencies and preserving the detailed areas of the image. Secondly, we propose the using of optimized free tree modeling and better color quantization for recently proposed progressive lossy-to-lossless compression algorithm. The new algorithm provides better compression and quality of the lossy progression. Thirdly, we propose a novel scheme for lossy compression of scanned map images based on color quantization and generalized context tree modeling. The new approach provides better lossy performance in sense of compression-quality tradeoff.

Keywords: Map images, lossless compression, lossy compression, reconstruction, filtering, context tree, bit plane separation, mathematical morphology.

Acknowledgments

The work presented in this thesis was carried out …

Joensuu, December 2007

Alexey Podlasov

List of original publications

P1. Podlasov A., Ageenko E., Fränti P., Morphological reconstruction of semantic layers in map images. *Journal of Electronic Imaging*, 15(1), 013016, January–March 2006.

P2. Podlasov A., Fränti P., Lossless image compression via bit-plane separation and multi-layer context tree modeling, *Journal of Electronic Imaging*, 15(4), 043009, October–December 2006.

P3. Podlasov A., Fränti P., Merge-based color quantization and context tree modeling for compression of color-quantized images", *IEEE International Conference on Image Processing (ICIP'06)*, Atlanta, Georgia, USA, pp. 2277–2280, October 2006.

P4. Podlasov A., Kopylov P., Fränti P., Statistical filtering of raster map images using a context tree model, *Int. Conf. Signal-Image Technology & Internet-based Systems (IEEE-SITIS'07)*, Shanghai, China, December 2007. (to appear).

P5. Podlasov A., Kolesnikov A, Fränti P., Lossy compression of scanned map images, *17th International Conference on Computer Graphics and Vision (Graphicon'07)*, Moscow, Russia, pp. 79–83, June 2007.

Contents

1 Introduction

The use of *Geographical Information Systems* (GIS) to provide users with digital navigation information is widespread and becoming more and more popular. Examples of this are the personal car navigator and PDA-based digital topographic map for foresters, geologists and engineers. Usually the architecture of such systems does not depend on the application area. A typical example is a system where the user’s coordinates are obtained via a satellite positioning service, such as *Global Positioning System* (GPS), and geographical information about the current location is obtained from a local or remote map database.

Map images in a database can be stored in two principally different formats: *vector* and *raster*. The vector format assumes that the map is stored as a set of geometrical primitives (lines, symbols, curves, textures) describing the image content. Each primitive is described by a set of required parameters. For example, a straight line segment is described by four real numbers defining the end points. In order to be displayed, the data must be projected on a plane with the desired scale and rotation, and then drawn on the screen of the client device. However, some geographical data is still unavailable in vector form, and the only sources are traditional maps printed on paper sheets. Although vectorization is an actively developing technology [1][2], a universal non-supervised vectorization algorithm still does not exist. It is often too expensive to manually convert such data into vector format, and therefore storage in raster format can be a better solution.

Raster format assumes that the image is stored as an array of values that represents the rectangular matrix of pixels forming the picture. Depending on the application, the storage of one pixel requires one bit for binary images, one byte for gray-scale or indexed images, three bytes for true color, and even more in the case of a multi-spectral image. A natural advantage of raster format is that it does not require any additional processing for displaying the image. The image can be represented immediately after the data is received. A typical way of combining vector and raster format in the same system is to use the global database stored data as vectors and provide the user with a raster image converted from the vector original to represent the area needed.

The main drawback is that raster data is not flexible when some transformation of the image is needed. For example, zooming, rotation and projection of the image is impossible without degradation. The storage size needed is also a problem. In contrast to vectors, raster images store all pixels of the line instead of coordinates of the corresponding segment. In the case of geographical maps, the size of digitized images can be huge. For example, in maps from *National Land Survey of Finland* (NLS) topographic database [3], a single map sheet of 10×10 km^2 1:20,000 scale is represented by a single image of 5000×5000 pixels, which requires about 70 Mbytes of memory to be stored. Another example is a map of A4 size scanned with 300 dpi in true color, which results in 2500×3500 pixel image requiring about 25 Mbytes. The necessity for image compression is obvious since more efficient storage space utilization as well as faster map transmission is needed to make digital navigation services more usable, reliable and cheaper.

Features that are distinctive for map images can be characterized as follows. A map image contains only a few unique colors; in cases where the image is converted from a vector source the number of colors rarely grows higher than several tens. A map image also contains a lot of uniform areas representing particular regions like water, forests or background. The areas of the map are usually distinctly separated from each other. This makes a map image which contains sharp and easily localized edges. Smooth gradation is rarely present in map images. A typical map contains thin details and symbols, the presence of which is vital for the semantics of the map. Features of map images are illustrated in Figure 1.

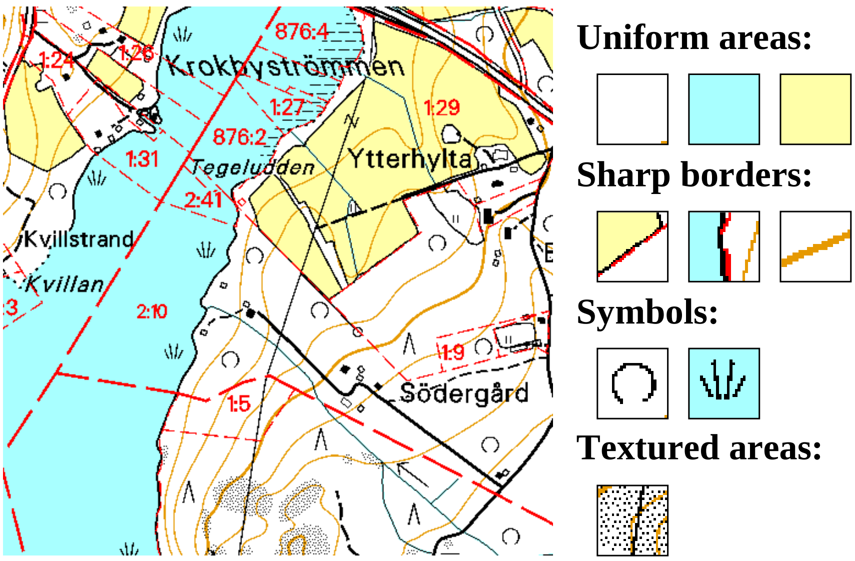


Figure 1. Features of a map image.

In addition to raster maps, which are converted from vector databases, there is another class of scanned map images. These images are produced by digitizing printed paper maps. Scanned map imagery unintentionally combines the properties of natural imagery and converted raster maps. Edges and details on a scanned map are smooth since the image is acquired with a physical sensor of the scanner. Besides this, a scanned map image may have special patterns such as dithering. The number of paints available in typography is limited and color gradation is usually represented as a pattern of color dots. These structures are acquired by a scanner and appear in the scanned map image. Typical features of scanned map images are illustrated in Figure 2.

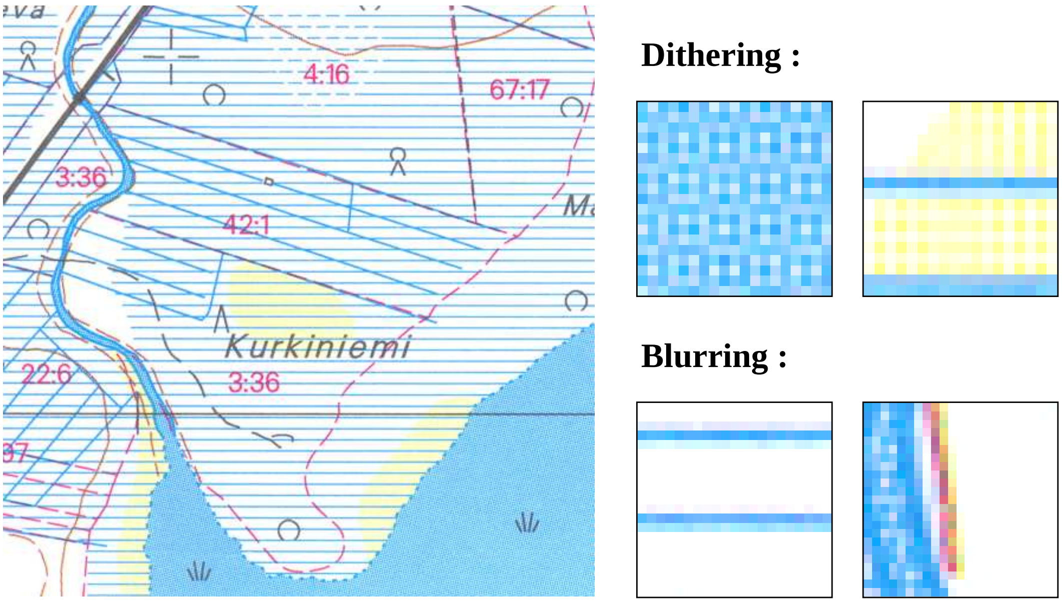


Figure 2. Features in a scanned map image.

A typical scanned map can contain hundreds of thousands of unique colors in contrast to the converted maps that contain only a few. Though these colors are visually grouped around the original colors in the distribution in the color space, they are far from being easily clustered. A visual example is presented in Figure 3 where the color distribution for a sample scanned map image in L*a*b* [4] color space is illustrated. One can see that the distribution does not contain clear centroids. For example, the water pixels, which are supposed to be grouped around a dominant blue color, are actually a mix of blue and white clouds due to the dithering effects. The same holds for the yellow fields. For the whole image this effect makes the distribution uniformly spread.

We consider map imagery as a class of images with distinctive properties separating them from photographic, computer generated or other classes of images. Digital map images (both scanned and converted) are widely used among a great variety of users worldwide. However, general purpose algorithms rarely take the properties of this kind of imagery into account. The work in this thesis is motivated by the fact that better understanding of the properties of map images together with designing and optimization of algorithms exploiting these features can make map image processing and compression algorithms more efficient.

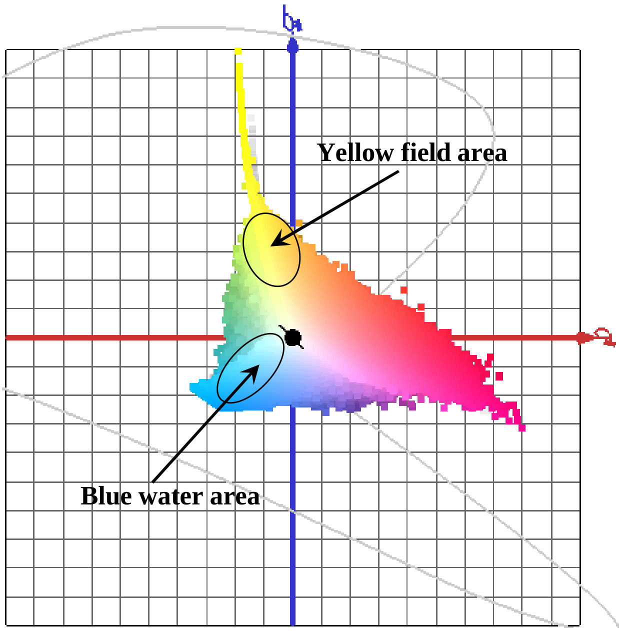


Figure 3. Distribution of colors in a sample scanned map image in L*a*b* color space

This thesis is aimed at two specific topics. The first topic of the thesis is layer-wise map image processing for reconstruction and compression. In paper **P1**, lossless compression is improved by trying to reconstruct the semantic layers of the map. In paper **P2**, bit plane separation and binary context-based compression of natural and palette images are studied.

The second topic is dedicated to context tree (CT) modeling. In paper **P3**, we apply highly optimized CT modeling for the progressive lossy-to-lossless compression of map images. In paper **P4**, we propose a CT filter for improving the quality of noisy map images. In paper **P5**, we generalize the method for lossy compression of scanned maps.

2 Image compression

Compression algorithms can be separated into two classes: *lossless* and *lossy* algorithms [5][6][7]. Lossless compression assumes that the data before and after the compression-decompression process is equal, *i.e.* no loss of information occurs. In contrast, lossy compression makes no such assumption, and allows distortion to happen. This is essential in those situations where some degradation of the data is tolerable for the benefit of better compression efficiency. The algorithms of the first type (lossless) are used in applications where information loss is not acceptable, *e.g.* compression of text, programs and executable code. In image compression, lossless compression can be used for compression of medical images, engineering drawings and circuits. The algorithms of the second type (lossy) are applied in photographic image, audio and video compression because minor degradation can be tolerated if it is visually not perceptible, and because lossless methods alone are inefficient for this type of data.

2.1 Lossless compression algorithms

Images as a class can be of very different natures, structures and contents. Therefore, any successful compression technique is usually adapted to be applied on a particular type of images. Lossless image compression algorithms can be organized into three groups: *continuous-tone*, *discrete-tone* and *universal* algorithms. The compression algorithms referred as continuous-tone are optimized to perform on *natural imagery*, usually photographic or other types of images obtained with a physical sensor. Discrete-tone algorithms are designed to perform on other types of images that contain fewer colors and less gradation, with more sharp edges and uniform areas. Images of this type are mostly of an artificial nature such as web graphics, engineering drawings, maps and circuits. Universal compression algorithms are usually applied when the type of the data is not predefined.

Popular universal compression techniques are based on various adaptations of a classical dictionary-based LZ77 or LZ78 [13][14] algorithm. For example, CompuServe *Graphics Interchange Format* (GIF) [8], which is widely used for the compression of palette images, uses LZC [9] improvement of LZW [10]. The *Portable Network Graphics* (PNG) algorithm [11], which was proposed as the replacement for the relatively old GIF uses *DEFLATE* [12] algorithm. It uses a combination of LZ77 [13] and *Huffman coding* [15]. The *ITU Group 4* algorithm [16] incorporated in *Tagged Image File Format* (TIFF) [17] uses simple data compression techniques based on run-length coding, prefix coding and differential *relative address designate* (READ) coding to utilize line to line correlations.

However, universal compression algorithms suffer from the one-dimensional nature of the method, and thus present relatively low compression efficiency.

A natural way to increase the efficiency of the compression algorithm is to optimize the compressor for the particular class of images. This approach was realized in *Joint Bi-Level Image Experts Group* (JBIG) [18], which is an algorithm optimized for bi-level images containing pixels of two types: background and foreground. As originally proposed in [21][22], the algorithm is based on local probability estimation via context modeling followed by an arithmetic coding [19] performed by so called Q-Coder [20]. The JBIG standard was then expanded by JBIG2 [23][24].

Popular examples of lossless continuous-tone compressors are CALIC and JPEG-LS. *Context-based adaptive lossless image compression* (CALIC) [25][26] is based on *gradient-adjusted prediction* (GAP), which is adjusted via an error feedback loop. The residue of the predictor is entropy-coded based on eight estimated conditional probabilities in eight different contexts. JPEG-LS [27] is based on *Low Complexity Lossless Compression for Images* (LOCO-I) [28], which is also based on context-adaptive prediction and adaptive *Golomb-Rice coding* [29][30].

Discrete-tone images are of a different nature and prediction-based techniques usually fail to present high compression efficiency. Discrete-tone oriented compressors exploit different ways of removing the redundancy. For example, *Piecewise-Constant Image Model* (PWC) compression algorithm [31] uses a two pass model to capture the characteristics of a discrete-tone image. During the first pass, boundaries between constant color areas are detected. The second pass then determines the color of the area. Encoding is performed in an object-oriented manner using the so called *PWC language* consisting of four decision possibilities.

Embedded Image-Domain Adaptive Compression (EIDAC) [32] compresses an input image as a sequence of bit planes starting from the most significant to the least significant bits allowing a progressive transmission. Coding is performed by inter-layer context modeling and an adaptive binary arithmetic coder providing high compression efficiency.

2.2 Context modeling

Context modeling is a well known tool which is widely used in image compression [22]. The main idea is to estimate the probability distribution of symbols in the input data using the knowledge of the context in which the unknown symbol appears. This concept is effective when there are statistical dependencies in the data, which holds true for most of the images. Usually, the encoder gathers the statistics of pixel occurrences taking the configuration of the already processed neighboring pixels into account. Using this information, variable length code words are assigned to pixels so that shorter codes are assigned to more probable pixels and vice versa.

We refer to a configuration of positions of neighboring pixels as a *context template*, *i.e.* the context template defines the shape of the neighborhood to be examined. The choice of the context template is essential for the compression performance. We refer a configuration of pixel values in the neighborhood as a *context*. The principles of context-based probability estimation are easy to illustrate for a binary case when only back- or foreground pixels are possible in the image. Sample 10-pixel context template used in JBIG compression standard [18], and sample contexts for a binary and four-color image together with the corresponding probabilities are shown in Figure 4.

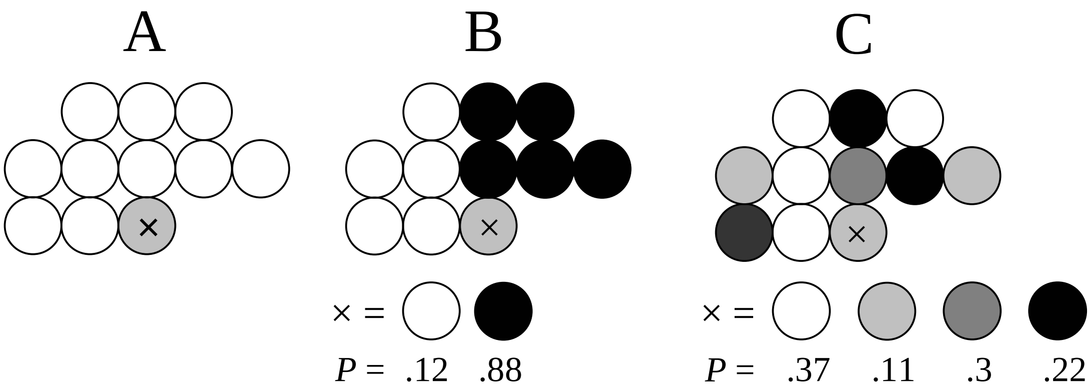


Figure 4. A sample 10-pixel contest template (left); a binary 10-pixel context with the corresponding probability distribution (middle); a four-color 10-pixel context (right) with the corresponding probability distribution.

When using large context modeling, its extensive memory consumption is a major problem. With any increase of the context template size the number of possible contexts grows exponentially; for an N-pixel context there can be 2^N contexts in total. In a binary case, one must keep two counters for each context to track the probability distribution. Though this number is reasonable for $N \approx 10$, any further increase of the model size is problematic. For K number of colors is possible in the image, the number of possible contexts raises to K^N making the storage of K counters for each context impossible.

Another problem with context modeling is *context dilution* [66]. Usually, a bigger context allows more accurate estimation of the probabilities. However, at some point the improvement will stop and further increase becomes counter productive. Since the image is restricted by size, bigger contexts tend not to appear frequently enough to make an accurate estimate of the probability. Wrong estimation causes deterioration of the compression efficiency.

Both problems are usually overcome by considering *context tree* (CT) modeling [22]. The idea behind CT is that although the number of possible contexts is huge, the number of contexts actually appearing in an image is upper limited by the size of the image. Therefore, if memory is allocated only for really appearing contexts, the reliable estimation of the pixel probabilities becomes possible.

2.3 Context tree modeling

The storage of pixel counters in CT is organized in a tree structure, see Figure 5. The nodes of the tree represent the contexts appearing in the image. Symbol '×' denotes the unknown pixel within the particular context. The statistics *i.e.* the counters represent how many times the unknown pixel appeared as a particular color. In case of binary image, two counters are needed: N_W represents the number of white pixels which appeared, and N_B represents the number of black pixels.

The positions of the context pixels in the context template are arranged in a predefined order; that is, the construction of the tree starts from the root. For every pixel of the image, the algorithm sequentially examines its neighbors according to the defined context template. If the first pixel of the template appears as white, the transition to the left child node is made; otherwise the right transition is made. The process continues recursively: the algorithm examines the second neighbor position in a template and makes the transition to the next level of the tree corresponding to the value of the next context pixel. With every transition from node to node, the algorithm updates the pixel counters according to the value of the current (unknown pixel). In cases where the current transition does not exist, the necessary node of the tree is created dynamically. Accordingly, only nodes corresponding to existing contexts are created in the tree and no memory is wasted for non-existing pixel combinations.

In order to prevent context dilution problem the size of the model used must be restricted. In context modeling this is usually done by using a *context quantization* approach [67], which in context tree modeling is usually referred as *tree pruning*. The simplest way is to require that every node has children only if the code size provided by the children is less than the one provided by their parent. This guarantees that all surplus nodes of the tree will be pruned and the undesired increase of the context model will be prevented. However, this greedy-style approach does not provide the optimal performance and more sophisticated pruning algorithms can be found in literature [68][72][73].

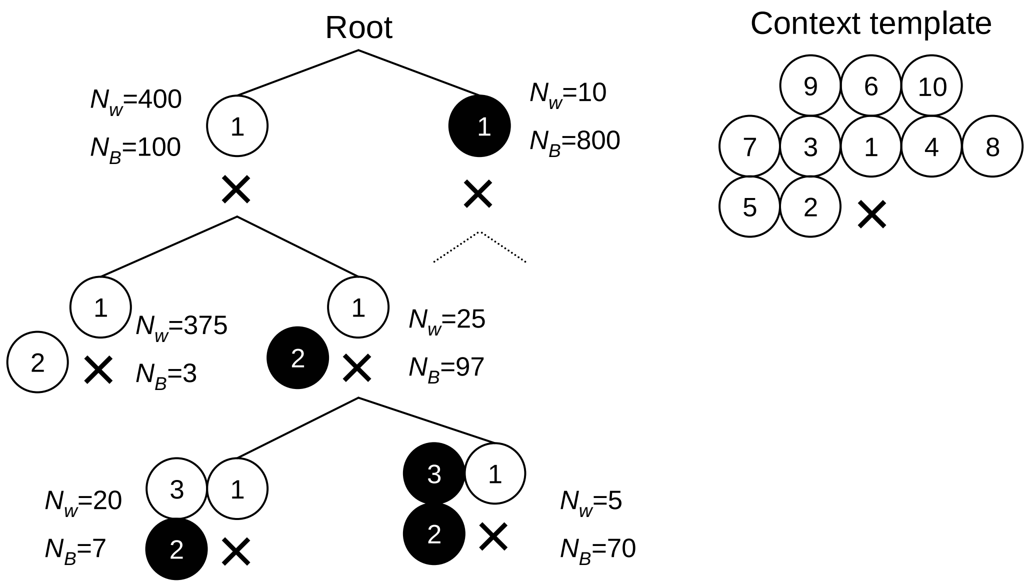


Figure 5. Construction of binary context tree

Although context tree modeling is not restricted to binary images, the usage of a more general approach when there are more than two colors are possible is restricted by the resource allocation problems. The principle of constructing a *generalized context tree* (GCT) as proposed in [72] is illustrated in Figure 6. In contrast to the binary case, the nodes of the tree have more than two children. Potentially there are as many children as there are colors in the image. Every node must track the appearance counters for all possible colors. Pruning is also more complicated in the generalized case. For K children, there are 2^K pruning configurations and each can provide different code size. The selection of the optimal combination by a full search is extremely slow and therefore impractical. In GCT, sub-optimal pruning by a *steepest descent search* algorithm was considered for solving the pruning problem in a reasonable time, and still providing performance close to the optimum.

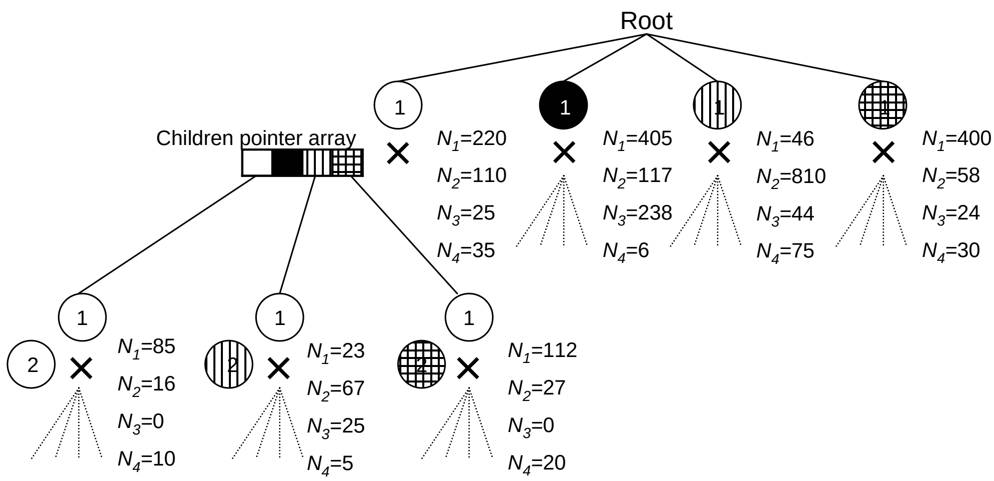


Figure 6. Construction of the generalized context tree

The order of pixel positions in a context template is also essential and can be a subject of optimization. A solution for the CT model is called *Free Tree* [73] and it has been considered both for binary images [70] and for gray-scale [72]. It provides better compression performance in comparing to a static order CT [68]. Sample contexts optimized by free tree are illustrated in Figure 7

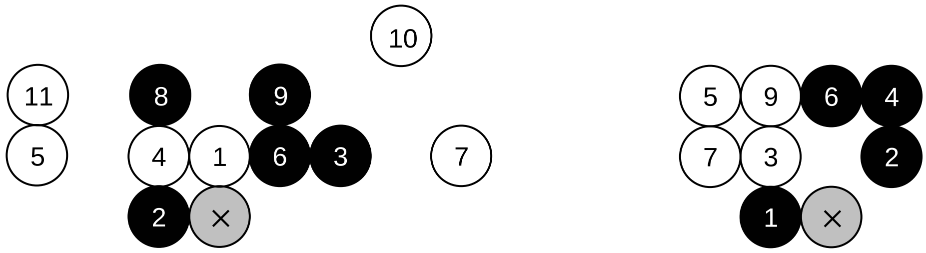


Figure 7. Two sample free tree contexts.

2.4 Lossy compression algorithms

Most popular lossy algorithms are used for compression of photographic imagery since the nature of the human eye's perception allows significant reduction of information in the image without any subjective loss of quality. However, in some applications the properties of the input imagery can significantly differ from natural photography, thus requiring different compression principles to be applied.

2.4.1 Existing methods

The classical examples of popular lossy compression algorithms are *Joint Photographic Expert Group* (JPEG) [33] and a more recent standard JPEG2000 [34]. These algorithms are based on image transforms: *discrete cosine transform* (DCT) [35] for JPEG and *wavelet transform* [36] for JPEG2000. The transform coefficients are rounded and quantized causing partial loss of information. These algorithms are optimized for compression of photographic images, which are mostly used in computer industry. There are also transform-based algorithms optimized for different tasks such as *Enhanced Compression Wavelets* (ECW) [37] and *Multiresolution Seamless Image Database* (MrSID) [38]. These are commercial solutions for the compression of aerial and satellite photos. DCT and especially wavelet based algorithms present excellent compression efficiency in terms of compression *vs.* degradation tradeoff for the class of images which they were optimized to.

DjVu [39] algorithm was proposed for lossy compression of scanned imagery containing text and line drawings, especially scanned books. The algorithm utilizes the fact that scanned images of that type contain a lot of sharp edges and details, which are difficult to represent by DCT or wavelets. The algorithm therefore separates the image into two parts: text and background, and applies different compressors for each. Binary context-based algorithm JB2 is a variant of variant of the JBIG2 [23] standard and is applied for text. The low resolution wavelet-based IW44 is proposed to compress the background.

Lossy predictive coding is also used for the so called *near-lossless compression* when the degree of imposed degradation is limited. Lossy predictive coding assumes that the prediction error is not encoded precisely but quantized, thereby causing minor errors when the image sample is reconstructed. This technique is used in JPEG-LS [27] near-lossless mode, for example.

Quantization of signal can also be seen as an approach of lossy compression [40]. Reducing the number of unique colors (or gray scale gradations) in the image imposes distortion, and at the same time, reduces the informational content of the image, thus improving its compressibility. For example, the GIF standard operates only on indexed palette images requiring quantizing colors to a predefined palette (typically 256-color) before the compression. The impact of quantization on compression efficiency has been studied in several papers [41][42][43].

2.4.2 Lossy-to-Lossless approach

In some applications, it is not necessary to transfer the whole image data in one continuous transmission. It is often more important to have a schematic thumbnail of the image faster than the whole image. This requirement is typical for browsing and retrieval applications in restricted bandwidth transmitting channels, when one must decide whether the acquired image is relevant to the query.

This task is usually solved by designing the compression algorithm in a way allowing *lossy-to-lossless (progressive) decompression* [34][44]. The image is decompressed step-by-step so that the most important part of the information is decompressed first. Each step then updates the data finally giving the exact lossless reproduction of the image. An importance criterion is usually defined by minimizing the *mean squared error* (MSE) distance from the partially reconstructed image to the original. An example of progressive reconstruction is given in Figure 8 where JPEG quality progression is illustrated. Progressive decompression is a popular feature of existing compression standards such as JBIG [18], JPEG [33], JPEG2000 [34], GIF [8] and PNG [11].

Figure 8. Sample JPEG quality progression.

2.4.3 Lossy compression by color quantization and GCT modeling

The class of scanned raster map images is commonly used in navigational applications in cases when vector map is not available. Scanned map image combines the properties of two classes: natural and artificial imagery. They originally contain only a few colors, sharp edges and small details. After the scanning, however, this image is corrupted by noise caused by the acquisition sensor imposing blurring and other inconsistencies. Therefore, neither traditional lossy image compression algorithms like JPEG and JPEG2000, nor lossless image compression techniques like PNG are well suited for scanned maps.

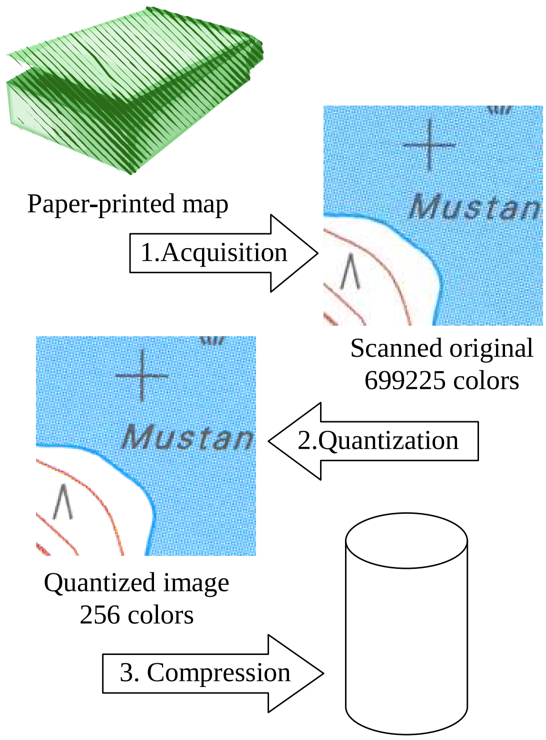


Figure 9. Overall scheme of the proposed lossy compression technique.

In **P5,** we propose an algorithm for lossy map image compression based on *Median Cut* [71] color quantization and generalized context tree modeling (GCT) [72]; see Figure 9 for the overall scheme. Samples representing the quality provided by the proposed technique are presented in Figure 10. The upper row represents 0.72 bit per pixel compression results, and the artifacts and blurring imposed by JPEG2000 along the edges are clearly seen. The corresponding image provided by the proposed algorithm is free from these artifacts. The lower row represents a higher quality level for the proposed technique using 256 colors. Although any difference with JPEG2000 is hardly visible, the objective measurement shows an advantage of the proposed algorithm. In general, when comparing images at a similar objective quality level the proposed algorithm provides up to 50% better compression efficiency than JPEG2000.

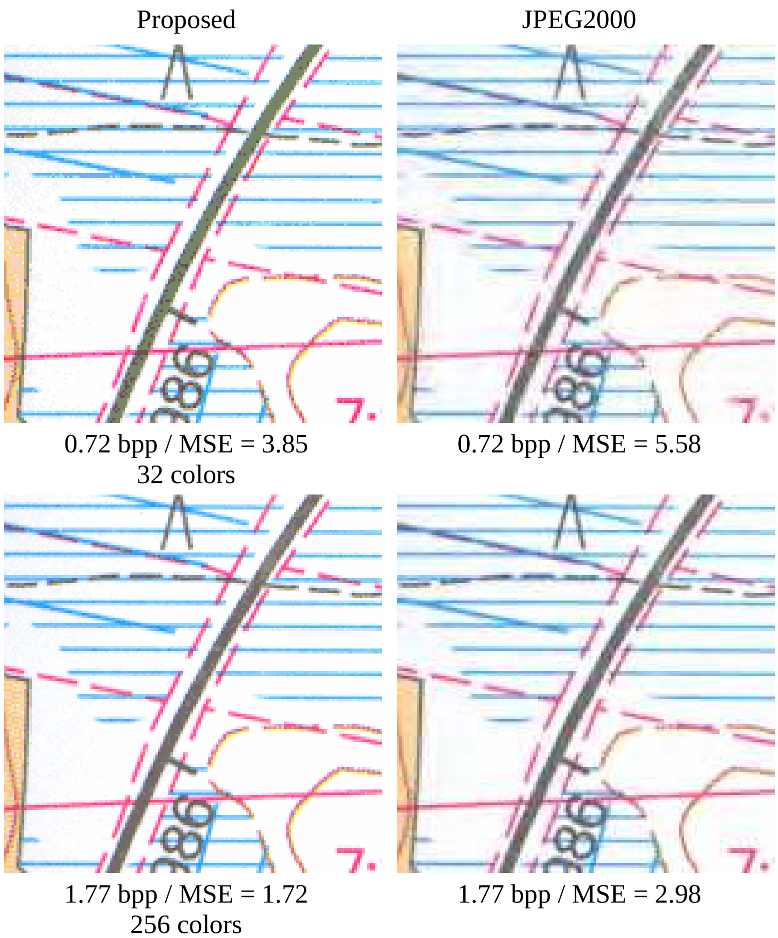


Figure 10. Visual comparison of JPEG2000 and the proposed lossy compression algorithms.

3 Image filtering

Image filtering aims at reconstructing the original image before degradation [45][46][47][48]. As a rule, the reconstruction involves a criterion for measuring the quality of the desired result. There are two principally different approaches for the quality measurement: *objective* and *subjective*. Objective quality measurement assumes that it is possible to establish an objective metric. The most common examples of these metrics are MSE and *peak signal-to-noise ratio* (PSNR). The objective measurement measures a ‘distance’ between the original image and the result of reconstruction. This is possible when the original image is available for measurement, which is not always the case.

Another approach is subjective quality estimation. In the case when the uncorrupted image is not available, one can estimate the restoration quality by subjective observations of the reconstructed image. This approach is less analytical than the first one and, therefore, less popular. Besides the two above mentioned approaches, different performance evaluation methods can be defined. For example, in **P1** we use image compressibility as a quality evaluation criterion.

3.1 Existing algorithms

Linear filtering is an approach widely used since the beginning of the computer era. The filter replaces a pixel with a linear combination of its neighbors combining the simplicity of implementation with robustness to various tasks from smoothing to edge detection. Linear filters, however, are not well suited for filtering of map images since the imposed smoothing is not (always) tolerable. Linear filters homogeneously process all pixels, which is another drawback for a filtering of images consisting of complex structures.

Later, a great variety of more general *non-linear filtering* algorithms were considered. In the early sixties, the investigations of Matheron and Serra led to a new quantitative approach in image analysis, now known as *mathematical morphology* [49][50][51]. The central idea of mathematical morphology is to examine the geometrical structure of an image by matching it with small patterns at various locations in the image. By varying the size and the shape of the matching pattern, called *structuring element*, one can obtain useful information about the shape of the different parts of the image and their interrelations. Flexibility of the concept allows various filters to be designed [52][53][54][55]. Mathematical morphology is widely applied in various disciplines such as mineralogy, medical diagnostics, machine vision, pattern recognition, granulometry and others [56].

There exists a great variety of heuristical filtering approaches, which exploit knowledge about the noise. For example, edge preserving filters are trying to smooth uniform areas while keeping the edges untouched. One of the most popular edge preserving filtering methods is *vector median filter* (VM) [57], which is a non-linear operator. The filter replaces the current pixel value with a value called *vector median* defined in a local neighborhood. An attempt to design a filter that would be invariant to the features of the particular image was made in [58]. The filter is called *rank-conditioned vector median filter* or *adaptive vector median filter* (AVM), and it uses noise detector before applying VM. An overview of weighted median filters can be found in [59].

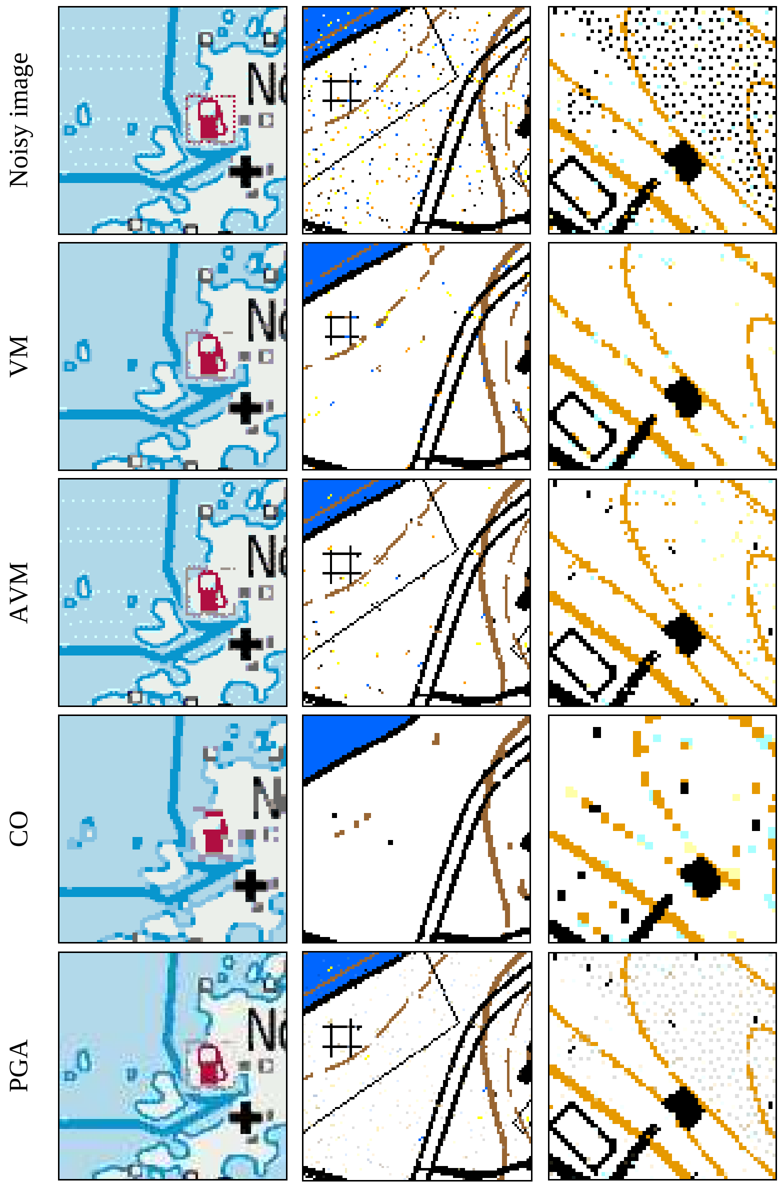


Figure 11. Application of vector median (VM), adaptive vector median (AVM), morphological Close-Open (CO) and peer-group analysis filters to sample map images.

Classical *Kuwahara filter* [60] examines the neighborhood for a sub-region with the smallest variance and replaces the current pixel with the mean of the region. A similar approach is used by *peer group analysis* (PGA) [61], which is an edge-preserving smoothing technique based on finding a group of pixels similar to the current one in a local neighborhood. When such a group is found, the current pixel is replaced with the average of the group. Statistical non-linear filters use local probability estimation for noise detection and correction. A *gain-loss filter* was proposed in [62] for improving the compression of binary images. Various vector-based filters are discussed in [63] and [64]. Application of selected VM, AVM, morphological *Close-Open* (CO), and PGA filters is illustrated in Figure 11.

3.2 GCT filtering

Map images are typically highly structured. The patterns are usually clearly defined and commonly repeated in the image. This makes context tree modelling an effective tool for statistical analysis and processing. Consider a map image which is slightly corrupted by impulsive or content-dependent noise. By *content-dependent* noise, we assume that the corruption occurs at the borders of the objects. The presence of noise corrupts the statistical consistency of the image and, therefore, statistical analysis is an appropriate tool for noise detection and removal.

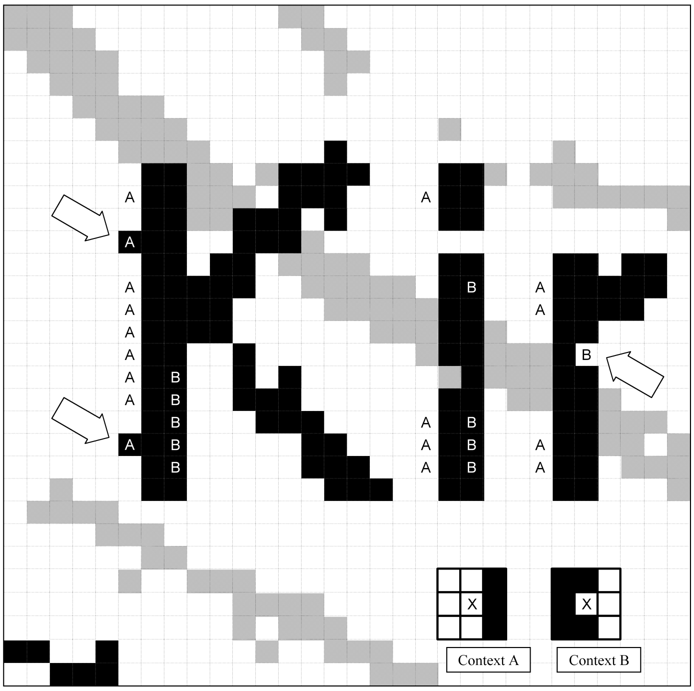


Figure 12. Principles of context-based filter.

In **P4,** we propose a statistical context tree based filter for map images basing on the preliminary works published earlier [62][65]. The filter analyzes statistical distribution of the colors within a local neighborhood using a generalized context tree model. Pixels are considered as noisy if their conditional probability falls below a predefined threshold. The size of the neighborhood is dynamically adapted via using a tree pruning technique. The principle of the algorithm is illustrated in Figure 12, where two 3×3 contexts A and B are presented. One can see that black is much less probable than white in the context A, and vice versa; white is less probable than black in the context B. By replacing noisy pixels by the most probable ones, the filter is able to reconstruct the initial structure of the image. The proposed filtering is very sensitive to the original structure of the image and the amount of the corruption imposed is rather small. Sample corrupted and reconstructed images are presented in Figure 13.

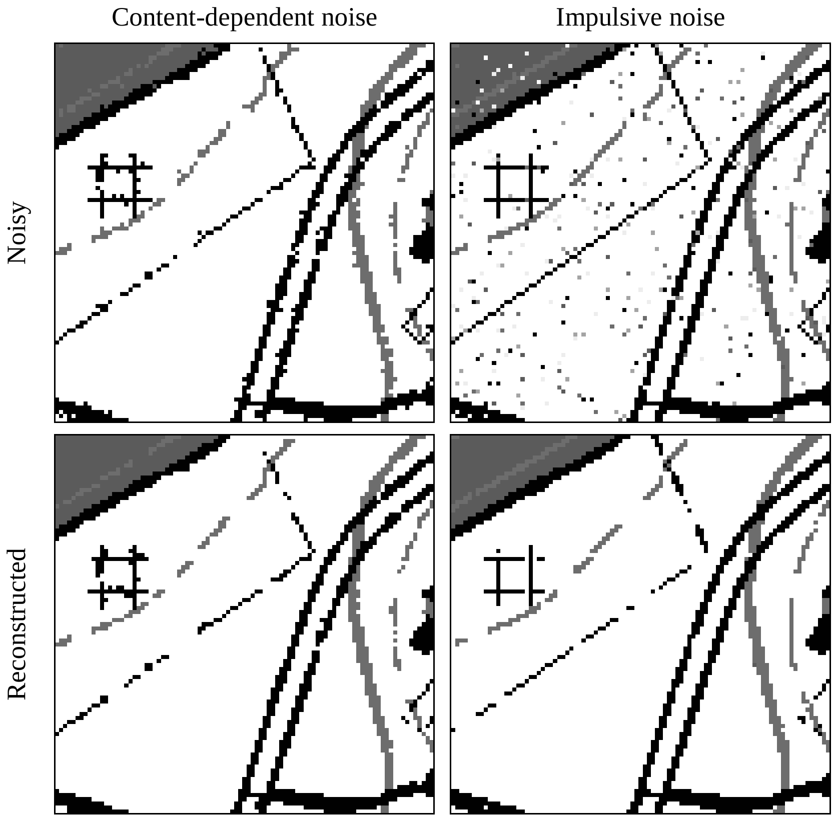


Figure 13. Sample noisy and reconstructed images.

4 Layerwise processing

There is a class of images consisting of a set of binary *layers*. Maps are a typical example of this kind of images since it consists of *semantic layers*: binary images representing geographical objects of similar nature and depicted with a particular color. For example forests are depicted in green and one can extract all green regions of the image into a binary image (a forest layer); see an example of layered image in Figure 14. In an atlas map, colors can represent a great variety of parameters such as density of population, pollution and temperature. Bit plane separation is another example of the layer decomposition approach. Gray-scale image can be decomposed into a set of binary layers according to the bits of each pixel value.

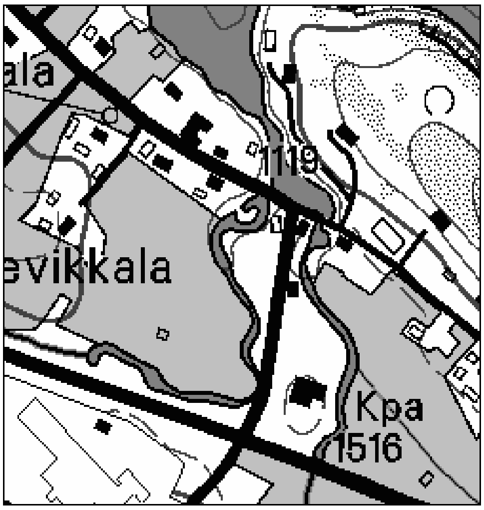


Figure 14. Illustration of a multilayer map image from the NLS topographic database [3].

If there is a strong correlation between layers, it can be utilized to improve the performance of the compression, filtering, or other processing algorithms. This correlation certainly exists in map images between their semantic layers [69]. In [70], inter-layer correlation was used to improve the performance of a two-layer context-based lossless compressor. This motivates us to research layerwise processing of map and gray scale images for improving the performance of lossless compression algorithms.

4.1 Morphological reconstruction of semantic layers

When producing a raster map image, map layers of different semantic nature are combined together overlapping each other in a predefined order. This image is well suited for user observation but less appropriate for further processing since the layer structure has been corrupted when the raster map image was produced. The problem is that the overlapping introduces severe artifacts in places where the information on different layers overlap each other; see Figure 15, upper row. The holes on the face of the lake left by the overlapping letters are typical examples of the artifacts. The presence of these artifacts degrades the compressibility of the color map image, in comparison to the situation when the original semantic layers were available.

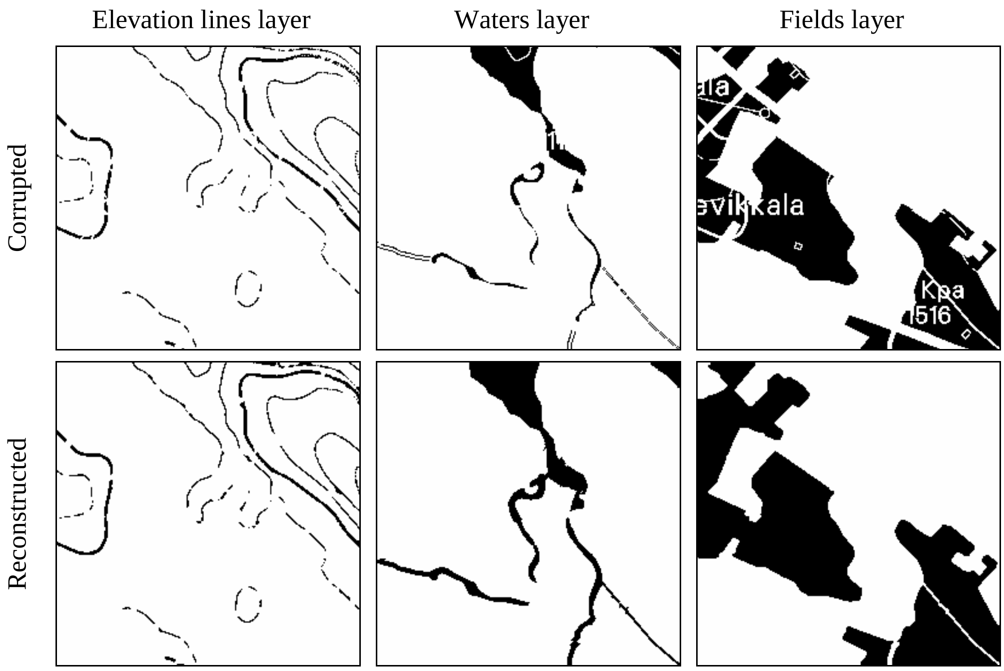


Figure 15. Semantic map layers: corrupted layers due to the color separation (upper row); reconstructed with the proposed algorithm (lower row).

This led us to develop an algorithm for reconstruction of the corrupted layers of map images. The algorithm proposed in **P1** approximates the original layer structure existing before the color combination by repairing the corrupted layers as close to the original ones as possible. Since the converted raster map images are usually compressed by a lossless algorithm, we require that the color combination of the reconstructed layers must be equal to the originally received raster map image.

The results of the proposed reconstruction technique are presented in Figure 15. The removal of overlapping artifacts provides 30-50% better compression on standalone layers, and 5-10% better compressibility for 4-layer map images without any loss of quality. Besides that, the proposed technique can be used for the removal of unnecessary layers from the map.

4.2 Compression of gray scale images

Other types of images can also be treated as layered images via the use of bit plane separation. This assigns one bit from the binary representation of the pixel value into each bi-level layer, thus losslessly separating any gray scale image into eight layers.

The overall scheme of this approach is shown in Figure 16. In the case where there is a correlation between the bit layers, it is possible to utilize this to gain better compression efficiency. For example, in context modeling, involving neighboring pixels from already processed binary images can improve the probability estimation and, therefore, the compression. Among the existing implementations we can mention EIDAC [32] lossless compression algorithm, which uses a binary multi-layer context model that operates on bit-layers of the image using both the actual bit values and their differential characteristics as context information. Two-layer context modeling with optimization of the order of layer processing was considered in [70].

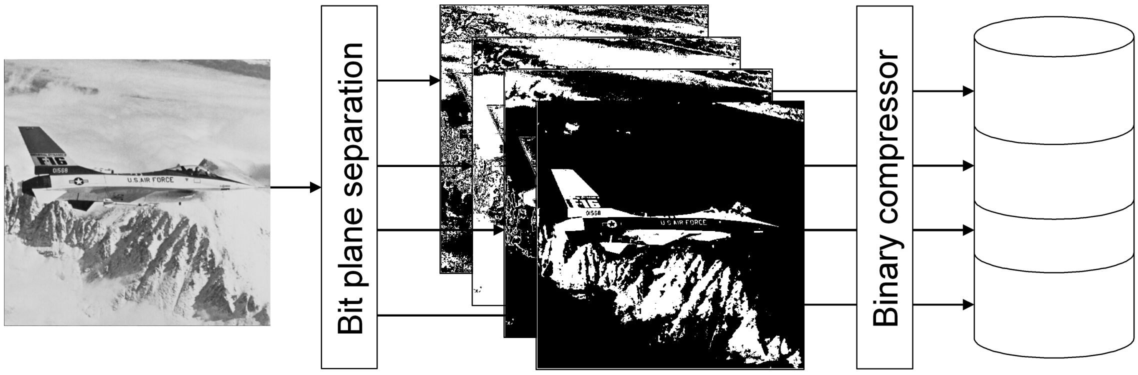


Figure 16. The overall scheme of bit-plane-based compression.

In **P2,** we study how well the bit-plane-based approach can work on natural and palette images. We consider four different bit plane separation schemes: straightforward bit plane separation, Gray-coded bit plane separation, bit plane separation of prediction errors and separation of Gray coded prediction errors. We use the highly optimized MCT context modeling method for lossless compression and, furthermore, extend the two-layer MCT model to a multi-layer context model for better utilization of cross-layer dependencies. In general, any previously compressed layer can be used to provide the contextual information for the next layer being compressed. An example of a multi-layer neighborhood used in **P2** is presented in Figure 17. We extensively evaluate the proposed combinations of the different bit plane separation and context modeling schemes, by applying them to natural and palette images. The efficiency of the bit-plane-based compression is compared to the existing compressors. Moreover, the dependency of the compression on the image content is studied by modeling the transition between natural and palette image classes.

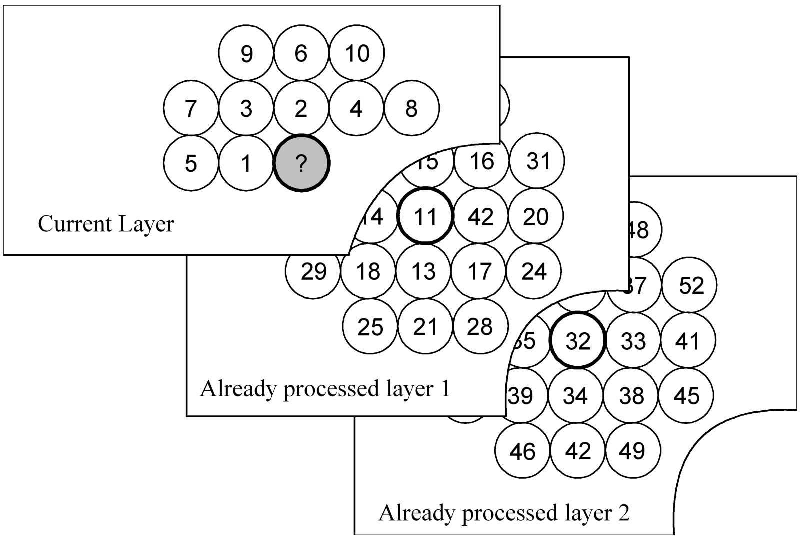


Figure 17. A sample multi-layer context template.

4.3 Progressive compression via binary layers

Binary context modeling is also used for progressive encoding of color quantized images in [74]. The algorithm uses binary tree representation of the color palette following by a progressive binary context-based encoding. In [75], an improved version has been proposed. In **P3,** we continue the development of this approach by improving the quality of the color progression by using merge-based color clustering [76] instead of the original splitting-based approach. We also propose the use of binary free tree modeling instead of the static context model. The proposed improvements provide better subjective quality of the color progression (see Figure 18), and 10-20% better compression performance for the set of palette images.

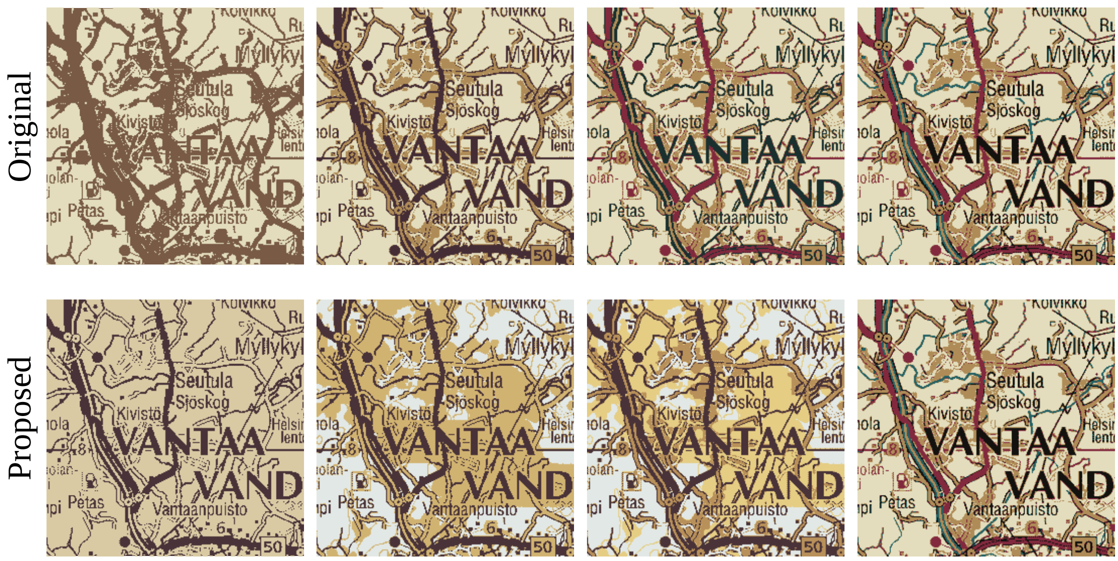


Figure 18. Color progressions provided by the original and proposed algorithms.

5 Summary of the publications

In the first paper (P1), we propose a technique for reconstruction of binary semantic layers of map images from the corruption imposed by overlapping of color layers in the map. Separation of the map into color layers and compressing them individually provides better compression performance than using standard techniques. However, color separation causes artifacts in areas where one layer overlaps another. The proposed algorithm approximates the original structure of the layer existing before the overlap by a sequential application of masked morphological operations. The image is processed so that the color map itself remains untouched, and only the underlying binary layers are modified. The proposed technique obtains up to 30-50% compression improvement for single layers, and improves the compression ratio of the whole map up to 5-10%.

In the second paper (P2), we explore the efficiency of binary-oriented compression algorithms applied to gray-scale and palette images. In contrast to map images, gray-scale imagery contains much more gradation and it is more difficult to exploit spatial dependencies via binary layers and color separation as in **P1** is not possible since it would lead into too many (256) layers. In this work, we consider four different bit-plane separation schemes using error prediction and Gray-coding. For prediction-based schemes we evaluate three different predictors. Bit-plane separation schemes are combined with two binary-oriented compressors: one known (MCT) and one novel referred as *N-layer Context Tree* (NCT).

We evaluate the proposed variants on natural and palette images. The variants providing the best compression are compared with six existing compression algorithms. We conclude that despite the high order optimization a binary-oriented compressor cannot outperform the best lossless gray-scale oriented algorithms due to the nature of the signal. For palette images we found out that highly optimized binary compression is able to provide compression performance close to the best existing compressors but at the cost of higher processing time.

In the third paper (P3), we improve a recently proposed layerwise lossless compression algorithm, which is based on binary tree representation of the colors and on context-based arithmetic coding. We considered two improvements for the algorithm: merge-based color quantization instead of the original split-based strategy, and a context tree modeling optimized for each layer separately. The improved algorithm is evaluated on natural and palette images. The proposed method provides better subjective quality of the color progression, and compression improvement of 12% in the case of color palette images.

In the fourth paper (P4), we propose a statistical filter for map images based on local probability estimation using context tree. The estimation is followed by replacing less probable pixels by the most probable one. The size of the context is dynamically adjusted according to the proposed tree pruning procedure. The main feature of the proposed filter is that the use of the context tree allows investigating larger neighborhoods with reasonable time and memory consumption. The filter effectively reconstructs patterns of the image in the presence of moderate impulsive and content-dependent noise.

In the fifth paper (P5), we consider a novel lossy compression scheme for scanned map images. The proposed compression algorithm consists of two stages. First the number of colors in the original image is reduced by color quantization. The quantized image is then compressed with the lossless GCT compression algorithm. In this work, two improvements for the original GCT were considered: a fast pre-pruning method and an optimized memory allocation. Both improvements reduce the memory consumption as well as the processing time of the algorithm significantly. The proposed compression scheme is evaluated on a set of scanned topographic maps. The evaluation shows that the algorithm provides a compression improvement of about 50% in comparison to the closest competitor, JPEG2000, at the similar objective quality level.

In paper **P1**, the author developed the principles of the algorithm, implemented and evaluated it. The other two authors took part in the problem formulation and editing of the article. In paper **P2**, the author implemented the N-layer context tree modeling, bit-plane separation schemes, predictors and performed all the experiments. In paper **P3**, the author implemented and tested the improved compression algorithm. Paper **P4** is based on a preliminary version published in a conference by the second and third authors. The contribution of the author includes the tree pruning procedure, a new implementation of the filter with significantly better memory consumption, performing new experiments and a broader evaluation. In paper **P5**, the author considered and implemented the improved GCT compressor and performed the experiments.

6 Conclusions

In this thesis, we have studied lossless and lossy compression of raster map images as well as layerwise processing algorithms for their improvement.

We have proposed a morphological algorithm for restoration of binary semantic layers of multi-layer map images from the corruption appearing in areas where semantic layers overlap each other. The proposed reconstruction allows improving the lossless compression of the layers up to 30-50% for standalone layers and in the total compression rate up to 5 to 10%, depending on the compression method applied.

We have studied the efficiency of highly-optimized binary-oriented compression algorithms to examine if their high performance presented for maps is possible to utilize for grayscale and palette images. We consider a set of binary layer separation schemes. Besides that we consider two schemes for context modeling: one existing and one novel (NCT). The experiments show that statistical context modeling and arithmetic coding cannot outperform the best grayscale-oriented compressors. On the other hand, when applied to artificial palette-like imagery, the optimization of the model results in a compression performance which is close to the best existing algorithms and further improvement is possible.

We have proposed a novel filter for reconstruction of map images in presence of noise. In contrary to the existing edge-preserving filters designed to preserve areas of high color variation, our filter aims at preserving the repetitive structures of the image which is an essential property for raster map images. The problems of the appropriate context size and resource allocation are solved. The proposed filter outperforms edge-preserving competitors both in objective and subjective comparisons.

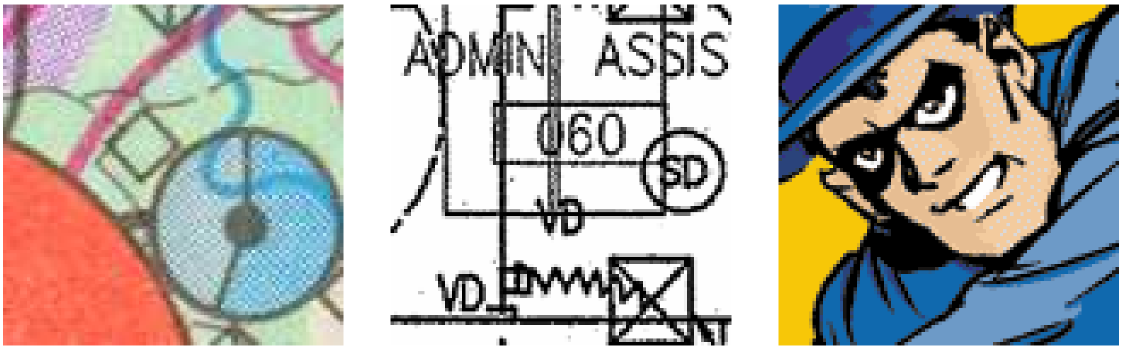
We have improved a recently proposed lossy-to-lossless compression algorithm based on layerwise progressive binary compression. The improved algorithm provids better visual quality of the lossy progression; 12% better compression is achieved for palette images.

We have proposed a novel scheme for lossy compression of scanned map images utilizing the common features of the map imagery. The novel scheme provides up to 50% better compression at the same quality level when compared to its closest competitor JPEG2000.

7 Future work

We believe that this thesis can be used as a basis for further research. In **P5** a pioneer work in lossy context-based compression has been done. Although the algorithm is applied on scanned topographic maps only, we expect the results to generalize to similar image classes, such as other types of maps, engineering drawings, schemes, comic books and similar art imagery. Of this kind images have properties similar to map imagery, see Figure 19. A similar lossy algorithm DjVu can be developed where background and textual information would be separated, and the textual part compressed by a GCT-based encoder, which is not restricted to work only for bi-level images as DjVu.

The potential of GCT-based compression is possible to extend to video compression. The method is expected to be efficient for video where features of the imagery are close to the ones illustrated in Figure 19, such as high quality cartoons and animation, see Figure 20. Since subsequent frames of the video are highly correlated, multi-layer GCT modeling is expected to be a very efficient compression tool.



Atlas | Engineering drawing | Comic book

Figure 19. Sample images with features similar to map imagery.

Anime frame 1 | Anime frame 2

Figure 20. Two consequent anime frames.

8 Summary of the results

8.1 Publication 1

Table 1. Compression of topographic map images with different compressors. The average results (size, bits per pixel and compression improvement) presented for original, corrupted with color separation and reconstructed layers.

Compression algorithm	Original		Corrupted		Proposed reconstruction		
	Size	bpp	Size	bpp	Size	bpp	imp.
PNG	2 085 871	0.66	2 149 490	0.68	2 078 254	0.66	3.31%
TIFF	1 473 824	0.47	1 708 362	0.54	1 480 657	0.47	13.33%
JBIG	684 978	0.21	790 257	0.25	720 185	0.23	8.87%
AKF2	624 117	0.19	696 017	0.22	660 661	0.21	5.08%

8.2 Publication 2

Table 2. Compression results (bits per pixel) for the natural images.

Image	Proposed	Competitive							
	MCT-GCPES	JBIG-BPS	JBIG-GCS	CALIC	JPEG-LS	PWC-G	PWC-P	JPEG2K	PNG
Average	4.42	5.64	4.75	**4.11**	4.18	4.21	4.84	4.36	4.56

Table 3. Compression results (in bytes) for the palette images.

Image	Proposed	Competitive							
	NCT-BPS	JBIG-BPS	JBIG-GCS	EIDAC	CALIC	JPEG-LS	PWC-G	PWC-P	PNG
Total	175590	351913	211943	**140957**	226296	272555	198931	144344	245745

8.3 Publication 3

Table 4. Compression results of PNG, Chen's, Pinho's and the proposed algorithm as well as obtained compression improvement (comparing to the closest competitor) for natural and palette test sets.

Test set	PNG	Chen *et al.*	Pinho *et al.*	Proposed	Improvement
Natural	7261542	2521448	2426446	2399451	1%
Palette	712726	274700	257126	226469	12%

8.4 Publication 4

Table 3. The efficiency of mathematical morphology (MM), vector median (VM), adaptive vector median (AVM) and the proposed (CT) filters measured as Δ*E* distance to the original image for 20% content-dependent (CD) and 5% impulsive noise (I).

	Image 1		Image 2		Image 3		Image 4		Image 5		Image 6	
	CD	I	CD	I	CD	I	CD	I	CD	I	CD	I
MM	23.52	24.37	29.66	30.28	27.75	28.33	14.10	14.48	4.54	8.68	30.45	31.11
VM	3.16	2.51	8.50	7.73	8.58	7.37	3.27	2.46	1.99	1.66	7.81	6.67
AVM	2.51	1.70	4.60	2.46	5.05	3.12	2.18	1.18	1.33	1.15	5.07	3.10
PGA	2.51	1.50	5.48	3.71	5.76	3.79	2.24	1.32	1.75	1.56	5.90	4.02
CT	2.14	0.89	3.95	2.89	3.96	2.44	1.70	0.94	1.19	1.18	3.86	2.94

8.5 Publication 5

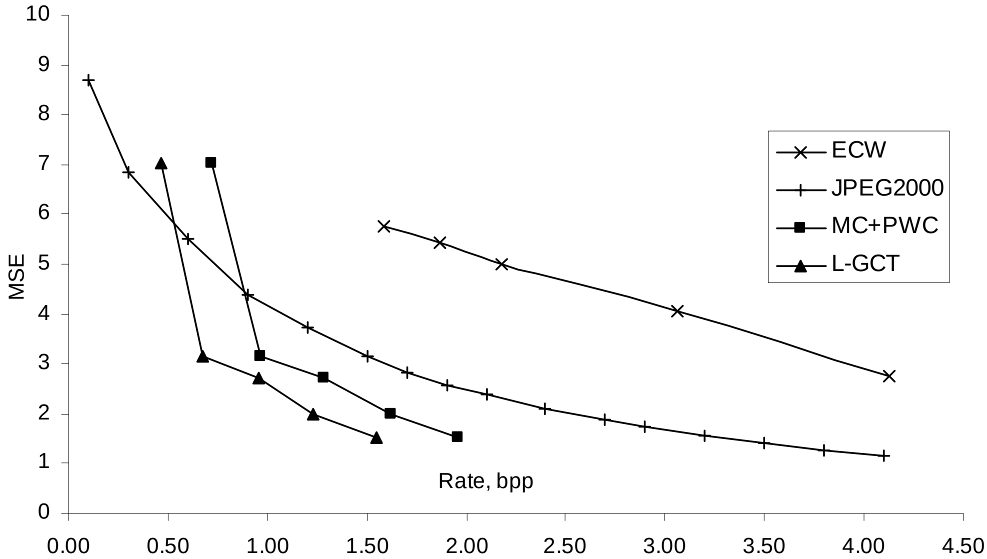


Figure 21. Operational rate-distortion function of the proposed algorithm (L-GCT) and its competitors.

Table 5. Compression performance of JPEG2000 and the proposed algorithm for similar objective quality level.

MSE distance	JPEG2000, Bpp	Proposed, Bpp	Improvement,%
1.52	3.20	1.55	51
1.99	2.40	1.23	48
2.71	1.70	0.95	44

References

[1] S. Ablameyko, T. Pridmore, *Machine Interpretation of Line Drawing Images*, Springer, 2000.

[2] A.K. Chhabra, D. Dori (eds.), Graphics Recognition – Recent Advances, Lecture Notes on Computer Science, vol. 1941, Springer, Verlag, 2000.

[3] NLS: National Land Survey of Finland, Opastinsilta 12 C, P.O.Box 84, 00521 Helsinki, Finland. http://www.nls.fi/index_e.html.

[4] CIE, *Colorimetry*, CIE Pub. No. 15.2, Centr. Bureau CIE, Vienna, Austria, 1986.

[5] K. Sayood, *Introduction to Data Compression*, 2nd edition, Academic Press, 2000.

[6] D. Salomon, *Data Compression: The complete reference*, 3rd Edition, Springer Verlag, 2004.

[7] I.H. Witten, A. Moffat, T.C. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Images. 2nd Edition*, Morgan Kaufmann, 1999.

[8] Graphics Interchange Format(sm), Version 89a, http://www.dcs.ed.ac.uk/home/mxr/gfx/2d/GIF89a.txt, 1990.

[9] S. W. Thomas, J. Mckie, S. Davies, K. Turkowski, J. A. Woods, and J. W. Orost. *Compress (version 4.0) program and documentation*, 1985.

[10] T. Welch, “A technique for high-performance data compression”, *Computer Magazine*, vol.17(6), pp. 8-19, 1984.

[11] T. Boutell, “PNG (Portable Network Graphics) specification”, ftp://ftp.uu.net/graphics/png/documents/

[12] P. Deutsch, “DEFLATE compressed data format specification”, RFC1951, http://www.faqs.org/rfcs/rfc1951.html, 1991.

[13] J. Ziv, A. Lempel, “A universal algorithm for sequential data compression”, *IEEE Trans. on Information Theory*, vol. 23(6), pp. 337-343, 1977.

[14] J. Ziv, A. Lempel, “Compression of individual sequences via variable rate coding”, *IEEE Trans. on Information Theory*, vol. 24(5), pp. 530-536, 1978.

[15] D. Huffman, “A method for the construction of minimum-redundancy codes”, *Proceedings of the I.R.E.*, pp. 1098-1102, 1952.

[16] International Telegraph and Telephone Consultative Committee (CCITT), “Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus,” Recommendation T.6, 1984.

[17] TIFF Revision 6.0, Adobe Developers Association, Adobe System Incorporated. http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf, 1992.

[18] ITU-T recommendation T.82, “Information technology – coded representation of picture and audio information – progressive bi-level image compression”, 1993.

[19] J. J. Rissanen, Langdon G. G., “Arithmetic coding”, *IBM Journal of Research*, Development 23: 146-162, 1979.

[20] W. Pennebaker, J. Mitchell, “Probability estimation for the Q-coder”, *IBM Journal of Research*, Development 32(6), pp. 737-759, 1988.

[21] G. G. Langdon and J. Rissanen, “Compression of black-white images with arithmetic coding,” *IEEE Trans. on Communications*, vol. 29, no. 6, pp. 858-867, June 1981.

[22] J. Rissanen, “A universal data compression system”, *IEEE Trans. on Information Theory*, vol. 29 (5), pp. 656–664, 1983.

[23] ITU-T recommendation T.88, “Information technology – coded representation of picture and audio information – lossy/lossless coding of bi-level images”, 2000.

[24] P. Howard, F. Konssentini, B. Martins, S. Forchhammer, W. Rucklidge, “The emerging JBIG2 standard”, *IEEE Trans. On Circuits and Systems for video technology*, vol. 8(7), pp. 838-848, 1998.

[25] X. Wu, N. Memon, “Context-based, adaptive, lossless image coding”, *IEEE Trans. on communications*, 45(4), pp. 437-444, 1997.

[26] X.Wu, “L_∞–constrained high fidelity image compression via adaptive context modelling”, *IEEE Trans. on Image Processing*, 9, (2000) 536-542.

[27] ISO/IEC JTC1/SC29/WG1, “JPEG-LS part-2 WD.” ISO/IEC JTC1/SC29/WG1 N938, 1998.

[28] M. Weinberger, G. Seroussi, G. Shapiro, “The LOCO-I lossless image compression algorithm: principles and standartization into JPEG-LS”, *IEEE Trans. on Image Processing*, vol. 9 (8), pp. 1309–1324, August 2000.

[29] S.W. Golomb, “Run-length encodings”, *IEEE Transactions on Information Theory*, vol. 12(3), pp. 399-401, 1966.

[30] R. F. Rice, "Some Practical Universal Noiseless Coding Techniques", Jet Propulsion Laboratory, Pasadena, California, *JPL Publication*, pp. 79-22, March 1979.

[31] Ausbeck, P.J., Jr. “The piecewise-constant image model”, *Proceedings of the IEEE*, Vol. 88, Issue 11, pp. 1779–1789, Nov 2000.

[32] Y. Yoo, Y. G. Kwon, A. Ortega, “Embedded image-domain compression using context models”, *Proc. of IEEE Int. Conf. on Image Processing 1999 (ICIP 99)*, Kobe, Japan, vol. 1, pp. 477–481, 1999.

[33] W.B. Pennebaker, J.L. Mitchell, *JPEG: Still Image Data Compression Standard*, New York: Van Nostrand Reinhold, 1993.

[34] D. Taubman, M. Marcellin, *JPEG2000: Image Compression Fundamentals, Practice and Standards*, Kluwer Academic Publishers, 2001.

[35] R.J. Clarke, *Transform Coding of Images*, Academic Press, 1985.

[36] S.Mallat, *A Wavelet Tour of Signal Processing*, 2nd Edition, Academic Press, 1999.

[37] C. Ueffing, “Wavelet based ecw image compression”, *Photogrammetric Week 01*, Wichmann Verlag, Heidelberg, pp. 299–306, 2001.

[38] LizardTech, “MrSID Technology Primer”, http://www.lizardtech.com/files/geo/techinfo/MrSID_Tech_Primer.pdf

[39] L. Bottou, P. Haffner, P. G. Howard, P. Simard, Y. Bengio, Y. Le Cun, “High Quality Document Image Compression with DjVu”, *Journal of Electronic Imaging*, vol. 7 (3), pp 410-425, SPIE, 1998.

[40] A. Gersho, R.M. Gray, *Vector quantization and signal compression.* Norwell, MA: Kluwer, 1992.

[41] W. Ding, Y. Lu, F. Wu, S. Li, “Rate-distortion optimized color quantization for compound image compression,” *Proc. SPIE Visual Communications and Image Processing (VCIP 2007)* (6508-98), San Jose, CA, USA, Feb. 2007.

[42] R. Balasubramanian, J.P. Allebach, “Sequential scalar quantization of vectors: an analysis”, *IEEE Trans. on Image Processing*, vol. 4 (9), September 1995.

[43] S.-C. Pei, C.-M. Cheng, “Dependent scalar quantization of color images”, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 5(2), pp. 124-139, Apr 1995.

[44] Y.-Kheong Chee, “Survey of progressive image transmission methods”, *International Journal of Imaging Systems and Technology, Special Issue: Image and Video Compression*, vol. 10(1), pp. 3-19.

[45] Gonzalez, R. C., Woods, R. E., *Digital Image Processing*, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 2002.

[46] I. Pitas, *Digital Image Processing Algorithms and Applications*, New York: Wiley, 2000.

[47] Dougherty E.R., Astola J. (eds), *Nonlinear Filters for Image Processing*, SPIE Optical Engineering Press, 1997.

[48] Pitas, I., Venetsanopoulos A.N., *Nonlinear digital filters: principles and applications*, Boston, Mass.: Kluwer, 1990.

[49] Serra J., *Image Analysis and Mathematical morphology*, London: Academic Press, 1982.

[50] Matheron G. *Random Sets and Integral Geometry*, J. Wiley & Sons, New York, 1975.

[51] Heijmans H.J.A.M., *Morphological image operators*. Boston: Academic Press, 1994.

[52] Koskinen L., Astola J. “Soft morphological filters: A robust morphological filtering method”. *Journal of Electronic Imaging*, vol. 3(1), pp. 60-70, 1994.

[53] M.L. Comer, E. J. Delp, Morphological operations for color image processing, *Journal of Electronic Imaging* 8(3), 279-289, July 1999.

[54] Dougherty E.R., “Optimal mean-square n-observation digital morphological filters. Part I: Optimal binary filters”, *Computer Vision, Graphics, and Image Processing*, vol. 55, pp. 36-54, 1992.

[55] Ping Z., Lihui C., Alex K.C., “Text document filters using morphological and geometrical features of characters”, *Proc. Int. Conf on Signal Processing-ICSP’00*, pp. 472-475, 2000.

[56] Doughberty E., Lotufo R., Hands-on morphological image processing. *SPIE Optical Engineering Press*, 2003.

[57] J. Astola, P. Haavisto, Y. Neuvo, “Vector median filters”, *Proc. of IEEE*, 78 (4), pp. 678-689, 1990.

[58] R. Lukac, “Adaptive vector median filtering”, *Pattern Recognition Letters*, 24, pp. 1889-1899, 2003.

[59] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, “Weighted median filters: a tutorial,” *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 43, issue 3, pp. 157-192, Mar. 1996.

[60] M. Kuwahara, K. Hachimura, S. Eiho, and K. Kinoshita, “Digital Processing of Biomedical Images”, *Plenum Press*, pp. 187-203, New York, NY, 1976.

[61] C. Kenney, Y. Deng, B.S. Manjunath, G. Hewer, “Peer group image enhancement”, *IEEE Trans. on Image Processing*, vol. 10 (2), 326-334, 2001.

[62] Ageenko E., Fränti P., "Context-based filtering of document images", *Pattern Recognition Letters*, 21 (6-7), 483-491, Elsevier Science, 2000.

[63] K. N. Plataniotis, D. Androutsos, and A. N. Venetsanopoulos, “Color image filters: The vector directional approach,” *Optical Engineering*, vol. 36, no. 9, pp. 2375–2383, 1997.

[64] R. Lukac, B. Smolka, K. Martin, K.N. Plataniotis, and A.N. Venetsanopoulos, “Vector filtering for color imaging”, *IEEE Signal Processing Magazine*, vol. 22 (1)74-86, January 2005.

[65] P. Kopylov, P. Fränti, “Filtering of color map images by context tree modeling”, *Proc. IEEE Int. Conf. on Image Processing (ICIP'04)*, Singapore, vol. 1, 267-270, October 2004.

[66] M. J.Weinberger, J. Rissanen, and R. B. Arps, “Applications of universal context modeling to lossless compression of gray-scale images,” *IEEE Trans. on Image Processing*, pp. 575–586, Apr. 1996.

[67] X. Wu, “Lossless Compression of Continuous-tone Images via Context Selection, Quantization and Modeling”, *IEEE Trans. on Image Processing*, no. 6, pp. 656-664, 1997.

[68] Ageenko E.I., Fränti P., “Compression of large binary images digital spatial libraries”, *Computers and Graphics*, vol. 24(1), pp. 91-98, 2000.

[69] S. Forchhammer and O. Jensen, “Content layer progressive coding of digital maps,” *IEEE Trans. Image Process.*, vol. 11, no. 12, pp. 1349–1356, Dec. 2002.

[70] P. Kopylov, P. Fränti. “Compression of map images by multi-layer context tree modeling”, *IEEE Trans. on Image Processing*, vol. 14, pp. 1-11, Jan. 2005.

[71] P. Heckbert, “Color image quantization for frame buffer display”, *Comput. Graph.*, vol. 16, pp. 297-307, 1982.

[72] A. Akimov, A. Kolesnikov and P. Fränti, “Lossless compression of color map images by context tree modeling”, *IEEE Trans. on Image Processing*, vol. 16 (1), 114-120, 2007.

[73] B. Martins and S. Forchhammer, “Tree coding of bilevel images,” *IEEE Trans. Image Process*., vol. 7, no. 4, pp. 517-528, Apr. 1998.

[74] X. Chen, S. Kwong, and J.-F. Feng, “A new compression scheme for color-quantized images”, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, no. 10, pp. 904-908, Oct. 2002.

[75] A.J. Pinho and A.J.R. Neves, “A context adaptation model for the compression of images with a reduced number of colors”, *Proc. of the IEEE Int. Conf. on Image Processing (ICIP-2005)*, September 2005, Genova, Italy.

[76] P. Fränti, T. Kaukoranta, D.-F. Shen and K.-S. Chang, “Fast and memory efficient implementation of the exact PNN”, *IEEE Trans. Image Process.*, vol. 9, no. 4 pp. 773-777, May 2000.