UNIVERSITYOF JOENSUU
COMPUTER SCIENCE
DISSERTATIONS X

ALEXANDER KOLESNIKOV

EFFICIENT ALGORITHMS FOR VECTORIZATION AND POLYGONAL APPROXIMATION

ACADEMIC DISSERTATION

To be presented, with the permission of the Faculty of Science of the University of Joensuu, for public criticism in Science Park, Länsikatu 14, Joensuu, on Month Xth, 2003 at 12.00 noon.

UNIVERSITY OF JOENSUU
2003

Efficient algorithms for vectorization and polygonal approximation

Alexander Kolesnikov

Department of Computer Science

University of Joensuu

P.O.Box 111, FIN-80101 Joensuu FINLAND

koles@cs.joensuu.fi

University of Joensuu, Computer Science, Dissertations N

Abstract

The thesis is dedicated to the study of problems of vectorization and vector data processing. The main purpose of the study is to develop efficient algorithms for large input data. The topic is divided into two areas.

In the first part of the thesis, we present the developed vectorization system. Firstly, we introduce locally adaptive binarization algorithm. Then we propose efficient implementation of distance transform based thinning algorithm for large binary images. The developed algorithm can be used in multiprocessor parallel machines as well as in single-processor computers. To reduce the processing time of the vector data analysis, we use AVL-trees for vector data presentation. We apply the developed system for feature-based filtration of binary images, which is used as a part of a context-based image compression.

The second part of the thesis is dedicated to the problem of polygonal approximation of digital curves for the raster-to-vector conversion and vector data processing (map simplification). The main purpose of the research is to bridge the gap between fast heuristic methods and the optimal but slow algorithms of quadratic or cubic time complexity. To solve the problem we introduce new paradigm of bounding corridor in the state space constructed along a reference solution. Using iterative reduced search approach in the corridor we give fast algorithms for *min*-ε *problem*. Then we apply the approach for solving *min*-# *problem* with joint use of error measures L_2 and L_∞. Furthermore, we introduce algorithms for the case of closed contours by using cyclical dynamic programming search for *min*-ε and *min*-# *problems*. Finally, we generalize the iterative reduced search approach to the case of multiple-object *min*-ε *problem* for vector data reduction. The time complexity of the developed algorithms is between $O(N)$ and $O(N^2)$ time. Trade-off between time and optimality can be controlled by the corridor width and the number of iterations. Near-optimal results can be achieved in linear time.

Keywords: Vectorization, binarization, thinning, polygonal approximation, optimal approximation, *min*-# *problem*, *min*-ε *problem*, dynamic programming, vector data reduction.

Contents

List of original publications

P1. Kolesnikov A., Belekhov V., Chalenko I., Vectorization of the raster images. *Pattern Recognition and Image Analysis*, 6(4): 784-794, 1996.

P2. Fränti P., Ageenko E., Kolesnikov A., Vectorizing and feature-based filtering for line-drawing image compression. *Pattern Analysis & Applications*, 2(4): 285-291, 1999.

P3. Kolesnikov A., Trichina E., A parallel algorithm for thinning of binary images. *Optoelectronics, Instrumentation and Data Processing*, 6: 7-13, 1995.

P4. Kolesnikov A., Fränti P., Reduced-search dynamic programming for approximation of polygonal curves. *Pattern Recognition Letters*, 24(14): 2243-2254, 2003.

P5. Kolesnikov A., Fränti P., A fast near-optimal *min-#* polygonal approximation of digitized curves. *Proc. of the IASTED International Conference on Automation, Control and Information Technology-ACIT'2002*, Novosibirsk, Russia, pp. 418-422, June 2002.

P6. Kolesnikov A., Fränti P., Polygonal approximation of closed contours, in *Lecture Notes in Computer Science*, Vol.2749, G.Goos, J.Hartmanis, J. van Leeuwen, (Eds.): Proc. of the Scandinavian Conference on Image Analysis-SCIA'2003, Göteborg, Sweden, pp. 778-785, June 2003.

P7. Kolesnikov A., Fränti P., *Data Reduction of Large Vector Graphics*. Research Report, A-2003-2. 2003.
Preliminary version: Kolesnikov A., Fränti P., Fast algorithm for multiple-objects *min-ε problem. Proc. of the International Conference on Image Processing-ICIP'2003*, Barcelona, Spain, September 2003. (accepted)

1. Introduction

1.1 Raster-to-vector conversion

Vectorization (raster-to-vector conversion) consists of analyzing of a raster image to convert its pixel representation to a vector representation The basic assumption is that a vector representation is more suitable for interpretation of the image, which typically is scanned graphical document (map, scheme, technical and construction drawing).

Initially the vectorization of maps and other documents has been performed manually on a digitizing table using a mouse. With appearance of document scanners the manual procedure was modified: the scanned map and the result are displayed on a high resolution screen in an overlay, and the vectorization is done manually using mouse. Later, with the development of the correspondent software, the procedure of vectorizaton became semi- or fully automatic.

To support the transition from manual processing to full automatic vectorization, a lot of problems in image processing, image analysis and pattern recognition has have to be solved during the last 30 years. Recently commercial software has became is available on market. But at the same time, we cannot say that all the problems have been completely solved. Tombre *et al*. wrote [Tombre'98]: "*Actually, the methods do work, but none of them is perfect. ... Although these methods yield good results, they all have their specific weakness, so that we cannot say that perfect raster-to-vector conversion is available*".

Of course, the future progress in this area is mostly connected with further development of methods and techniques of vector image analysis and interpretation. Along with this, the future progress is impossible without improvement of the basic (so called low-level) algorithms in image processing, including algorithms for image preprocessing and vectorization. This is caused by the following reasons. The first reason is that, with continuous increasing computer resources (speed, memory), the practical demands are also growing. For example, scanned document of E-size (34"×44") at 200 dpi resolution generates 80 Mb of raw data; at 2000 dpi resolution 800 Mb are needed [Ablameyko-Pridmore'00]. Using powerful workstations or special purpose hardware for solving practical tasks is expensive for most customers. Development of more efficient algorithms and methods for processing of large data with an ordinary desktop is still an important problem.

The second reason is introducing of new types of computers and devices with more constrained power and memory resources (laptops, palmtops, pocket PCs, wearable computers, communicators, mobile phones). Now image for processing can be acquired with handy scan, digital camera or received via Internet and using wireless connection. For example, user can scan Chinese or Japanese hieroglyphs on the street with pocket device with digital camera. Then the text will be recognized by the device and translated to the user. Another example of new area of vectorization module is sketchpad interface for palmtops and wearable computers.

Development of new algorithms and more efficient implementations for existed algorithms for the problem in question is an important task for the researchers. With the term "efficient algorithms" we mean algorithms that have good balance between quality and time performance.

1.2 Polygonal approximation

The problem of approximating a given two-dimensional piecewise linear curve by another coarser one is of fundamental importance [CG Impact Task Force'99]. In our case, the problem is important because of two reasons. At first, approximation of curves is an essential part of vectorization procedure. At second, the approximation of digital curves is widely used for vector data processing (data reduction, compression, map simplification) in digital cartography, GIS applications, and CAD systems.

For the last 30 years, the problem of polygonal approximation has been studied by many researchers. The approximation problem can be solved with optimal methods, but as Heckbert and Garland wrote [Heckbert-Garland'97]: "*Optimal simplification typically has quadratic or cubic cost, making it impractical for large inputs*". Heuristic algorithms of lower complexity have been developed for vectorization and vector data processing in practical applications. One can count up to dozen of different heuristic approaches to polygonal approximation whereas the number of proposed algorithms exceeds one hundred.

On the one hand, the existed optimal algorithms are too slow to be practical. On the other hand, the fast heuristic algorithms lack the optimality. The main goal of our research has been to develop efficient algorithms for polygonal approximations, which are very close to optimal (or even optimal), and have linear or near-linear time complexity. In other words, the developed algorithms have to be as fast as heuristic algorithms and yet provide results that are very close to those of the optimal algorithms.

1.3 Structure of the Thesis

In Chapter 2 we study the problem of raster-to-vector conversion in general, and then we consider the problem of processing of large input data, paying attention to more efficient implementation of some low-level processing algorithms including binarization and skeletonization. We also present binary noise filtering methods, paying special attention to methods that use vectorization.

In Chapter 3, we explore more deeply the polygonal approximation of digitized curves and vector data. Our goal is to develop efficient algorithms in this area including solutions for *min*-ε and *min*-# *problems*, approximation of closed contours, approximation of planar curves and 3-D paths, approximation of single and multiple curves, approximation of digitized curves for vectorization, vector data reduction and map simplification.

1.4 Summary of the publications

In the first paper (P1), we study the problem of vectorization of grayscale images of large size. The main purpose of the developed algorithms is processing of image of very large size. First, we offer algorithm for locally adaptive binarization of large grayscale images. Second, we present fast implementation of skeletonization algorithm for images of very large size. The algorithm was used in the development of parallel implementation in the paper **P3**. With this algorithm very large image can be processed by a single run of reading of the image file, instead of several time-consuming reading of the file in forward and backward directions. Third, we present algorithm for the analysis of vectors to define the most informative part of the obtained vector data. The developed algorithms and software were used in practice for vectorization as well for solving problem of noise filtering presented in publication **P2**.

In paper **P1**, the author developed the main principles and algorithms for the proposed raster-to-vector conversion system and implemented the pre-processing and vectorization parts. He is the principal author of the paper. The other two authors took part in the implementation of algorithms.

In the second paper (P2), we explore the problem of noise filtration in binary images of certain type. Normally, the noise filtration is a part of raster-to-vector conversion procedure. In the publication we use vectorization for noise filtration in binary images. The vector presentation was used for construction of reference raster image, which contains global context information about the input image. The context information was used for filtration of border noise in binary images. The main goal of the approach was to achieve a better compression of binary images by the elimination boundary noise. The vectorization procedure used algorithms and software presented in the publication **P1**. Due to the efficient implementation of the vectorization algorithm, the time cost of the vectorization phase is negligibly small (<10%) in comparison to the total processing time.

In paper **P2**, the author designed and implemented algorithms and software for the raster-to-vector conversion used in the feature-based filtering, and provided the vectorization experiments.

In the third paper (P3), a practical implementation of Distance Transform (DT) based skeletonization algorithm on parallel multi-processor system is presented. The parallel realization of the algorithm is based on a procedure developed for the raster-to-vector conversion system presented in paper **P1**. The image is divided into blocks, and the blocks are distributed among the parallel processors. After processing of the obtained data with two passes over the block in two directions the processors exchange the DT data on the borders. Then the processors perform a short additional run to complete the skeletonization process. The depth of the run is defined by the DT values on the borders of the blocks.

In paper **P3**, the ideas were developed and implemented by the author. He is the principal author of the paper. The second author contributed to the selection of the parallelization strategy.

In the forth paper (P4) we study the problem of the optimal approximation of open N-vertex polygonal curve with minimum error for a given number of linear segments M (*min*-ε *problem*) with error measure L_2. The purpose of the algorithm is to fill the gap between fast but sub-optimal and optimal but slow algorithms. We introduce paradigm of *bounding corridor* in the state space, and *iterative reduced search* in the corridor. The paradigm is the used for constructing efficient algorithm for *min*-ε *problem* in question as well as for solving a number of other approximation problems, which are presented in publications **P5, P6, P7**.

The proposed iterative reduced-search algorithm consist of three steps: (a) find a reference solution with any fast heuristic algorithm; (b) construct a bounding corridor of fixed width W along the reference solution in the state space; and (c) perform search in the bounding corridor with Dynamic Programming (DP) method. We repeat the steps (b) and (c) of the procedure, using output solution as a reference one for the next iteration. The time complexity of the developed algorithm is $O(W^2N^2/M)$, which is between $O(N)$ and $O(N^2)$, and the space complexity is $O(WN)$. Trade-off between performance and optimality can be controlled by the corridor width and number of the iterations.

We also propose modification of the state space in the case of *min*-ε *problem*, which reduces time complexity of the original full search algorithm in the case of large M. Then we offer more efficient computational scheme for the algorithm, which reduces processing time by elimination of approximation errors recalculation.

In the fifth paper (P5) we propose a fast near-optimal algorithm for solving the problem of *min*-# polygonal approximation of digitized curves. The algorithm is based on the reduced search approach introduced in **P4**. The algorithm consists of two steps. It first finds a reference approximation with minimum number of segments for a given error tolerance by using L_∞ error metrics. It then improves the quality of the approximation by the reduced-search algorithm with additive L_2 error measure. To combine the practicality of the distortion measure L_∞ and the high visual quality obtained with the error measure L_2, we proposed to use the distortion measure with metrics L_∞ as input control parameter ε, and the error with measure L_2 as cost function for the optimization. The algorithm is tailored for high-quality vectorization of digitized curves. The time complexity of the algorithms varies between $O(N)$ and $O(N^2)$.

In the sixth paper (P6), we introduce a new approach for *min*-ε and *min*-# approximation of *closed contours* based on dynamic programming method for open curves. It performs approximation of the cyclically extended contour and then makes analysis of the state space to select the best starting point. The processing time is double of that of the approximation of the corresponding open curve. The time complexity of the algorithms is defined by the complexity of approximation algorithms for open curves in use.

In the case of *min*-# *problem*, the analysis of the state space is reduced to the analysis of one-dimensional array of parent states. In fact, if any closed approximation polygon of size $M/2$ or less starts and ends the same vertex, we can select this vertex as the optimal starting point. With the introduced algorithms for approximation of closed contours, solution can be found for double processing time compared to the case of open curve. For solving the *min*-ε *problem* the suggested method can be used

along with iterative reduced search algorithm with time complexity between $O(N)$ and $O(N^2)$.

In the seventh paper (P7), we consider *multiple-object min-ε problem*, which can be formulated as optimal approximation of K objects (curves) with a given total number of linear segments M. To find optimal solution of the problem in question, we have to find (1) the optimal distribution of segments number among the objects as well as (2) the optimal approximation of all the objects with the optimal number of segments.

At first, we introduce a general solution to the problem using full search in the state space, and then we extend the iterative reduced search of **P4 t**o the case in question. The proposed algorithm includes three steps: a) approximation of the objects by reduced search in the *multiple-goal* bounding corridor, and calculation of the cost functions; b) calculation of the optimal allocation of the constrained resource among the objects using the calculated cost functions; c) restoration of the optimal solution for the found optimal number of segments. The procedure is then repeated iteratively several times (for near-optimal solution), or until no changes appear (for practically optimal solution). The time complexity of the proposed algorithm is between $O(N)$ and $O(N^2)$; and the space complexity is $O(N)$. Trade-off between performance and quality is controlled by the number of iterations and the corridor width. The developed algorithm is intended to be used for high quality vector data reduction.

In papers **P4-P7,** the author is responsible for developing the ideas, and implementation of the algorithms. He is also the principal author of the papers. The role of the second author has been mainly that of a supervisor.

2. Vectorization

2.1 Introduction

2.1.1 Problem formulation

Vectorization (raster-to-vector conversion) consists of analyzing a raster image to convert its pixels representation to a vector representation The basic assumption is that such a vector representation is more suitable for further interpretation of the image; this typically holds for a scanned graphical documents (maps, schemes, drawings). The topic was in focus for the last 30 years, for more details see the books [Kasturi-Trivedi'90, O'Gorman-Kasturi'94, Kasturi-Tombre'96, Tombre-Chhabra'98, Chhabra-Dori'00, Ablameyko-Pridmore'00], PhD Dissertations [Yu'94, Janssen'95, Trier'96, Lladós'97, Song'01, Novikov'01], surveys [Kasturi-O'Gorman'92, Kanungo-Haralick-Dori'95, Wenyin-Dori'99, Tombre–Ah-Soon–Dosch–Masini–Tabbone'00] and publications [Ablameyko *et al.*'95, Röösli-Monagan'95, Ogier-Mullot-Labiche-Lecourtier'98, Song-Su-Li-Cai'00, Song-Su-Tai-Cai'02].

The procedure of raster-to-vector conversion can be divided into three main stages: pre-processing, processing, and post-processing (see Fig. 2.1.1).

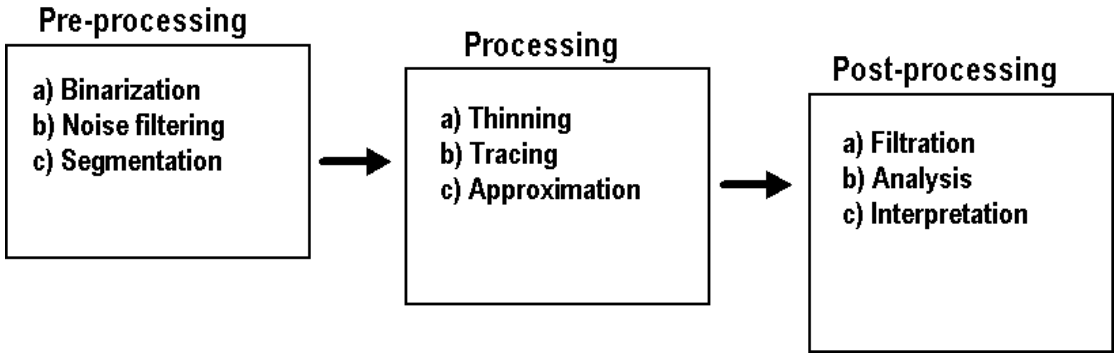


Figure 2.1.1. Three stages of raster-to-vector conversion.

2.1.2 Preprocessing

The purpose of the stage is to prepare the input raster image for processing (vectorization) at the next stage. Grayscale image should be binarized, grayscale or binary image can filtered for noise reduction, and color image has to be represented by monochrome layers (see Fig. 2.1.2). The type of pre-processing algorithm is defined by the type and quality of the input image.

We developed locally adaptive thresholding algorithm for binarization of large images [**P1**]. We also studied problem of noise filtration of binary noise in application to compression [**P2**].

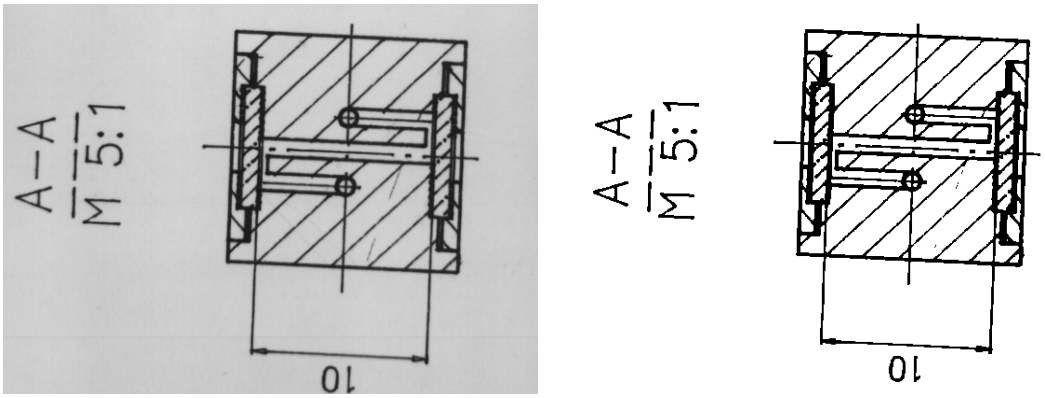


Figure 2.1.2: Illustration of raster image binarization: a) input halftone image; b) binary image after thresholding of the input image.

2.1.3 Vectorizing

Conversion of the raster binary image to vector form is performed at this stage. Output data of the stage is vector presentation of the input binary image. There are six basic approaches for vectorization of binary image:

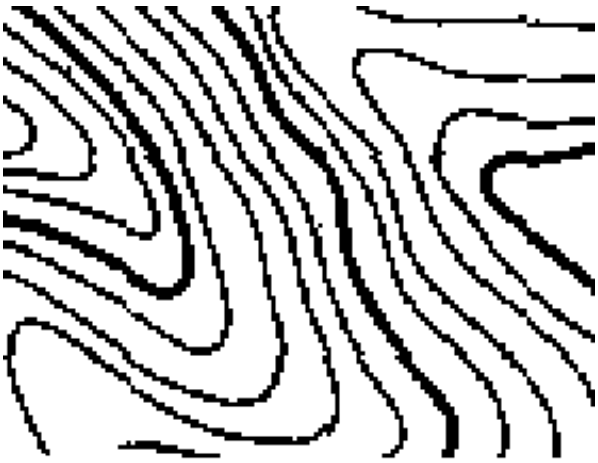
1) Thinning-based methods [Janssen-Vossepoel'97];
2) Contour-based methods [Hori-Tanigawa'93];
3) Graph-structure based methods [Novikov'01];
4) Sparse pixel tracking methods [Chiang-Tue-Leo'98, Dori-Liu'99];
5) RLE-based methods [Piper'85, Pavlidis'86, Monagan-Röösli'93];
6) Orthogonal zig-zag method [Chai-Dori'92, Dori'97].

Every method has advantages and drawbacks. The skeleton-based methods have good results, but tend to be very sensitive to noise. Contour-based methods are more noise-tolerant but they rely on heuristic and complex matching schemes. Correct choice of method is defined by data type (maps, drawings, schemes), and practical goals. For example, for 2-dimensional object vectorization the contour-based methods are more relevant, but for elongated objects the thinning-based approach seems to be reasonable.

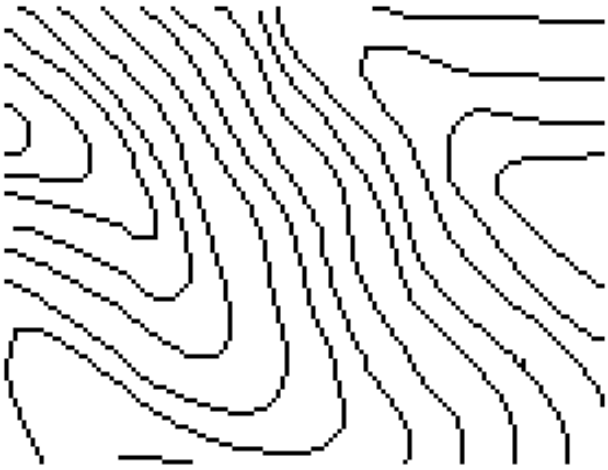
We use vectorization method based on thinning algorithm [**P1, P3**]. The procedure consists of three steps (see Fig. 2.1.3):

1) Binary image skeletonization by Distance Transform based thinning;
2) Tracking of skeletal branches to get chain coded digitized curves;
3) Primary vectorization of the digitized curves.

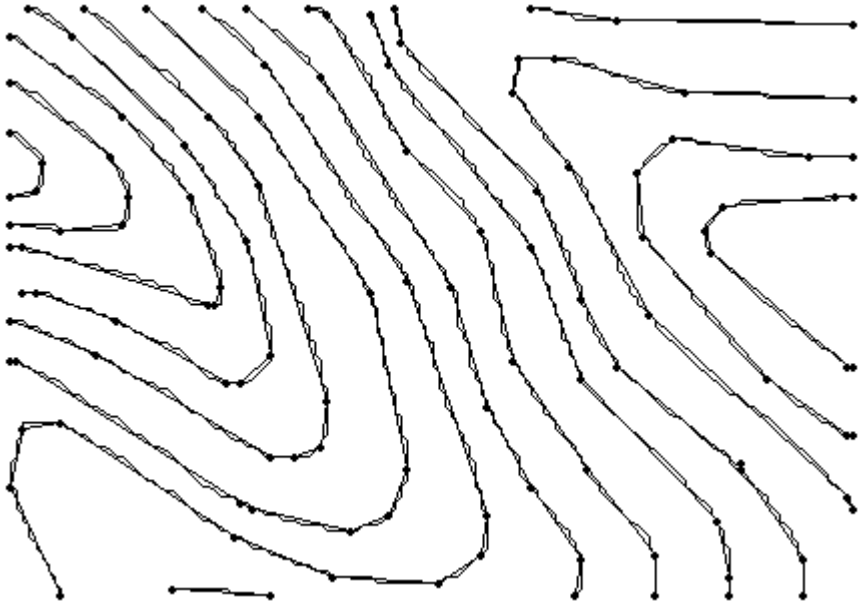
The primary vectorization can be performed with zero or non-zero approximation error. In the first case the chain code data are converted into segments of digital line in $O(N)$ time [Aoyama-Kawagoe'91]. The primary vectorization can be performed with polygonal approximation with error tolerance defined by width of the curve [**P5**].



(a)



(b)



(c)

Figure 2.1.3. Illustration of primary vectorization of elevation map (fragment): a) raster image; b) skeleton of the raster image; c) primary approximation of the skeleton with error tolerance ε=0.99. Vectors are labeled with dots.

2.1.4 Postprocessing

The main goal of this stage is vector data analysis and interpretation. The purposes of this step are following: a) removing noise from the vector model; b) object recognition; c) recovering entities from vector data (object vectorization).

Usually the algorithms for vector data interpretation are based on domain knowledge [Ablameyko *et al*.'94, Trier'96, Lee-Cho-Choy'00, Song-Su-Chen-Cai'00, Song-Su-Li-Cai'02, Song-Lyu-Cai-Cai'02]. The main purpose of the developed system [**P1**] was to reduce the number of manual operations required for inputting the graphic data into a computer, leaving the editing of the resulting vector representation for an operator. To reduce processing time for large images the obtained vector data were presented as AVL-tree. The employed model ensures a fast and simple access to the vector data for further analysis. The analysis of vector data includes the following procedures: gaps filling, vectors classification, false branches elimination, right corners rectification [**P1**] and vector data simplification by polygonal approximation [**P5, P6**].

2.1.5 Proposed improvements

The purpose of the study is realization of raster-to-vector conversion system. Special attention was paid to development of efficient algorithms for processing of large size images. We have concentrated on the following low-level processing algorithms:

a) Binarization of large images [**P1**];

b) Thinning of large images [**P3]**;

c) Binary noise filtration [**P2]**;

d) Analysis of vector data [**P1**];

e) Polygonal approximation of curves [**P4-P7**].

In Sections 2.2-2.4, we consider some problems of all stages of raster-to-vector-conversion: binarization, thinning and binary noise filtering. Correspondent problems of polygonal approximation will be discussed in details in the Section 3.

2.2 Binarization of large images

2.2.1 Problem formulation

The input gray-scale image has to be binarized before the skeletonization stage. The purpose of the binarization procedure is to segment the image into background and object pixels (see Fig.2.2.1). The quality of the skeleton depends on the quality of the input binary image.

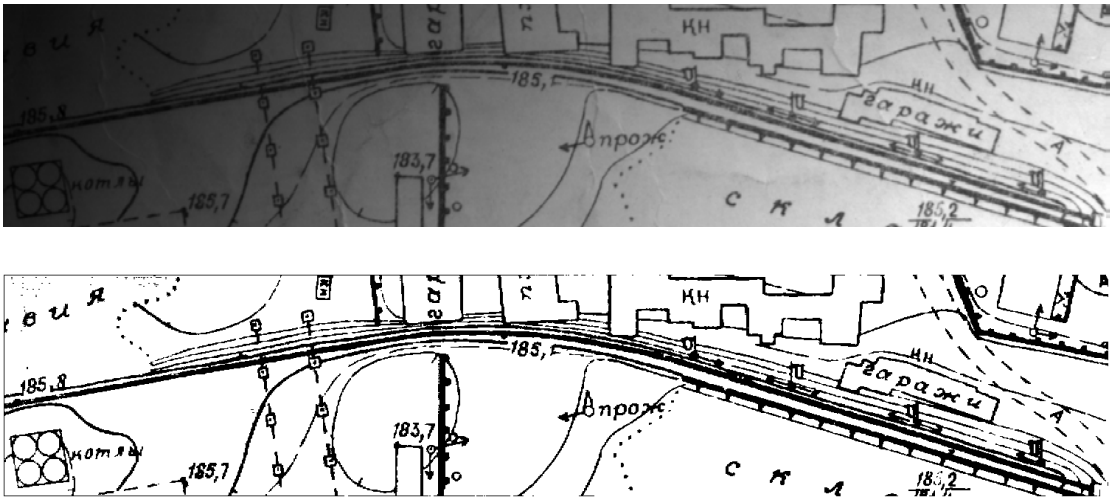


Figure 2.2.1. The input greyscale image (top); result of binarization with locally adaptive threshold (bottom).

The binarization of greyscale image can be performed by thresholding with some threshold T. If pixel value $p(x, y)$ is less than the threshold T, the pixel belongs to background, and thresholded pixel value will be '0', otherwise the pixel belongs to an object and value '1' will be assigned to the pixel.

In *global* thresholding, the same threshold value is applied to every pixel of the input image. In practice, because of non-uniformity of the background in the input image, the thresholding with global threshold provides poor result. In this case, *local* thresholding with threshold adjusted to the local properties of the image should be applied to obtain more reliable binary image (see Fig. 2.2.1). For this purpose, the input is divided into a rectangular blocks, and each of them is processed with *adaptive threshold* defined by the statistical properties of the block (see Fig. 2.2.2).

There is a lot of algorithms for threshold calculation based on different approaches [Weszka'78, Otsu'79, Sahoo-Soltani-Wang'88, Dunn-Joseph'88, Taxt-Flynn-Jain'89, Lee-Chung-Park'90, Glasbey'93, Pal-Pal'93, Trier-Jain'95, Trier-Taxt'95, Trier'96, Jawahr-Biswas-Ray'00, Ablameyko-Pridmore'00]. We have used Otsu's algorithm [Otsu'79], which is based on clustering of gray-level histogram by maximization of between-class variance. The method has shown good results for the test images in use.

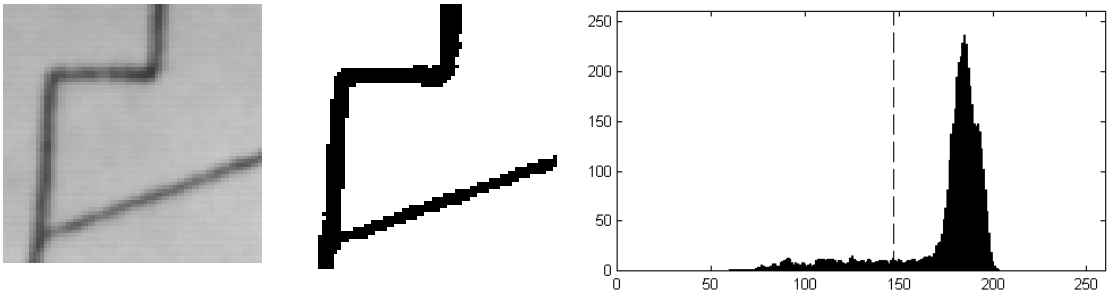


Figure 2.2.2. An example of input image thresholding: 60×60 block of greyscale image (left); result of binarization with threshold T=147 calculated by Otsu's method (center); histogram of the block (right). Threshold is shown by dashed line.

2.2.2 Locally adaptive binarization

Calculation of the locally adaptive threshold is based on information collected in a window surrounding the pixel to be processed [Niblack'86, Bernsen'86, Eivkil-Taxt-Moen'91, Sauvola -Pietikäinen'97, Sauvola'97, Sauvola-Pietikäinen'00].

In our study we considered the case of binarization of scanned maps and drawings under assumptions that quality is satisfactory and illumination over the image is smooth (quasi-linear linear) (see Fig. 2.2.1). The main attention was paid to development of algorithm for binarization of large size images. The term "large size image" means that the size of the image exceeds size of available computer memory.

To achieve this goal, we, first apply *locally adaptive* thresholding technique, and then perform the binarization of large image in *file-to-file* manner without keeping the whole image in memory. The input image is sequentially loaded and processed in stripe-by-stripe way so that partition of the image for loading and processing does not depend on the partition of the image for the analysis.

The developed algorithm consists of two steps: a) collecting and analysis of the histogram data, b) thresholding the image.

a) Collecting and analysis of the histogram data

The input image is processed fragment-by-fragment to collect histogram for the blocks. Histograms are collected for all blocks. If block contains objects, threshold is calculated for the block, otherwise the block is treated as empty. To improve the balance between object and background pixels in histogram we eliminate pixel from consideration when Laplacian $L(x, y)$ is *below* a certain threshold [Weszka-Nagel-Rosenfeld'74]. In this way, most of the background points will be excluded from histogram, while most of the object points will be kept in the histogram (see Figs. 2.2.3 and 2.2.4). Of course, mostly the noisy background points do contribution to the modified histogram. Nevertheless, usually these points belong to the background, so they can be treated as quite representative pixels for threshold calculation.

Analysis of the block histogram is performed to classify the block either as empty or non-empty. If block contains both background and object pixels, the histogram is expected to be wider than those of the blocks with background pixels only. So, if histogram width of a block is smaller than threshold W_H, the block is treated as empty without any object points. For empty blocks we get a preliminary threshold $T_0(i, j)$ from the nearest non-empty neighbours. If some non-empty block was classified as empty, it will be provided with threshold value from the nearest non-empty blocks.

For non-empty fragments we calculate the preliminary threshold $T_0(i, j)$ by Otsu's method [Otsu'79] from the block histogram.

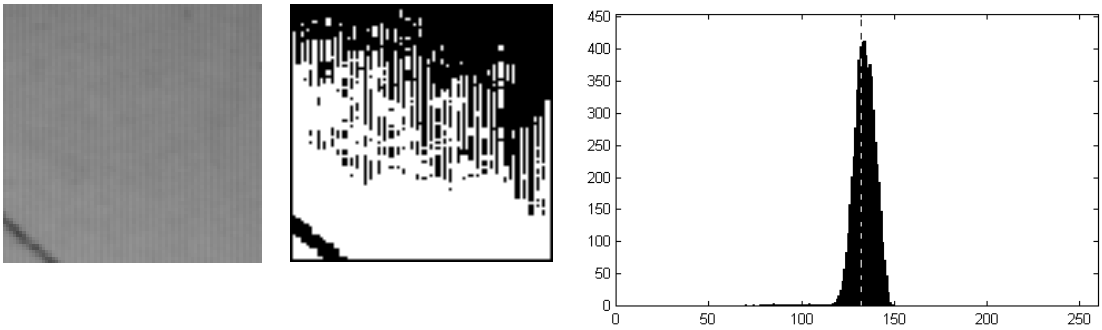


Figure 2.2.3. Thresholding of a block with a small object: block of the input greyscale image (left); result of binarization (center) with threshold T_0=132 calculated from histogram of the block (right). Threshold in histogram is labeled by dashed line.

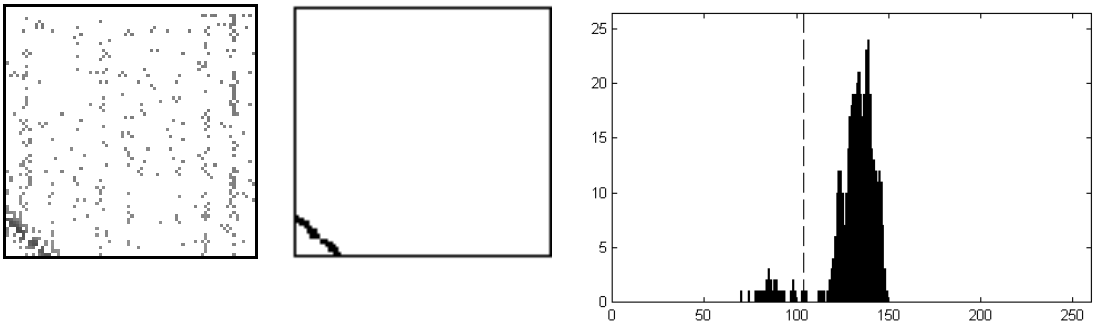


Figure 2.2.4. Thresholding of block with a small object: the masked block of the input greyscale image, defined for Laplacian threshold L_1=25 (left); result of binarization (center) with threshold T_1=104 calculated from histogram of the masked block (right). The masked pixels are shown as white. Threshold in histogram is shown by dashed line.

We calculate threshold $T_1(i, j)$ for blocks as average of the preliminary thresholds $T_0(i, j)$ of their eight neighbours. If some empty block was erroneously classified as non-empty one with wrong preliminary threshold, this operation can correct the final threshold for the block. Taking into consideration that histogram of background pixels is relatively narrow, even a small shift down of the threshold gives correct thresholding of the block.

b) Image thresholding

The input image is thresholded using the calculated 2D array of thresholds $T_1(i, j)$ for blocks. The local threshold $T_L(x, y)$ for point (x, y) is computed as bilinear approximation of thresholds $T_1(i, j)$ for four neighbouring blocks (see Fig. 2.2.5).

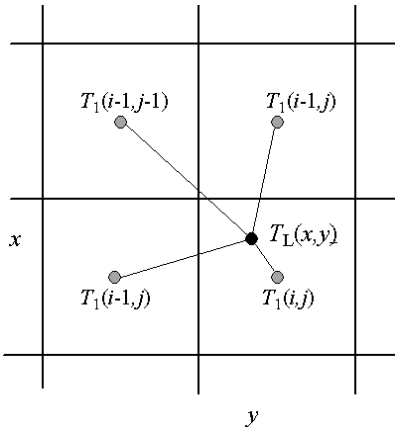


Figure 2.2.5: Scheme of four neighbouring blocks for computation of the local threshold $T_L(x, y)$ with bilinear interpolation.

2.2.3 Summary

The designed scheme for locally adaptive binarization of large grayscale images has been tested with grayscale images acquired by scanner of projective type. The provided tests have shown high efficiency of the algorithm for grayscale images with quasi-linear non-uniformity of background illumination.

2.3 Thinning algorithm for large binary images

The skeleton of a binary object is a shape descriptor, which can be regarded as a convenient alternative of the elongated object itself [Pfaltz-Rosenfeld'67]. Thinning a binary image down to its skeleton allows one to transform the image into a line drawing, which still contains the relevant information (see Fig. 2.3.1). For many applications such transformation is very useful, because it reduces drastically the amount of data to be handled, and simplifies computation procedures required for description and classification purposes.

The skeletonization (thinning) process can be seen as an isotropic retraction of the original object, down to its unit width subset. This subset is placed in the medial region of the object, has the same topology, and allows the evaluation of the spatial dimensions as well as orientation of the object. The skeleton can be obtained by thinning of the binary object in two ways:

a) Algorithm based on morphological thinning that preserve homotopy (iterative peeling);

b) Algorithm based on Distance Transform (DT) that preserve reversibility.

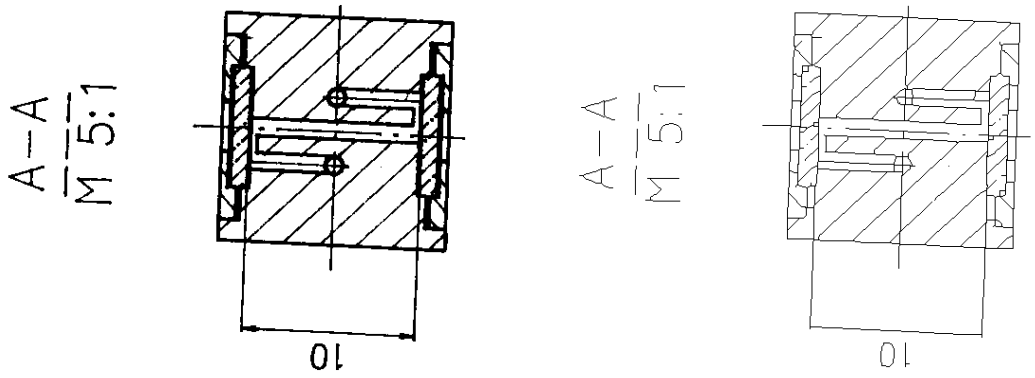


Figure 2.3.1: The input binary image (left) and the thinned image (right).

2.3.1 Iterative peeling algorithms

The skeleton of an image is built by iteratively peeling off the boundary pixels until only no erasable pixels remain in the image [Arcelli-Cordella-Levialdi'75, Poty-Miguet'94, Manzanera-Bernard-Prêteux-Longuet'99, Mazzariol'01, Manzanera-Bernard'03]. Number of runs for this method is equal to half-width of the object. The iterative thinning algorithms can be further divided into two categories: *sequential* and *parallel*.

We have to distinguish difference between parallel or sequential *algorithm* and parallel or sequential *realization* of the algorithm. The terms "sequential realization" or "parallel realization" are related to practical implementation of thinning algorithm with sequential or parallel machines, whereas the terms "sequential algorithm" or "parallel algorithm" specify the main principle of raster data processing.

Sequential algorithms examine contour points for deletion by either a) raster scanning, or b) contour following algorithms. In sequential algorithm result of

processing for the current pixel depends on the results for already processed points in the current point neighbourhood. Usually the thinning is performed by sequential scanning with window of 3×3 size [Arcelli-Cordella-Levialdi'75], or larger [Poty-Ubeda'93].

In *parallel* thinning algorithms result for the current pixel is performed independently on the current states of the neigbouring points and depend only on results for the previous iteration of thinning applied to the whole image [Stefanelli-Rosenfeld'71, Zhang-Suen'84, Jolt-Stewart-Clint-Perrott'87, Wang-Zhang'89, Wu-Tsai'92, Chen-Hsu'93, Zhang-Wang'94]. Parallel algorithms for preserving the connectivity of skeleton use either larger neighbourhood than 3×3 or use more than one pass over the image [Zhang-Suen'84, Suzuki-Abe'87, Chen-Hsu'88, Hall'89, Guo-Hall'89].

Sequential and parallel realization of the peeling algorithm

If image file is larger than available memory resources of the ordinary single-processor machine, the image has to be divided into overlapped fragments of smaller size and loaded for processing in fragment-by-fragment fashion. Size of the fragments is defined by size of available memory resource.

Skeletonization of large images with the iterative algorithm can be a time-consuming procedure. To reduce processing time for solving practical tasks, parallel multiprocessor systems or a special hardware are used. The parallel version of the iterative peeling algorithm for skeletonization is included into so called Cowichan problem set [Willson'94, Wilson-Bal'96] as benchmark task for parallel programming systems.

To be processed on distributed-memory multi-processor machines [Chung-Prasanna-Wang'87, Baek-Teague'90, Heydorn-Weidner'91, Hayat-Naqvi-Sandler'92, Xu-Lau'94, Prasanna-Wang'94, Baek-Teague-Chang'94, Poty-Miguet'95, Chung-Prasanna'95, Ubeda'95, Xu-Lau-Diekmann'97, Montoya-Garcia'98, Mazzariol-Gennart-Hersch'00] image is divided into overlapped rectangular fragments (blocks). The fragments are distributed among the processors. After every run the processors exchange data on the border of the fragments (one line). Number of runs is defined by maximum width of the objects in the image. It affects in big number of time-consuming data exchange operations between processors. Exist a number of massively-parallel realizations of the peeling algorithms with $O(N)$–$O(N^2)$ processing elements, here N is size of the image [Milgram-Saint Pierre'90, Zhang-Mahgoub'95, Rhee-Lee'97, Zhou-Wei-Li-Woo'99].

2.3.2 Distance-transform based methods

The alternative method is to first calculate the Distance Transform (DT) of the binary image. Distance Transform is defined for an object point as a distance from the pixel to the nearest background point [Rosenfeld-Pfaltz'66, Pfaltz-Rosenfeld'67, Rosenfeld-Pfaltz'68, Borgerfors'84,'86]. At first Distance Transform is performed then skeletal points are detected on the DT using some rules [Rosenfeld-Pfaltz'66, Pfaltz-Rosenfeld'67, Rosenfeld-Pfaltz'68, Toriwaki-Yokoi'81, Arcelli-Sanniti di Baja'84,'85, Suzuki-Abe'86, Borgerfors'86, Arcelli-Sanniti di Baja'89, Niblack-Gibbons-Capson'92, Ragnemalm-Ablameyko'93, Shih-Pu'95, Pridmore-Ablameyko'96]. Choice of the distance metrics depends on the task to be solved:

Euclidean, octagonal, chessboard, city-block, weighted chamfer. In our application we use chessboard distance measure for Distance Transform, because it provides undistorted skeleton of rectangular objects (see Fig. 2.3.2).

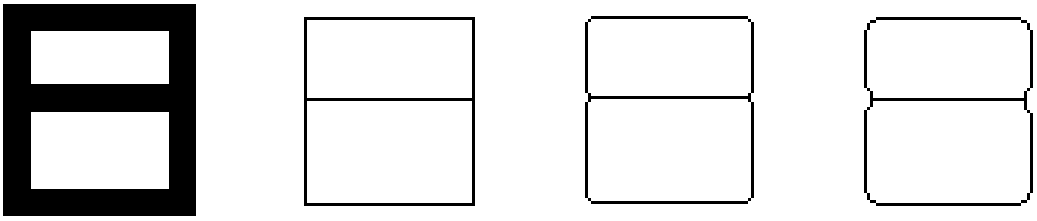


Figure 2.3.2: Illustration of skeletons for different distance metrics: a) the test rectangular shape with lines of width 9; skeleton for b) chessboard, c) Euclidean, and d) city-block distance metrics.

Distance transform can be completed in two raster scans through the whole image a 2×3 window. The first scan is carried out in a top-left to bottom-right direction. During this scan for every pixel in an object the distance form the top and left object boundary is determined. The second scan in a vice versa direction determines the distance from every pixel in the object to the bottom and right boundaries in a similar way. Detection of skeletal points is performed by means of two subsequent scans through the whole image by a 3×3 window in the same fashion as for the distance transform. One additional scan is required to reduce two-pixels width skeleton to the unit-width one and to convert a distance-labeled skeleton into width-labeled one. Thus the image skeletonization can be accomplished in a sequence of five raster scans through the image. Every subsequent scan of DT and skeletal point detection is carried out in the direction opposite the previous one and requires the complete result from the previous scan.

Sequential and parallel realization of the DT-based algorithm

As we can see, the thinning procedure with DT-based algorithm can be performed in a fixed number of runs regardless the objects size, but the algorithm demands more complicated processing operations for large images than those for the peeling algorithms. Tombre *et al*. wrote in [Tombre–Ah-Soon–Dosch–Masini–Tabbone'00]: "*However, it is difficult to compute the distance transform without storing the whole image in memory*". Although DT-based algorithm is faster than peeling procedure, the method was not widely used in practical applications for processing of large images for ordinary single-processor machines and parallel multiprocessor systems. Analysis of publications has shown that most papers are dedicated to realization of peeling thinning algorithms with parallel multi-processor systems. The more efficient parallel or sequential realization of DT-based thinning algorithm was out scope of researchers.

2.3.3 Fast implementation of the thinning algorithm

Number of different skeletonization algorithms exceeds at least 300 items, for more details see the comprehensive surveys [Lam-Lee-Suen'92,'95, Ablameyko-Pridmore'00]. In our research we were concentrated on *efficient implementation* of the existed thinning algorithms than on development of new methods for thinning. The

main goal of our studies was development of efficient implementation of DT-based skeletonization algorithm, which is suitable in practice for processing of large input with ordinary single-processor or parallel multiprocessor machines. The classical DT-based thinning algorithm of Arcelli and Sanniti di Baja [Arcelli-Sanniti di Baja'85] with chessboard distance metrics has been selected for realization, but the approach can be used for any other DT-based algorithm with different distance metrics.

At first, we developed realization of the skeletonization algorithm for sequential computers as a part of raster-to-vector conversion system [**P1**, **P2**]. Later we extended the approach on the case of parallel multiprocessor systems [**P3**].

In the case of image skeletonization with ordinary single-processor machine the image file is being read and processed by overlapped blocks (see Fig. 2.3.3). The loaded current image block is processed in both directions several times according the skeletonization algorithm in use. When the total processing of the current block is completed, size δ of the overlapping is defined by the maximum value $D_{\max}$ of Distance Transform in the last row: $\delta = \lceil D_{\max}/2 \rceil$.

The main idea of the proposed approach is as follows: a) all the necessary processing operations are performed within one block, and b) size of overlapping is controlled with DT value to insure the correctness of the procedure. With this method image of any size can be processed with only one reading of the image file with minimum overlapping of blocks.

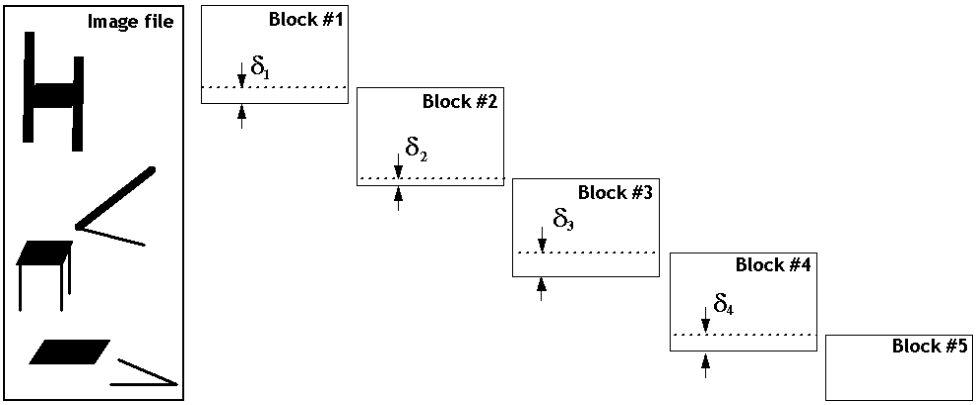


Figure 2.3.3: Scheme of overlapped blocks loading. Size δ_k of the overlapped zones depends on thickness of objects at the blocks border and calculated from the distance transform at the last row in the current block.

For realization of the proposed skeletonization algorithm for parallel processing system we selected distributed-memory Multiple Instructions Multiple Data (MIMD) model of parallelization [Kolesnikov-Trichina'95,'96, **P3**]. The selected model is more practical than massively-parallel machine, which demands special hardware with $O(N)$–$O(N^2)$ processing elements [Schwarzkopf'91, Chandran-Kim-Mount'92, Jenq-Sahni'92, Fujiwara-Inoue-Masuzawa-Fujiwara'96,'97, Lee-Horng'99, Chia-Wang-Chen-Lou'02].

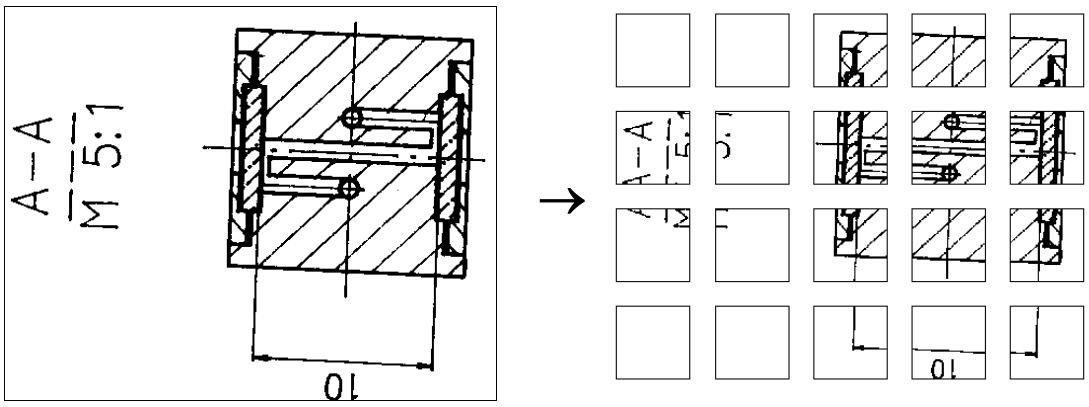


Figure 2.3.4: Scheme of input image partition into rectangular blocks among processors: input binary image (left), twenty image blocks for 4×5 processor mesh grid (right).

According the computational scheme for parallel processing the image is divided into blocks, which are distributed among the processors (see Fig. 4). After the first run the processors exchange data on the border (one line). After the second run with the correspondent data exchange, an additional short run is performed. The depth of the third run δ is defined by maximum difference of values of the distance transform on the border $D_B(x)$ and in the received image row (or column) $D_R(x)$ from the neighbouring processor: $\delta=\max\{0, \lceil (D_B(x)-D_R(x)/2 \rceil\}$, where $1 \le x \le$ size. The approach allows getting result of skeletonization after fixed number of runs regardless maximum line width with minimum number of additional operations in comparison with the case when the whole image is available.

Vossepoel *et al*. [Vossepoel *et al*'97] proposed the analogous approach for DT-based thinning algorithm of large images. But according to their approach depth of the third (additional) run was fixed: δ=100 pixels. In the case of processing with parallel system, which is considered in the paper [Vossepoel *et al*'97], the difference between two approaches is not crucial, because the total processing time depends on the depth of the last run about linearly. For the ordinary single-processor machine the dependence of the processing time on the size of overlapped zone is nonlinear: $T \sim H/(H-\Sigma\delta_k)$, where H is the image height and δ_k is overlapped zone size for kth block. That is why in this case the processing time is more sensitive to the size of overlapped zones, especially for very wide input images. With the proposed method [**P3**] with control of overlapping size we can reduce time-cost of thinning algorithm to the minimum. For example, if the maximal width of lines on blocks borders does not exceed 20, the size of overlapped zone will be about 10 pixels, that it much less than 100 pixels as in the approach [Vossepoel *et al*'97].

Another method, which has been applied to reduce processing time, was using of look-up tables for DT-based skeletonization. During scanning procedure 9-bit index I_k for the current point is computed recursively from the index I_{k-1} for the previous point using three new pixels that just appeared in the 3×3 window. To avoid redundant computation of the index for background points, using of the recursive scheme starts with beginning of object pixels run.

2.3.4 Summary

The developed efficient realization of the skeletonization algorithm for large images has allowed us to drastically reduce processing time for raster-to-vector conversion system [**P1**]. As it was mentioned in **P2**, where the vectorization procedure was used as part of binary image compression scheme, processing time for vectorization takes only 10% of the total processing time.

2.4 Vectorization-based binary noise filtering

Scanned binarized images have a noise which can appear because of low quality of originals, paper defects, non-optimal threshold setting of binarization process, and non-uniform illumination. In the case of binary images, the noise appears as *content-dependent*, distorting the contours of objects, and as salt-and-pepper additive noise (randomly scattered noise pixels). The noise level may be low enough not to significantly detract from the quality, but it introduce unnecessary details that decrease the analysis or compression of the binary image, and distort the vector presentation of the binary image.

Normally the size of noise patterns is less than those of binary objects (lines, strokes, symbols). Also the objects details are more structured than the noise patterns. Therefore, noise pixels in the noise patterns are less correlated with neighbours than pixels of the objects. This correlation is used in most algorithms for binary image enhancement and noise reduction.

2.4.1 Survey of solutions

Several approaches for binary noise filtering have been considered by analyzing the local pixel neighbourhood defined by a filtering template. These filters use a set of rules to accept or reject the pixel, such as predefined masks or a quantitative description of the local neighbouring area.

Recent research in mathematical morphology has shown that morphological filtering can be used as an efficient tool for pattern restoration in an environment with a lot of additive noise [Schonfeld-Goutsias'91, Dougherty'92, Loce'92, Heijimans'94, Liang-Haralick-Phillips'96, Dougherty-Astola'97].

Jin *et al*. [Jin-Ong-Jajasooriah'95] proposed a new class of morphological operators for binary images, it is the domain operators. The basic idea is taken from ranked-order filters, but generalized with the incorporation of the fuzzy index function in weight representation. Chinnasarn *et al.* [Chinnasarn-Rangsanseri-Thitimajshima'98] utilized mathematical morphology approach to modify *kFill* algorithm of O'Gorman [O'Gorman'92]. With the proposed heuristic rules for image filtering in 3×3 and 4×4 window, they reduced the number of iterations to a single-pass scan over the image.

Several methods have been considered for image processing by analyzing the local pixel neighborhood defined by a filtering template. Techniques have been proposed based on the analysis of context information [Ping-Lihui-Alex'00, Randolph-Smith'00, Zheng-Kanungo'01].

Wahl [Wahl'86] introduced algorithm which operates within 5×5 window. The objective of the processing is to eliminate noise utilizing four heuristic rules. Model-based approach for binary noise filtering have also been used [Haralick-Dougherty-Katz'91, Schonfeld-Goutsias'91, Sidiropoulos-Baras-Berenstein'94]. This approach assumes a specific probabilistic models that describe the behaviour of both signal and noise patterns, which are elementary geometrical primitives from which the signal and noise images are constructed.

The algorithm of Nikiel [Nikiel'96] is based on the calculation of fractal dimension of binary image in 8×8 sliding window. The filtering is performed in two steps: (1) extraction and estimation of the fractal dimension, and (2) classification and actual filtering (noise suppression and solid refinement).

Ping *et al*. [Ping-Lihui-Alex'00] proposed two algorithms for binary images filtering. The first algorithm called Modified Directional Morphological Filter (MDMF) is introduced with dual properties for eliminating document salt-and-pepper noise and for remedying eroded character stroke distortion. For eliminating larger noise, another algorithm called *Image Geometric Structure Filter* (IGSF) is proposed based on the geometric stroke information of characters.

Randolph and Smith [Randolph-Smith'00] used a *binary angular filter banks* for directional decomposition to enhance fax documents. The filter banks provide representations that delineate the directional components in the text letters enabling edges and contours to be smoothed appropriately.

For the small window size of 3×3, there are 512 unique table entries for binary image processing, which by itself is quote manageable. With the bigger filtering window more global information is taken into account, but straightforward using of larger windows demands more memory resources: 5.0×10^{14} unique entries in analysis tables for 7×7 window, which is making the table management impractical [Randolph-Smith'00]. Following a clustered mapping approach [Kossentinni-Smith'98] based on PNN algorithm, an efficient tree structured mapping function can be constructed that allows all entries to be mapped to a set of weights.

Ageenko and Fränti [Ageenko-Fränti'00] proposed two context-based filtering methods, namely *Simple Context Filters* and *Gain-Loss Filters* for the enhancement of document images. They used the 10- and 20-pixel causal templates from JBIG to collect statistics during the analyzing phase. Then, in the filtering phase all rare pixels in low entropy contexts are flipped.

In [Zheng-Kanungo'01a,'1b] *morphological degradation model* was proposed for binary images. According the model, the probability of a pixel flipping from foreground to background, or vice-versa, is an exponential function of its distance from the nearest boundary point. Based on the model they offered restoration algorithm, which includes two stages: a training stage to define parameters of the model, and a restoration stage. The training stage includes joint analysis of degraded and the correspondent *ideal* image by computing the conditional distribution between the noise pattern pairs.

Fränti *et al*. [Fränti-Ageenko-Kälviäinen-Kukkonen'98,'02] used Hough Transform (HT) for extracting vector features from binary image. A *feature image* is reconstructed from the extracted linear segments and it is utilized in the filtering phase. The filtering is based on noise removal procedure using the original and feature images. The noise-filtering algorithm was used to improve quality of context-based compression algorithm. The drawback of this approach is that the HT-based feature extraction phase dominates the processing time in the compression phase and makes it an order of magnitude slower than JBIG compression procedure. In practice it means that the feature extraction phase for 6 test binary images of total size 4 Mb takes about two hours with Pentium-200 machine. For comparison, the processing time for the compression phase is 1.5 min only.

2.4.2 Feature-based filtering

Usually, the noise filtration procedure is a part of the pre-processing stage of vectorization. We propose to use raster-to-vector conversion of input binary image for noise removal [**P2**]. We use the vector presentation to collect information about pattern structure at the neighbourhood of the pixel to be filtered. With this information we can smooth the borders of linear elements preserving details of other objects. The proposed approach is suitable for images that consist mostly of elongated linear objects (maps, drawings, schemes).

The main goal of this study is to improve the quality of binary image compression by noise filtering. The filtering reduces irregularities in the image caused by noise, and in this way, makes the image more compressible without degrading the image quality. The process of noise filtration consists of two stages: extracting of line features, and feature-based filtering.

In the first stage, global information is gathered from the image by extracting line features with vectorizing algorithm we introduced in publication **P1.** According to the vectorizing algorithm in use, the input binary image is skeletonized and then vectorized. The reference raster image is restored from the vector presentation by simple vector-to-raster conversion. In the second stage, the original image is processed for removing noise along the borders of extracted linear elements utilizing the original and the reference raster images.

The filtering is applied as part of a context-based image compression procedure. The compression remains near-lossless as only isolated pixels are eliminated. Experiments with test images show that from the compression point of view, the feature-based filtering with vectorization is twice as effective as traditional median filter, or a combination of three morphological filters: *opening*, *closing* and *annular filter* [Heijmans'94]. Comparison to other filtering algorithm that uses Hough Transform for vector feature extraction [Fränti-Ageenko-Kälviäinen-Kukkonen'98,'02], shown compression improvement of 19.2% for vectorization-based algorithm, and 12.7% against in comparison the HT-based approach (see Table 2.4.1).

Table 2.4.1: JBIG compression results [Ageenko'00] for images by feature-based filtering using Hough transform (HT) and the raster-to-vector conversion (RVC). The compression improvement is measured in comparison with the unfiltered image.

Input image	Original raster image (bytes)	Without Filtering (bytes)	HT-based Filtering (bytes)	RVC-based filtering (bytes)
Bolt	317,038	12,966	10,537	10,210
Power	512,199	17,609	16,271	14,581
Plan	484,561	5,098	4,319	3,978
TOTAL:	1,313,798	35,673	30,127	28,769
Improvement:	--	0.0%	12.7%	19.4%

Now let us consider question of time performance of this approach. Raster-to-vector conversion is time-consuming process involving a lot of image processing and

image analysis procedures. Processing time for large images is important issue in practical applications. In our case, due to fast implementation of algorithms we developed for the raster-to-vector conversion [**P1, P3**] the burden of the vectorization phase on the total processing time was reduced to 10%. In fact, the vectorization phase is up to 3-4 times faster than the JBIG compression procedure. In practice, the developed raster-to-vector conversion takes less 1% of the processing time for the HT-based algorithm [Fränti-Ageenko-Kälviäinen-Kukkonen'98,'02]. The compression ratio for the test binary images is also better for the vectorization-based method than for the HT-based approach.

2.4.3 Summary

The feature-based filtering technique removes additive noise form the original binary image and in this way, produces a better compression performance. Due to the developed of efficient methods for raster-to-vector conversion of large images the vectorization-based method outperforms Hough Transform based algorithm by quality as well by time performance.

3 Approximation

3.1 Introduction

3.1.1 Problem formulation

The general problem of approximation a given two-dimensional piecewise linear curve by another coarser one is of fundamental importance in computer graphics, vectorization tasks, vector map processing (see Figs. 3.1.3–3.1.8).

An open *N-vertex polygonal curve P* in 2-dimensional space is represented as the ordered set of vertices $P=\{p_1, \ldots, p_N\}=\{(x_1, y_1), \ldots, (x_N, y_N)\}$. The output coarser curve Q consists of $(M+1)$ vertices: $Q=\{q_1, \ldots, q_{M+1}\}$, where the set of vertices q_m is a subset of P and $M<N$. The end points of Q are the end points of P: $q_1 = p_1$, $q_{M+1} = p_N$. The approximation linear segment (q_m, q_{m+1}) of Q for curve segment $\{p_i, \ldots, p_j\}$ of P is defined by the end points p_i and p_j: $q_m = p_i$ and $q_{m+1} = p_j$. Thus, $(q_m, q_{m+1}) = (p_i, p_j)$.

As it was stated in [Kurozumi-Davis'82, Imai-Iri'88] there are two types of optimization problems connected with polygonal approximation problems:

Min-ε problem: Given a polygonal curve P, approximate it by another polygonal curve Q with a given number of line segments M so that the approximation error $E(P)$ is minimized.

Min-# problem: Given a polygonal curve P, approximate it by another polygonal curve Q with the minimum number of segments M so that the approximation error $E(P)$ does not exceed a given maximum tolerance ε.

3.1.2 Error measures

An approximation curve must satisfy some error criterion, which is specified appropriately for each application. In practice, the most of practical error measures in use are based on distance between vertices of the input curve and the approximation linear segments.

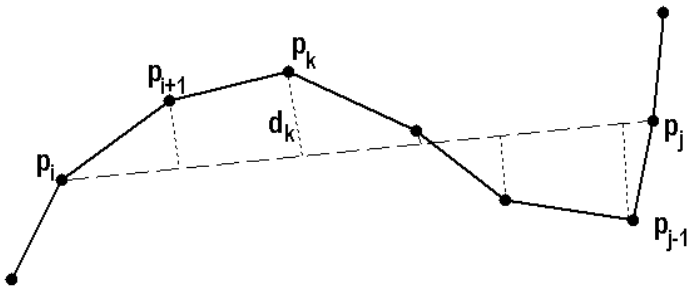


Figure 3.1.1: Distance d_k from the point p_k to the linear segment (p_i, p_j).

The distance $d_k(i, j)$ from curve vertex $p_k=(x_k, y_k)$ to the corresponding approximation linear segments (p_i, p_j) is defined as follows (see Fig. 3.1.1):

$$d(k;i,j) = \frac{| y_k - a_{i,j} x_k - b_{i,j} |}{\sqrt{1 + a_{i,j}^2}} \tag{3.1.1}$$

where the coefficients $a_{i,j}$ and $b_{i,j}$ are defined from the parameters of the linear segment (p_i, p_j):

$$a_{i,j} = (y_j - y_i)/(x_j - x_i)$$
$$b_{i,j} = y_i - a_{i,j} x_i . \tag{3.1.2}$$

The additive error measure L_p for curve segment $\{p_i, p_j\}$ is defined by the sum of distances $d(k; i, j)$ for all vertices in the segment as follows:

$$e_p(i, j) = \sum_{k=i+1}^{k=j-1} d^p(k; i, j) \tag{3.1.3}$$

For L_∞ the approximation error is defiend as the maximum deviation of input curves from approximation linear segment:

$$e_\infty(i, j) = \max_{i<k<j} \{d(k; i, j)\} \tag{3.1.4}$$

Approximation error $E_p(P)$ of the input curve P by Q with additive error measure L_p (where $p<\infty$) is defined as the sum of approximation errors for all segments:

$$E_p(P) = \sum_{m=1}^{M} e_p(i, j) \tag{3.1.5}$$

For error measure L_∞, the approximation error for the curve P is defined as the maximum

$$\Delta(P) = \max_{1 \le m \le M} \{e_\infty(i, j)\} \tag{3.1.6}$$

The error measure L_2 (integral square error, or ISE) is perhaps among the most widely used criteria for approximation *min-ε problem*. The error $e_2(p_i, p_j)$ with measure L_2 can be calculated in $O(1)$ time using stored arrays of coordinates cumulatives.

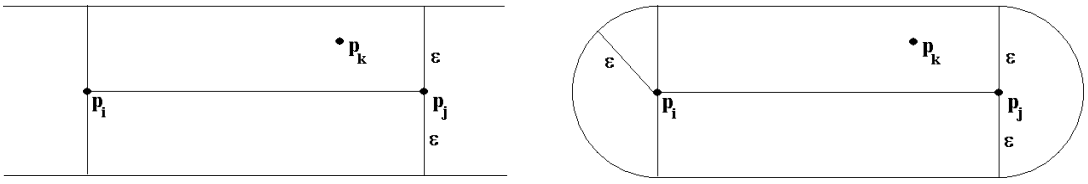


Figure 3.1.2. Error criteria with measure L_∞: the *parallel-strip* (or *infinite beam*) criterion (left), and the *segment distance* (or *tolerance zone*) criterion (right).

In most cases the error measure L_∞ is used in algorithms for *min-# problem* for practical reasons. In [Imai-Iri'88] different error criteria with measure L_∞ are given, including the most popular *parallel-strip* (or *infinite beam*) criterion [Toussaint'85], that is maximum distance between the line connecting p_i and p_j and the points of the curve segment $\{p_i, \ldots, p_j\}$ (see Fig.3.1.2, left). The *segment distance* (or *tolerance zone*) criterion is defined as maximum distance between the line segment (p_i, p_j) and the points of the curve segment $\{p_i, \ldots, p_j\}$ (see Fig. 3.1.2, right). In some algorithms, other error measures are in use for *min-# problem*: integral square error L_2 [Salotti'02, Schroeder-Laurent'02], and local integral square error (LISE) [Chung-Yan-Chen'02], or more complicated objective functions [Ray-Ray'93]. However, using error measure L_2 for *min-# problem* is not efficient in practice, especially in the case of multiple objects, because of its additive nature.

Since optimal algorithms are computationally expensive to be used (usually between $O(N^2)$ and $O(N^3)$), faster sub-optimal algorithms have been developed, often running in linear time. In order to evaluate the quality of sub-optimal algorithms, Rosin [Rosin'97] introduced two measures. *Fidelity* (F) measures how well the suboptimal polygon fits the curve relative to the optimal polygon in terms of the approximation error. *Efficiency* measures how compact the suboptimal polygonal presentation of the curve is. They are defined as follows:

$$Fidelity = \frac{E_{\min}}{E} \times 100$$

$$Efficiency = \frac{M_{\min}}{M} \times 100$$

where M is the number of segments for an algorithm under question, $M_{\min}$ is the number of segments for *min*-# solution, E is approximation error for an approximation algorithm, and $E_{\min}$ is the approximation error of the optimal solution for the *min*-ε *problem*.

3.1.3 Motivation

Important problems such as polygonal approximation have been explored very intensively for the last 30 years. The *min*-# and *min*-ε problems can be solved as an optimization task by dynamic programming or methods based on graph theory. Moreover, many heuristic algorithms have proposed. Optimal algorithms of complexity $O(N^2)$–$O(N^3)$ are very slow, whereas faster heuristic algorithms lack of optimality. Thus, development of *efficient* (fast and optimal) algorithms for large input data is still an open problem.

Introducing into practice more efficient algorithms for *min*-# problem we can reduce storage demands or transmission time for the same tolerance level. With more efficient algorithms for *min*-ε problem, we can reduce approximation error for the same amount of stored or transmitted data. Taking into account that nowadays polygonal approximation in vectorization, map service, CAD and GIS applications, the efficient solution of this problem is still of great practical importance (see Figs. 3.1.3-3.1.8).

At first we provide a short survey of heuristic algorithms for *min*-# and *min*-ε approximation problems in Section 3.2. Then we explore optimal algorithms for *min*-ε *problems* in Section 3.3. To bridge the gap between slow optimal and fast heuristic non-optimal algorithms we introduce paradigms of *bounding corridor* and *iterated reduced search* [**P4**]. Then we study optimal algorithms for *min*-# problem and present algorithm with joint using of different error measures based on the reduced search [**P5**]. Furthermore, we consider *min*-ε and *min*-# *problems* for the case of *closed contours* and provide cyclical DP algorithm with analysis of the state space [**P6**]. Finally, we extend the proposed iterative reduced search approach to the case of *min*-ε *problem* for *multiple objects* [**P7**].

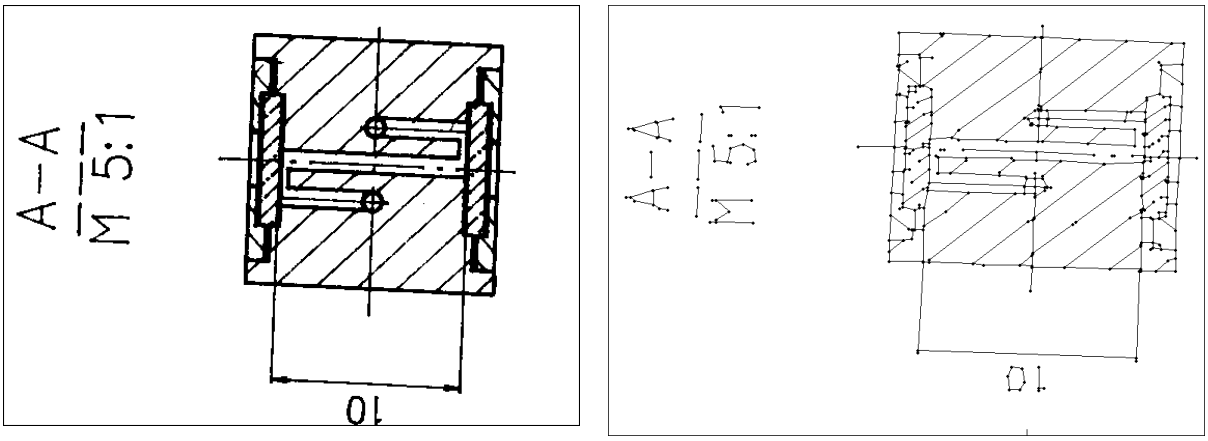


Figure 3.1.3: *Min*-# approximation of digitized curves (skeletons) for raster-to-vector conversion: input raster binary image (left); vector image for error tolerance ε=1.5 (right). Vectors are labeled by dots.

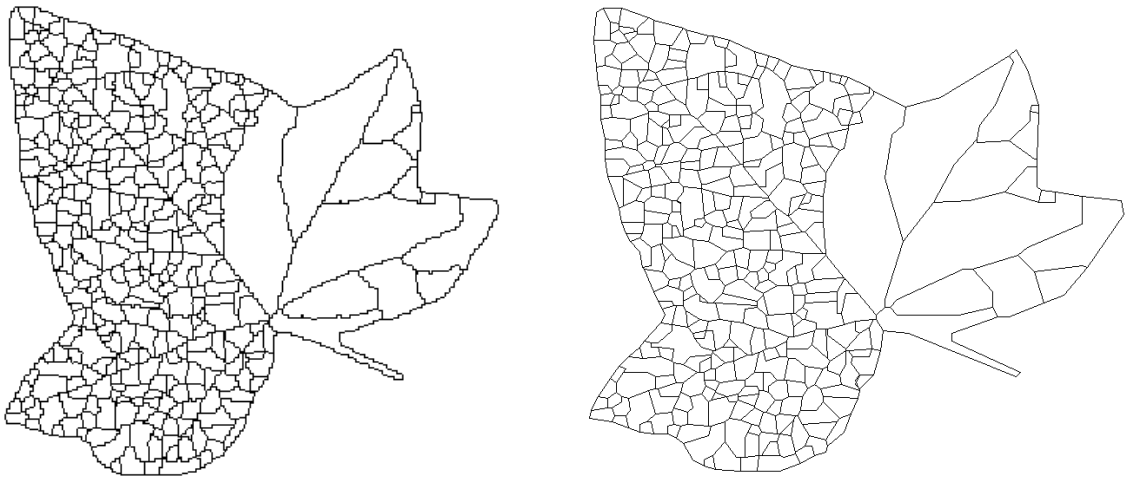


Figure 3.1.4: *Min*-# approximation of digitized curves: segmentation data (left); result of approximation for error tolerance ε=1.5 (right).

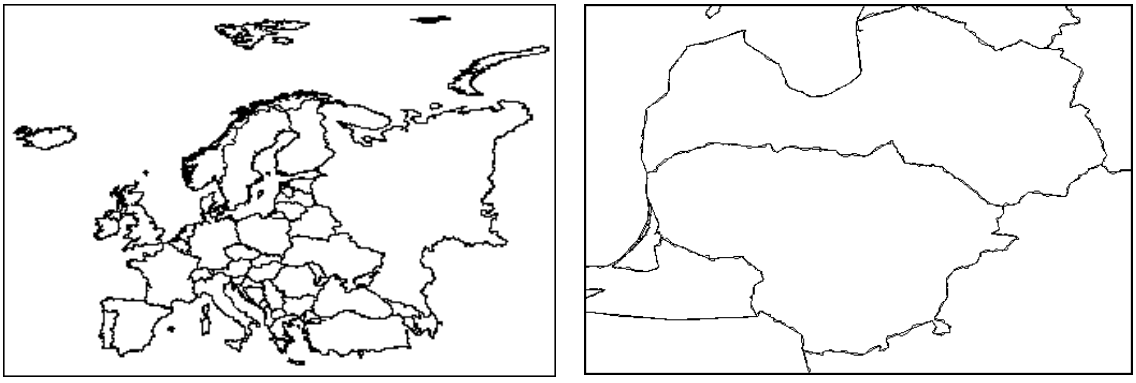


Figure 3.1.5: *Min*-ε approximation of multiple object vector data: input vector map of "Europe", 365 objects with 160,000 vertices (left); fragment of the vector map after 20:1 data reduction (right).

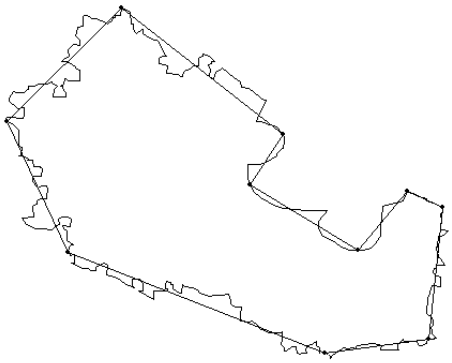


Figure 3.1.6: *Min*-ε approximation of simulated noisy 400-vertex curve by *M*=10 linear segments.

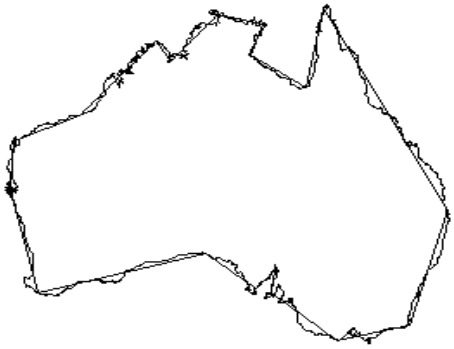


Figure 3.1.7. *Min*-ε approximation of closed 2900-vertex contour "Australia" by *M*=18 linear segments.

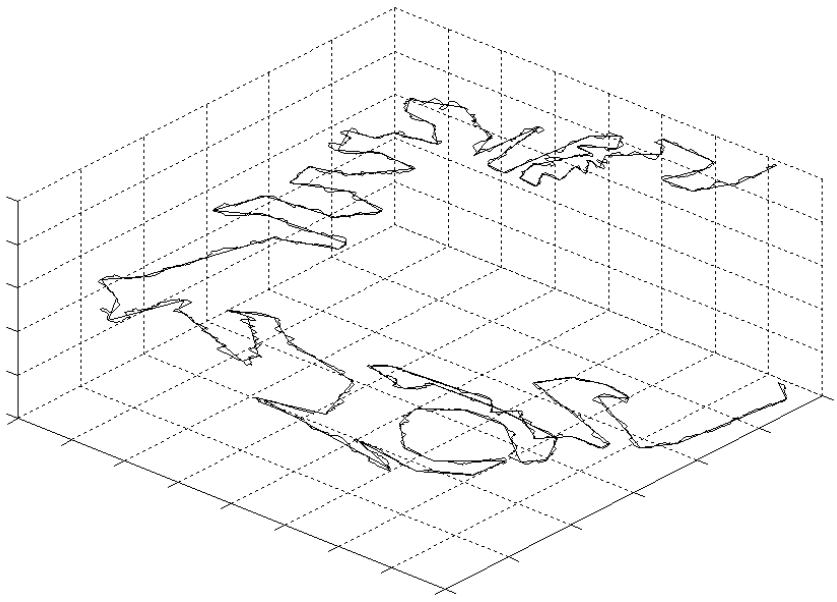


Figure 3.1.8: *Min*-ε approximation of 5000-vertex digitized path in 3D space by *M*=50 linear segments.

3.2 Heuristic algorithms

3.2.1 Survey of solutions

For the last 30 years many heuristic algorithms have been considered for approximation of polygonal curves. We can account about dozen of different heuristic approaches to the problem, and the number of algorithms exceeded one hundred items. In some extent, the existence of big amount and variety of the heuristic approximation algorithms can be explained by the variety of tasks, curves types, and error measures in use. But more likely the real reason for existence of numerous approximation methods is low fidelity and/or efficiency of the proposed heuristic algorithms [Rosin'97, Rosin'03], which leaves room for improvement.

Let us briefly consider the following approaches proposed for solving approximation problems. Some of the presented methods can be used for solving both problems, but some of the algorithms are designed for *min*-# or *min*-ε problem only. With algorithm for one of the problems (*min*-# or *min*-e) we can get solution for alternative problem in $O(\log N)$ steps of binary search [Chan-Chin'96].

a) Sequential tracing approach

The algorithms [Sklansky-Chasin-Hansen'72, Reumann-Witkam'74, Williams'78, Sklansky-Gonzales'80, Kurozumi-Davis'82, Gritzali-Papakonstantinou'83, Ishijama-Chin-Hostetter-Sklansky'83, Wall-Danielsson'84, Roberge'85, Cordella-Dettori'85, Aoyama-Kawagoe'91, Ray-Ray'94, Koski-Juhola'96, Zhao-Saalfeld'97] use a linear scan to evaluate error conditions, if the conditions are not satisfied a new segment search is started. The main problem of the methods is that the nodes sometimes do not correspond to the corners of the curve because a new vector is defined only when the criterion is violated.

b) Split method

The most widely used high-quality approximation algorithm is a heuristic method called the Douglas-Peucker algorithm [Douglas-Peucker'73]. It has been independently invented by many people [Ramer'72, Duda-Hart'73, Baumgart'74, Ballard'81]. The iterative procedure repeatedly splits the curve into smaller and smaller curves until the maximum of the perpendicular distances of the points on the curve from the line segment is smaller than the error tolerance ε. The complexity of the method is $O(N^2)$ in the worst case, and $O(N\log N)$ on average. The algorithm works in any dimension since it only depends on computing the distance between points and lines. The main disadvantage of this approach is the dependency on the starting point. It also suffers from stressing outliers, see more critics in [Visvalingam-Whyatt'90,'91]. Modified version of Douglas-Peucker algorithm was proposed [Hershberger-Snoyeink'92, Hershberger-Snoyeink'94] complexity of $O(N \log N)$ in the worst case, based on construction of convex hull of 2D point set. Later the result has been improved to $O(N \log^* N)$ [Hershberger-Snoyeink'98]. Unfortunately, the faster algorithm is not general, as it only works with simple 2-D planar curves, and not in higher dimensions.

c) Merge method

A common idea in some algorithms such as Sequential Tracing, Split, Dominant point detection, is to choose curve points to be vertices of the polygonal approximation. It can be done in opposite direction by choosing, at each stage, a curve point that will not be a vertex [Leu-Chen'88, Fahn-Wang-Lee'89, Wu-Leu'93, Boxer-Chang-Miller-Rau-Chaplin'93, Visvalingam-Whyatt'93, Ku-Chiu'95, Pikaz-Dinstein'95, Horst-Beichl'97, Latecki-Lakämper'99]. A reasonable choice is a point of which elimination will cause minimal increase in the approximation error. The procedure is halted when the desired number of linear segments M, or approximation error, is reached. Result is independent on the starting point. The complexity of the algorithm by Pikaz and Dinstein [Pikaz-Dinstein'95b] is $O(N \log N)$.

d) Split-and-Merge method

According to the technique, lines are fitted to an initial segmentation of the boundary and the least squares is computed. The procedure then iteratively splits a line if the error is too large and merges two points if the error is too small [Pavlidis-Horovitz'74, Ansari-Delp'91, Wu-Leou'93, Park-Jee'94, Held-Abe-Arcelli'94, Ray-Ray'95, Xiao-Zou-Yan'01]. This combines the split and merge methods with the same algorithm.

e) Dominant point detection

Attneave [Attneave'54] indicates that most shape information is contained in the corners (high curvature points), which are able to characterize the contour. To approximate curves using straight lines, high curvature points are the best place at which to break the lines. A large number of heuristic algorithms have been designed on the basis of the idea [Rosenfeld-Johnston'73, Rosenfeld-Weszka'75, Freeman-Davis'77, Davis'77, Sankar-Sharma'78, Cederberg'78, Anderson-Bezdek'84, Wall-Danielsson'84, Asada-Brady'86, Fishler-Bolles'86, Teh-Chin'89, Deguchi-Aoki'90, Ansari-Delp'91, Espelid-Jonassen'91, Wuescher-Boyer'91, Ray-Ray'92, Sarkar'93, Wu-Wang'93, Arcelli-Ramella'93, Zhu-Chirlian'95, Cornic'97, Hu-Yan'97, Garrido-Blanca-Garcia-Silvente'98, Inesta-Buendia-Sarti'98, Sanchiz-Inesta-Pla'98, Sanchiz-Pla-Inesta'98, Sohn-Kim-Alexander'98, Wu'9x, Beau-Singer'01, Neumann-Teisseron'02, Wu'03, Marji-Siy'03].

f) Relaxation labeling

According to the approach, the left and right slopes and curvature value is measured at every point of the input digital curve. Each point on the curve is associated with an attribute list containing slopes and curvature at the point. The attributes will determine the initial *probability* of the point p_i being a 'side, and probability of being an 'angle'. The relaxation process will change the probabilities. As the relaxation process is iterated, certain points became more certain that they are 'angles', while other points became more certain that they are 'side'. Finally, the probabilities converge to some values [Davis-Rosenfeld'77, Davis-Rosenfeld'78, Rutkowski-Peleg-Rosenfeld'81, Mikheev-Vincent-Faber'01].

g) *K*-means method

Phillips and Rosenfeld [Phillips-Rosenfeld'88] proposed *k*-means based clustering method to partition the contours into subsets of points such that each subset can be fitted by a straight line. They started with an initial partition and then evaluate the principal axis. In [Yin'98a] three algorithms have been proposed based on Phillips and Rosenfeld approach [Phillips-Rosenfeld'88] for *min*-ε *problem* with two error measures (L_2 and L_∞). A simpler line fitting method was used instead of the principal axis approach.

i) Genetic (evolutional) algorithms

Genetic algorithms [Yin'98b, Huang-Sun'99, Yin'99, Recatalá-Iñesta'99, van Dijk-Thierens-de Berg'00, Yin'00, Sun-Huang'00, Traver-Recatalá-Iñesta'00, Ho-Chen'01, Zhang-Guo'01] are based on stochastic search, which simulates the biological model of evolution [Goldberg'89, Michalewicz'92]. A population is set of chromosomes and an initial one is randomly generated. During each generation, the fitness (approximation error) of each chromosome is evaluated, and chromosomes are selected for reproduction based on their fitness values. Selection operator reproduces some chromosomes with smaller approximation error and eliminates chromosomes with bad approximation. The selected solutions the undergo reproduction under action of the crossover and mutation operations.

j) Ant colony optimization method

Ant Colony Optimization (ACO) is an optimization paradigm that mimics the exploration strategy of a colony of ants [Dorigo'92]: ants can construct the shortest path from their colony to the feeding source and back using pheromone trails. An ant leaves some quantities of pheromone on the ground and marks the path by a trail of this substances. The next ant choose path depending on the amount of pheromone on it and leaves its own pheromone. Vallone [Vallone'02] considered *min*-ε *problem* with maximum perimeter of polygon as an optimization criteria. Yin [Yin'03] considered *min*-# *problem* with error metrics L_2, including approximation of closed contours.

k) Tabu-search

Tabu-search, developed by Glover [Glover-Laguna'97], is one of the meta-heuristic methods that can be used to solve combinatorial optimization problems. It is different from the local search in the sense that tabu-search allows moves to a new solution, which makes the objective function worse in hope that it will achieve a better solution in a longer term [Yin'00, Zhang-Guo'01].

l) Vertex adjustment method

The main idea of the *vertex adjustment method* is to improve preliminary result by a local search [Chen-Ventura-Wu'96, Katsagellos-Kondi-Meier-Osterman-Schuster,'98, Lee-Chung-Kim'99, Chung-Lee-Moon-Kim'00, Nicolas'01, Yun-Lee-Kim'01, Horng'02, Neumann-Teisseron'02]. The local search can be applied to vertices successively (vertex-by-vertex) [Katsagellos-Kondi-Meier-Osterman-Schuster'98, Chung-Lee-Moon-Kim'00, Nicolas'01], or simultaneously to all vertices using optimization technique [Chen-Ventura-Wu'96, Lee-Chung-Kim'99, Horng'02]. Solution of the optimization task can be found by Viterbi algorithm for the shortest

path in a graph [Lee-Chung-Kim'99] or by dynamic programming [Chen-Ventura-Wu'96, Horng'02, Neumann-Teisseron'02].

In [Lee-Chung-Kim'99], the adjustment of approximation points is performed by Viterbi algorithm for the shortest path in graph. The search is performed among 4-neigbours of the initial approximation points. This technique, however, can be extended to the case where approximation points belong the input curve only.

Chen *et al.* [Chen-Ventura-Wu'96] studied *min*-ε *problem* for approximation by circular arcs and line segments they introduced modification to the dynamic programming algorithm for the problem in question. To reduce processing time, they search all possible combinations of the points that are within a given range with respect to an initial set of detected break points, instead of performing complete enumeration. The initial set of break points, $\{u_i,\ i=1, 2, ..., N\}$, is the solution obtained with heuristic algorithm for dominant (break) point detection [Chen-Ventura-Wu'96]. The possible sets of solutions are generated by varying each u_i within a range of (u_{i-h}, u_{i+h}), where h=3. Although this method has been proposed for approximation by circular arcs and lines, the approach can be applied in the case of piecewise linear approximation as well.

In [Lee-Chung-Kim'99] the adjustment of approximation points is performed by Viterbi algorithm for the shortest path in graph. The search is performed among 4-neigbours of the initial approximation points. However this technique can be extended to the case when approximation points belong to the input curve only.

Neumann and Teisseron [Neumann-Teisseron'02] and Horng [Horng'02] used the same approach as Chen *et al*. [Chen-Ventura-Wu'96] to reduce the processing time of dynamic programming algorithm. They performed search of the optimal location of approximation vertices within given range around the current position. Neumann and Teisseron defined the window where the dominant point can be located. Horng defined the window size as 1/3 of the number of the vertices in the curve segment between two correspondent dominant points. In both cases, the set of the initial approximation points $\{q_1, \ldots, q_{M+1}\}$ is defined by algorithms for the dominant points detection.

3.2.2 Summary

The presented heuristic approaches can be divided into two classes from optimality point of view:

1) Classical algorithms, namely sequential algorithms, split, merge, split-and-merge, dominant points detection, relaxation labeling;
2) Optimization algorithms: *K*-means, tabu search, genetic algorithms, ant colony optimization methods, and vertex adjustment methods.

The classical algorithms are mostly based on heuristic methods or approaches. The fidelity of these algorithms is usually low [Rosin'97] because the global optimization error is not subject to control during the process of the approximation. Some heuristic algorithms can be used as part of optimal algorithms. For example, the cone-intersection method was used in graph theory based algorithms to reduce the complexity of the graph construction procedure. Sallotti [Sallotti'00,'01] applied heuristic (Split) algorithm of Pavlidis [Pavlidis'77] to estimate upper bound for approximation error to reduce A*-search in graph.

In modern algorithms the approximation problem is considered as optimization task where the global approximation error is the main criterion to be controlled. The search of solution that provides minimal approximation error can be performed by *stochastic optimizaton methods* (as genetic algorithms and ant colony method) or by *local optimization methods* (as tabu search and vertex adjustment methods). The initial solution for starting the search can be obtained with some heuristic algorithm for approximation, or any random approximation can be used. Then the initial approximation (or approximations) is improved to find minimum of the global approximation error. Algorithms of this class can provide near-optimal or sometimes optimal results, but the global optimality cannot be guaranteed even in the case of iterative approaches.

3.3 Optimal algorithms for *min-ε problem*

The min-ε problem is formulated as follows: given polygonal curve P, approximate it by another polygonal curve Q with a given number of segments the minimum number of segments M so that the approximation error is minimal.

3.3.1. Early history of the subject

The early history of *min-ε problem* was begun from the problem of L_2-optimal approximation for continuous one-variable function. In 1961 Stone [Stone'61] considered piecewise-linear curve fitting as a formal optimization problem. The objective was to minimize the squared approximation error subject to a constraint on the number of linear segments. Bellman followed with a solution [Bellman'61] based on his principle of optimality [Bellman'57]. Later Gluss [Gluss'62] expanded upon Bellman's work. Lawson [Lawson'64] used dynamic programming (DP) to establish the existence of a balanced error property. Cox discussed a similar approach in his paper [Cox'71]. Cantoni [Cantoni'71] determined the optimal polygon of a known nonlinear function by minimizing the weighted integral square errors. The works performed by Bellman, Gluss, Stone, Cantoni and Cox hold only for a 1-D signal whose analytic forms are known. Nevertheless, it inspired other researchers to use the dynamic programming approach for optimal approximation of digital curves [Perez-Vidal'94, Chen-Ventura-Wu'96].

3.3.2 Dynamic programming algorithm for *min-ε problem*

In 1994 Perez and Vidal published the first optimal algorithm for *min-ε problem* for digital curves [Perez-Vidal'94]. The proposed algorithm was based on dynamic programming method for solving optimization task. The authors extended DP approach for approximation of 1D continuous functions [Bellman'57, Gluss'62] to the case of digital 1-D waveforms and 2-D planar curves. They wrote that the approach can be extended on the case of 3-D space as well with corresponding approximation error function. To illustrate their method Perez and Vidal presented solutions for error measures L_1 and L_2. The complexity of optimal algorithm with error measure L_1 is $O(NM^3)$, but complexity of the algorithm for error L_2-norm was reduced to $O(NM^2)$ by using the incremental scheme for computation of approximation error. Authors also considered approximation of closed contours and pointed out that at most $(N\text{-}M)$ runs of the basic algorithms are necessary to find globally optimal solution. They also mentioned cyclical DP as a possible way for solving approximation problem for closed contours.

In 1996, Chen *et al.* [Chen-Ventura-Wu'96] studied *min-ε problem* with measures L_2 and L_∞ for approximation of 2-D planar curves by circular arcs and line segments they proposed algorithm, which was also inspired by DP method of Bellman [Bellman'57, Gluss'62] for 1-D continuous functions.

Heckbert and Garland published a survey in 1997 [Heckbert-Garland'97], and they wrote about optimal approximation of digital 1-D digital waveform $f(x)$ with minimum error: "*the L_2-optimal approximation to a function f(x) can be found in*

$O(MN^2)$ time, worst case, using dynamic programming", but they did not provide any details of the algorithm or references.

Haugland *et al*. [Haugland-Heber-Husøy'97a, Haugland-Heber-Husøy'97b, Heber-Haugland-Husøy'97] represented optimal algorithm for *min*-ε *problem* for 1-D waveforms (ECG) with error measure L_2. In contrast to Perez-Vidal algorithm, the proposed solution was based on algorithm for resource-constrained shortest path in graph, introduced by Saigal [Saigal'68] and later corrected by Rosseel [Rosseel'68]. Nevertheless, from computational point of view the suggested DP-based algorithm for cardinality constrained shortest path (CCSP) is equivalent to Perez-Vidal algorithm for 1-D signals. The CCSP approximation algorithm was used for lossy compression of ECG signals with a given compression ratio [Nygaard-Melnikov-Katsaggelos'99, Nygaard'00, Nygaard-Melnikov-Katsaggelos'01]. Furthermore, the CCSP algorithm was extended to the case of planar curves [Nygaard-Husøy-Haugland'98, Nygaard'00]. This DP algorithm is also equivalent to solution originally given by Perez and Vidal [Perez-Vidal'94] for 2-D planar curves.

Tseng *et al*. [Tseng-Juan-Chang-Lin'98] presented DP algorithm for optimal approximation with a given error tolerance (*min-# problem*). Three error measures were used, including L_1, L_∞ and a length cost function. The essence of the algorithm is the same as that of Perez and Vidal. The only difference is the stop rule: in Perez-Vidal algorithm the DP search in the state space is continued until a given number M of segments is reached, in the algorithm of Tseng *et al*. the search is performed until the current approximation error is less than a given error tolerance ε.

Mori *et al*. [Mori-Wada-Toraich'99] proposed DP algorithm for optimal approximation of curves by linear segments, circular arcs and splines. The approach is the same as that of Perez and Vidal. The DP approach of Perez and Vidal was used for approximation of input curve using circular arcs [Pei-Horng'96], circular arcs and line segments [Horng-Li'01]. In [Chung-Yan-Chen'02] the Perez-Vidal algorithm was extended to the case of polygonal approximation in 3-D space with local integral square error (LISE) measure.

The main contribution to the reduction of the complexity of Perez-Vidal algorithm has been done by Salotti [Salotti'00, Salotti'01]. The main idea behind the algorithm is to stop the search as soon as possible using heuristic functions to estimate the cost function. The algorithm has been implemented in two versions: A*-search [Salotti'00, Salotti'01] and dynamic programming [Salotti'02a]. At first, the rough approximation with Pavlidis algorithm [Pavlidis'77] is performed to find approximation error to be used further as upper bound for approximation error γ. Then A* or DP search is performed. If the estimated cost function for the current vertex is bigger than the upper bound γ, the next vertices located further along the curve P are not examined as possible candidates as approximation points of Q. Salotti offered two methods for estimation of the remaining cost function from the current vertex to the goal vertex. The use of heuristics in the algorithms makes it difficult to estimate the complexity. Experiments provided with test shapes have shown that the complexity of the algorithm is close to $O(N^2)$.

The paper of Perez-Vidal [Perez-Vidal'94] is the key publication for *min*-ε *problem* for digital curves because it was the first publication where the optimal algorithm for the problem has been proposed. Moreover, it was the first paper, which contained deep analysis of all the questions concerning the *min*-ε *problem*. In more

recent publications [Haugland-Heber-Husøy'97a, 97b, Heber-Haugland-Husøy'97, Nygaard-Husøy-Haugland'98, Tseng-Juan-Chang-Lin'98, Mori-Wada-Toraichi'99] the algorithm of Perez and Vidal has been rediscovered.

Min-ε *problem* for the closed curves with maximum area or perimeter cost function can be solved in $O(MN)$ time with SMAWK algorithm [Agarwal *et al.*'89]. For error metrics L_∞ the *min*-ε problem for the convex curves can be solved in $O(N^2)$ time with algorithm of Chan-Chin [Chan-Chin'96], based on the search of the shortest path in directed graph. They offered to take advantage of the convexity of the input curve P to construct the graph $G(P)$ on the vertices of the P in $O(N^2)$ time, and to find the *min*-ε approximation with an additional $O(MN)$ time.

3.3.3 Full search dynamic programming algorithm

For error measure L_2 the error of the approximation of segment $\{p_i, ..., p_j\}$ by the line segment (q_m, q_{m+1}) of Q is defined as the sum of squared Euclidean distances from each vertex of $\{p_i, ..., p_j\}$ to the corresponding line segment (q_m, q_{m+1}) (see Eq. 3.1.3). The approximation error $E_2(P)$ of the curve P by the curve Q is the sum of the errors of approximating each segment $\{p_i, ..., p_j\}$ of P by the corresponding line segment (q_m, q_{m+1}) of Q. The optimal approximation of curve P is then the set of vertices $\{q_2, ..., q_M\}$ of Q that minimizes the approximation error $E_2(P)$:

$$E_2(P) = \min_{\{q_m\}} \sum_{m=1}^{M} e^2(q_m, q_{m+1}) . \tag{3.3.1}$$

The optimization problem can be solved by the dynamic programming algorithm as proposed by Perez and Vidal [Perez-Vidal'94] with the following recursive expressions (see Fig. 3.3.1):

$$D(n,m) = \min_{m-1 \le j < n} \{D(j, m-1) + e^2(p_j, p_n)\}, \quad m = 1,...,M; \tag{3.3.2}$$

$$A(n, m) = \arg\min_{m-1 \le j < n} \{D(j, m-1) + e^2(p_j, p_n)\}, \quad n = m,...,N .$$

Here $A(n, m)$ is the *parent state* that provides the minimum value for the cost function $D(n, m)$ at the state (n, m) of the state space Ω (see Figs. 3.3.1). Approximation vertices q_m and q_{m+1} of Q are vertices p_i and p_j of the correspondent line segment. The general scheme of the full search Perez and Vidal's algorithm is presented on Fig. 3.3.2.

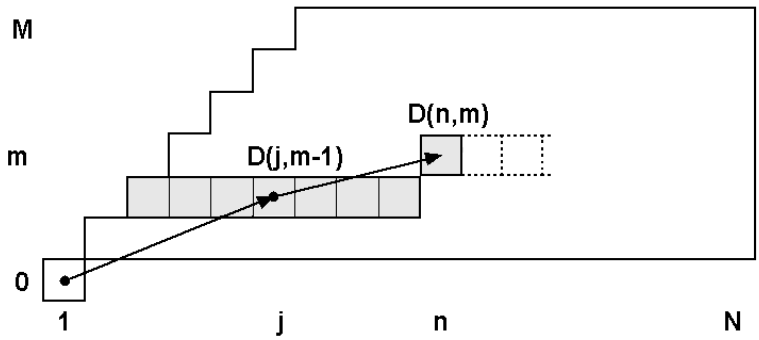


Figure 3.3.1: Scheme of computation of the cost function $D(n, m)$ in the state space Ω by dynamic programming algorithm of Perez and Vidal.

```
// Initialization
D(1,0) ←0
FOR  n = 2 TO  N DO
    D(n,0) ←∞
END

// Minimum search
FOR  m = 1 TO  M  DO
    FOR n = m TO  N  DO
       dmin ← ∞
        FOR  j= m–1 TO  n–1  DO
            d ← D(j, m–1) + e2(i,j)
            IF(d < dmin)
               dmin ← d;
               jmin ← j
            ENDIF
        END
        D(n, m) ← dmin
        A(n, m) ← jmin
    END
END

// Backtracking for the solution H(m)
H(M)= N
FOR m = M  TO  1 DO
   H(m–1) = A(H(m), m))
END
E2(P) ← D(N,M)
```

Figure 3.3.2: General scheme of full search dynamic programming algorithm of Perez and Vidal.

3.3.4 Time and space complexity of Perez-Vidal algorithm

As it was mentioned above, time complexity of L_2-optimal Perez-Vidal algorithm is $O(MN^2)$. Let us consider problem of approximation in a trivial case with M=N-2. Perez and Vidal proposed the algorithm for approximation of digitized curves, where normally $M << N$, but generally speaking, in GIS applications (vector map data reduction) any number of approximation segments can be given, including $M \approx N$.

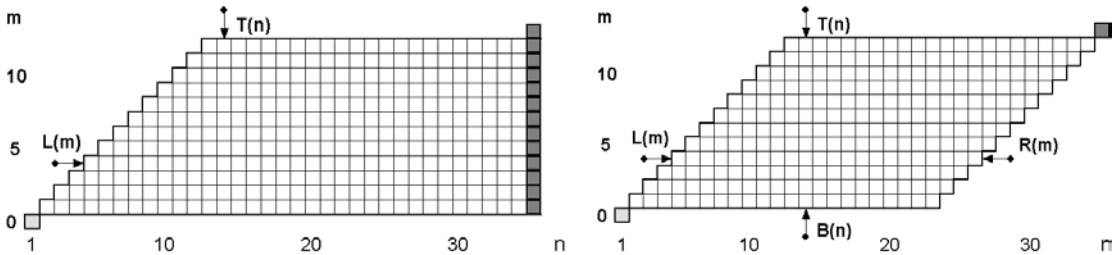


Figure 3.3.3: The original state space Ω in algorithm Perez and Vidal's algorithm (left); the modified *single-goal* state space Ω for *min*-ε *problem* in the proposed algorithm [**P4**] (right).

In fact, to obtain approximation with N-2 line segments we have to eliminate only one vertex from the input curve P (for example, with one step of Merge algorithm of Pikaz and Dinstein [Pikaz-Dinstein'95]). So, the best solution can be found in linear time by checking approximation error for every vertex as the eliminated one. On the other hand, complexity of the optimal algorithm for $M = N$-2 is given as $O((N\text{-}2)N^2))$=

$O(N^3)$ time. The reason for the high complexity of the algorithm is the redundancy of the search: in the case under consideration we are constructing *all* solutions for $m=1, 2, .., M\text{-}2$, although in fact we need to know solution for single goal state $\Omega(N, M\text{-}2)$.

To solve the paradox, we proposed in [**P4**] a modification of the state space. The state space has to be bounded to eliminate states that are not necessary for the construction of the goal state (see Fig. 3.3.3, right). The complexity of the dynamic programming algorithm with the modified state space is $O(M(N\text{-}M)^2)$. As we can see, the time complexity of the modified DP algorithm for the trivial case in question is $O(N)$, as it should be (see Fig. 3.3.4).

In same cases, the input curve can be approximated with zero error by less number linear segments as a given M. In algorithm of Perez and Vidal such situation can be detected and the computation can be stopped even if the current number of segments m is less than M. In full search algorithm in modified state space algorithm, to be sure that a given number corresponds non-zero approximation error, we can check it with a simple procedure in $O(N)$ time by elimination of those points, whose absence does not affect on the total approximation error. If the found number $M_{\min}$ is bigger than a given number of linear segments M we can find approximation solution for this M, otherwise we can approximate the input curve by smaller number of segments $M_{\min}$ with zero approximation error.

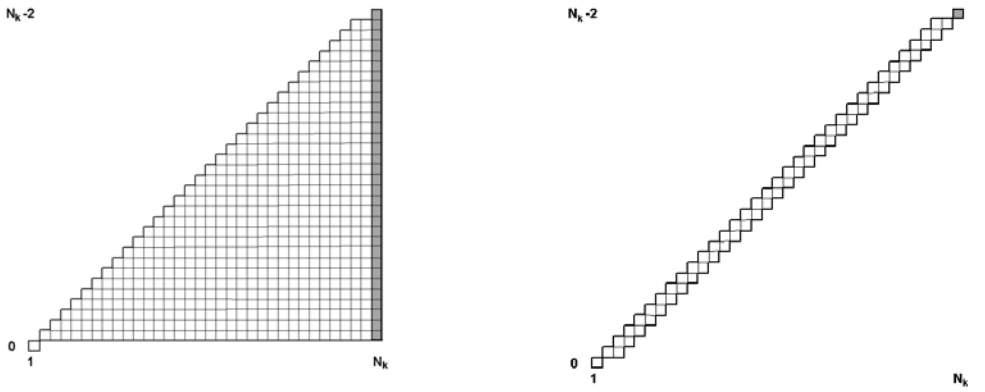


Figure 3.3.4: State space Ω for the case $N = M–2$: in Perez-Vidal algorithm (left), and in the proposed algorithm [**P4**] (right).

3.3.5 Using the preliminary computed error values

Haugland *et al*. [Haugland *et al*.'97a, 97b] and Horng and Li in [Horng-Li'02] offered to calculate approximation errors for all pairs of vertices in advance and store it in a 2-D array of $N \times N$ size. The reasons of this technique were different: Haugland *et al*. treat the *min*-ε *problem* as the search of the cardinality constrained shortest path in graph and all the weights in graph should be known prior the computations. The purpose of Horng and Li was to reduce processing time by performing all computations only one time and to use the stored values later.

The last idea seems to be reasonable, because it really permits to avoid recalculation of the values and reduce processing time. In practice, however, this method in the form suggested by Horng and Li works for relatively small N only because of high *space-complexity* of the method. Let us consider the following example of DP approximation: the 5004-vertex test shape #3 [Salotti'01] is to be

approximated by 50 linear segments. With algorithm of Perez and Vidal the approximation can be computed in 710 s [Salotti'01]. According to the mentioned above approaches [Haugland *et al*.'97a, 97b, Horng-Li'02] the error values have to be stored in 2-D array of $N \times N$ size; it gives 200 Mb for the N=5000. Moreover, for approximation of closed contours Horng and Lee proposed to use 2-D array of $2N \times 2N$ size that means allocation of 800 Mb in the case under question. When RAM size is limiting resource, demands for allocation of 200 Mb (or even 800 Mb) can be a problematic way to reduce processing time.

Thus, for small N the precalculation method [Haugland *et al*.'97a,'97b, Horng-Li'02] does work, but only in the case when the processing time is small even without using the technique. For large N, however, the precalculation method cannot work efficiently because of high space complexity $O(N^2)$.

In [**P4**], we have proposed modification of DP algorithm of Perez and Vidal with precalculation by total cost of $O(MN)$ space instead of $O(N^2)$ as in [Haugland *et al*.'97a,'97b, Horng-Li'02]. Actually this space complexity is the same as that of the original algorithm, namely $O(MN)$.

Let us consider memory demands of the original Perez-Vidal algorithm in details. According to the scheme represented in the algorithm, the DP calculations are performed sequentially for all m starting from 1 to M. To support the calculation we need 2-D array of $M \times N$ size to store parent states $A(m, n)$ for the optimal sub-paths. For storing the cost function $D(m, n)$ it is enough to store 2-D table of $2 \times N$ size. This is because we need to know only the previous row to calculate the current one. We also need 2-D table of $5 \times N$ size for five arrays of cumulatives of coordinates to calculate approximation error *on-fly*. The total space complexity of Perez-Vidal algorithm is $O(MN)$.

According to the approach of Horng and Li [Horng-Li'02] with storing of precalculated errors we need an additional 2D array of $N \times N$ size for the errors, which increases the total space complexity to $O(N^2)$. In addition, in [Haugland *et al*.'97a, 97b, Horng-Li'02] for the cost function $D(m, n)$ instead of $2 \times N$ array was used 2-D array of $M \times N$ size.

Now let us change the order of processing from *row-by-row* to *column-by-column*. We will fill the state space with solutions of sub-problem *column-by-column*, at first for n=2, then for n=3, until we reach the last vertex $n=N$. For every current vertex n, we have to find solutions of the sub-problems for all allowable values of m in the state space. For this computational scheme we need 2-D array for $A(m, n)$ of size $M \times N$ and five arrays of cumulatives of coordinates as in the original algorithm. Now we need 2-D array of size $M \times N$ for the cost function $D(m, n)$, not $2 \times N$ as in the previous case. Now all calculations are performed in vertex-by-vertex mode. For the processing of the current vertex, we need approximation errors for linear segments from the *current vertex* to the vertices already passed. We do not need anymore the approximation errors from all vertices to all ones for processing the current state. For this purpose it is enough 1-D array of size $1 \times N$. To support the storing pre-calculated cost function values, we have to use $O(MN)$ memory for the cost function $D(m, n)$.

Table 3.3.1. Space demands for DP algorithms for *min*-ε *problem* with precalculation.

	Full search [1)]	Full search [**P4**]	Reduced search [**P4**]
Cost function D	$O(MN)$	$O(MN)$	$O(WN)$
Parent states A	$O(MN)$	$O(MN)$	$O(WN)$
Cumulatives	$O(N)$	$O(N)$	$O(N)$
Errors $e^2(p_i,p_j)$	$O(\boldsymbol{N^2})$	$O(N)$	$O(N)$
Total space	$O(\boldsymbol{N^2})$	$O(MN)$	$O(WN)$

[1)] [Haugland *et al*.'97a,'97b, Horng-Li'02]

Comparing space demands for two DP algorithms with full search (see Table 3.3.1), we can use precalculation method at the cost of $O(MN)$ additional memory instead of $O(N^2)$ as suggested in [Haugland *et al*.'97a,'97b, Horng-Li'02]. For the considered above example of approximation of 5000-vertex curve by M=50 segments with the proposed scheme in [**P4**] we need additionally about 1 Mb, that is only 1% of amount we must allocate with the approach of Horng and Li.

3.3.6 Summary

Perez and Vidal have proposed optimal DP-based algorithm which can solve the *min*-ε *problem* with L_2 error metrics in $O(MN^2)$ time. We introduced two improvements to the core algorithm, including modification of the state space to reduce complexity of the algorithm to $O(M(N\text{-}M)^2)$, and a scheme for the reducing processing time by storing of pre-calculated approximation errors.

3.4 Iterative reduced search

3.4.1 Optimality versus time performance

Heuristic algorithms are fast, but they cannot provide optimal solution. Optimal algorithms of complexity $O(N^2)$–$O(N^3)$ are too slow to be used for large input. As it was already noticed by Heckbert and Garland who wrote [Heckbert-Garland'97]: "*Optimal simplification typically has quadratic or cubic cost, making it impractical for large inputs*". Zhang and Guo [Zhang-Guo'01] also wrote that, in practice, the vertices number of a curve which DP or other exact optimal methods can tackle is about a 100 points. They have not provided information concerning processing time for test shapes they used. Nevertheless, the note is symptomatic: time complexity of optimal algorithms is high. Understanding that fact, the authors of optimal methods have also proposed a number of approaches to reduce processing time at the cost of optimality.

Perez and Vidal [Perez-Vidal'94] in their concluding remarks have mentioned two possible techniques for obtaining *controlled* reduction in computational cost. The one is beam search, which essentially consists of discarding those branches that lead to an error greater than a certain margin at any given stage. The other method is to apply adjustment window that limits the maximum and minimum number of points assigned to any given edge.

Haugland *et al*. [Haugland-Heber-Husøy'97a] motivates the development of optimal algorithms that the optimal algorithms can serve as a powerful tool when analyzing possible heuristic algorithms. He proposes the following approaches for further studies: a) divide the samples into K parts and perform processing on each of these; the execution time is reduced by about K^2; b) start with the path involving the first and last vertices only, and augment it gradually by one vertex until an M-vertex path is achieved; c) start with an arbitrary path with M segments and change the vertices on the path successively so that the path length is gradually reduced. Haugland *et al*. developed algorithm for approximation of ECG signals. In the case of quasi-periodical ECG signals, this approach is quite satisfactory, but in the case of arbitrary planar (or space) curve it can cause significant lost of quality.

Mori *et al*. [Mori-Wada-Toraich'99] presented algorithm for optimal approximation of curves by linear segments, arcs and splines proposed 5:1 decimation of curve vertices to reduce processing time. Schroeder and Laurent [Schroeder-Laurent'99] proposed following two-step scheme for *min-# problem*: reduce the number of vertices using the approximation algorithm with bigger value of error tolerance $\rho\varepsilon$, where $\rho>1$, and apply the approximation algorithm to the obtained curve with a given error tolerance ε.

Salotti [Salotti'00,'01] used preliminary approximation of the input to get upper limit for the approximation error to be used further in the process of the state space exploration. He also introduced two heuristic functions for cost function estimation to reduce the search. His A*-search algorithm is optimal but the complexity is still $O(N^2)$ even in the best case.

3.4.2 Paradigm of bounding corridor

Optimal algorithms are slow, whereas heuristic algorithms are fast but they lack the optimality. There have been attempts to improve the quality of heuristic algorithms by using local search or stochastic optimization techniques (see vertex-adjustment methods), but these methods cannot guarantee optimality or high fidelity. On the other hand, there have been attempts to reduce the processing time of DP algorithms by cost of optimality.

Polygonal approximation which is very close to the optimal one, is quite enough in most practical applications, because the real accuracy of output polygonal approximation is defined by the following factors: a) fidelity of the approximation algorithm, and b) accuracy of input vertex data (digitized curves, vector map). The quest for 100% fidelity is justified from mathematical point of view, but this demand can be released in practical applications if we take into consideration all technical details of the application and the time cost of the optimal result.

In **P4**, we try to bridge the gap between the optimal and heuristic algorithms by introducing a new paradigm of *bounding corridor* in the state space. Instead of time-consuming search in the full state space (see Fig. 3.4.1), we offer to perform the search only in the most relevant part of it, bounded by a corridor (see Fig. 3.4.2). The corridor of fixed width *W* in the state space is constructed along a *reference path*, which can be obtained by any fast heuristic algorithm.

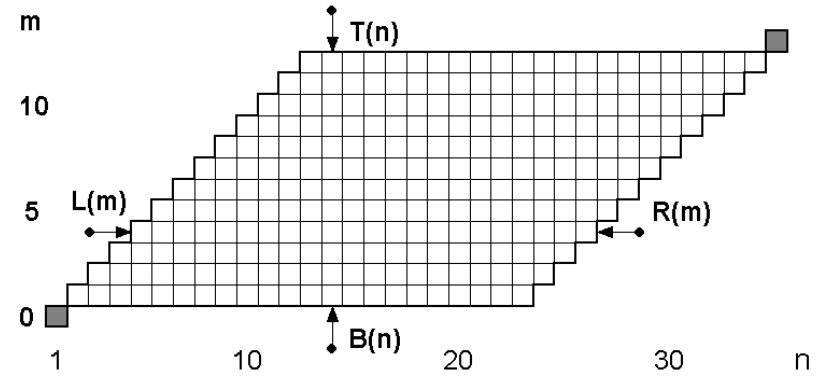


Figure 3.4.1: Illustration of the modified single-goal state space Ω for a sample problem size of *N*=35, *M*=13. The start and goal states are marked with the gray squares.

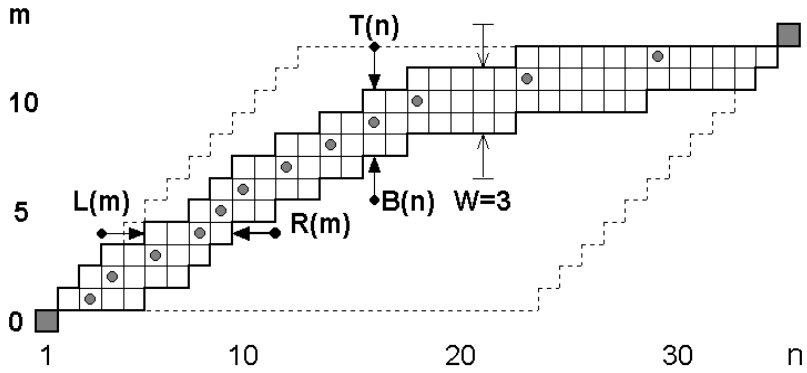


Figure 3.4.2: Illustration of the bounding corridor of width *W*=3 in the state space Ω. The reference path *H* is marked by the gray circles; the single-goal state space Ω is marked by the dashed line.

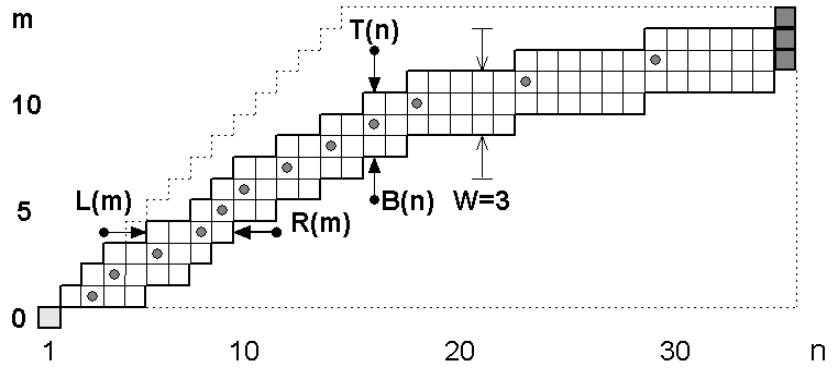


Figure 3.4.3: Illustration of the *multiple-goal* bounding corridor of width W=3 in the state space Ω. The reference path H is marked by the gray circles; the multiple-goal state space Ω is marked by the dashed line.

In contrast to constraints on approximation error [Salotti'00,'01], we use geometrical constraints on search area in the state space to control the breadth of the search. Width of the corridor W at some approximation point defines the range of possible numbers of approximation segments for the point. At the same time, location of the approximation points is optimized too. In the mentioned algorithms with local adjustment of approximation points the number of segments assigned to curve part from the first vertex to approximation point is the same, only the location of the approximation point can be changed within a narrow range.

The optimization algorithms [Chen-Ventura-Wu'96, Lee-Chung-Kim'99, Neumann-Teisseron'02, Horng'02] can be formally considered as a special case of the search in narrow non-continuous. On the other hand, the reduced search algorithm with reference approximation can be treated as optimization of the preliminary (reference) solution. The main difference of the reduced search algorithm from the vertex adjustment method is not only *quantitative* (wider range of the search), but it is *qualitative* one too. The introduced paradigm of bounding corridor allows us to solve a number of approximation problems, which cannot be solved by mentioned above vertex adjustment and search reduction methods because of the following reasons.

a) The proposed reduced search algorithm can be iterated using the output solution as a new reference path in the next iteration. The number of iterations can be given in advance, or adaptively varied depending on the development of the approximation error. With the iterative reduce search we can achieve practically optimal solution in $O(N) \div O(N^2)$ time [**P4**]. With vertex adjustment method we cannot improve obtained solution by additional iterations.

b) The presented *single-goal* bounding corridor can be extended to the case of *multiple-goal* corridor (see Fig. 3.4.3). With the search in the multiple-goal bounding corridor a family of solutions can be obtained, that is defined by the corridor width W. This kind of corridor have been further used for solving the multiple object *min*-ε *problem* [**P7**].

c) In the case of *min*-ε *problem* for closed contours, the corresponding analysis of solutions in the bounding corridor can be performed to find the sub-path with conjugate states that provides minimum of cost function (approximation error). This sub-path gives optimal approximation solution to the closed contour, including optimal selection of the starting point [**P6**].

3.3.3 Summary

Aiming to bridge the gap between slow optimal and fast heuristic algorithms for *min*-ε *problem* we introduced paradigm of bounding corridor and iterative reduced search approach. We can control trade-off between optimality and time performance with parameters of the proposed algorithm. The time complexity of the algorithm is between $O(N)$ and $O(N^2)$, that corresponds the complexity of fast heuristic algorithms. The processing rate with the proposed algorithm can be roughly estimated as 10^3–10^4 vertices per second depending on relative number of segments, solution fidelity, curve smoothness, and processor performance. On the other hand, with the proposed algorithm we can achieve optimal or near-optimal result, using a proper strategy.

Later we will use the iterative reduced search for the development of efficient algorithms in polygonal approximation field, namely, *min*-# *problem* for open curves, *min*-ε and *min*-# approximation of closed contours, and *min*-ε approximation of multiple objects.

3.5 Optimal algorithms for *min-# problem*

The *min-# problem* is formulated as follows: a given polygonal curve P, approximate it by another polygonal curve Q with the minimum number of segments M so that the approximation error does not exceed a given maximum tolerance ε. The *min-# problem* is motivated by necessity to obtain approximation with least number of segments that maintans a certain level of accuracy. The problem arises in practical task of vectorization, vector data reduction, and vector map simplification.

3.5.1 Survey of solutions

Probably, one of the the first optimal algorithms for *min-# problem* for error criterion L_∞ has been proposed by Papakonstantinou in 1985 [Papakonstantinou'85]. The *min-#* problem under uniform L_∞ measure (in fact, with infinite beam criterion) was formulated as optimization task and was solved by dynamic programming (DP) method. To reduce DP search, the longest line segment is defined by cone-intersection method of Sklansky and Gonzalez [Sklansky-Gonzalez'80].

One year later Dunham [Dunham'86] published *min-#* algorithm that uses DP approach for optimal approximation with least number of segments for error measure L_∞. He used modified scan-along algorithm of [Williams'78] and [Sklansky-Gonzales'80] with cone intersection to reduce DP search. The complexity of the algorithm is $O(N^3)$ in the worst case.

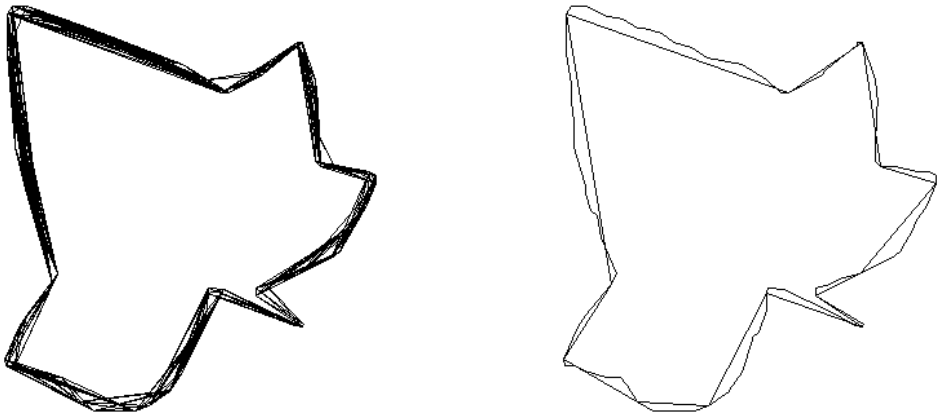


Figure 3.5.1: Graph $G_\varepsilon(P)$ constructed on digitzed curve P for error tolerance ε=10 (left). The *min-#* solution for ε=10 as the shortest path in the graph $G_\varepsilon(P)$ (right).

About the same time Imai and Iri [Imai-Iri'86, 88] present a unified approach for the problem by formulating it in terms of *graph theory*. At first, a directed acyclic graph $G_\varepsilon(P) = (V, E_\varepsilon)$ is constructed (see Fig. 3.5.1). Nodes V are vertices of the curve $P = \{p_1, \ldots, p_N\}$ and edges E_ε are the approximation line segments $E_\varepsilon = \{(p_i, p_j): 1 \le i < j \le N \mid D(p_i, p_j) \le \varepsilon\}$. Two nodes p_i and p_j of $G_\varepsilon(P)$ are connected with an edge (p_i, p_j), iff the correspondent approximation error (maximum deviation) $D(p_i, p_j)$ is at most the prescribed error tolerance ε: $D(p_i, p_j) \le \varepsilon$. Solving *min-#* problem consists of

finding the *shortest path path* from p_1 to p_N in digraph $G_\varepsilon(P)$. Since the graph is acyclic, this path can be found in time proportional to the number of edges [Dijkstra'59, Cormen-Leiserson-Rivest'90]. The brute-force method of constructing $G_\varepsilon(P)$ is to check for each pair of vertices p_i and p_j whether the error $D(p_i, p_j)$ is within error tolerance ε. There are $O(N^2)$ pairs of vertices and checking the error, corresponding to a pair, takes $O(N)$ time. This brute-force method for the graph $G_\varepsilon(P)$ construction takes $O(N^3)$, while finding the shortest path in G takes no more than $O(N^2)$ time. Thus, the bottlenck in the computation of *min*-# approximation is the construction of the graph $G_\varepsilon(P)$. The main efforts of researchers have concentrated on the problem of how to reduce complexity of the graph construction.

Melkman and O'Rourke [Melkman-O'Rourke'88] studied the 2-D *min*-# and *min*-ε problems. Their version of Imai-Iri's algorithms takes $O(N^2\log N)$ for *min*-# *problem*, and $O(N^2\log^2 N)$ for *min*-ε *problem*. Space complexity is $O(N^2)$ in both cases. Thus, the graph construction was reduced from $O(N^3)$ to $O(N^2\log N)$. They proposed cone-intersection algorithm to reduce the complexity of the graph constuction. This algorithm is analog of that of Sklansky and Gonzalez [Sklansky-Gonzalez'80].

Chan and Chin [Chan-Chin'96] reduced the time complexity to $O(N^2)$ for *min*-# *problem*, and $O(N^2\log N)$ for *min*-ε *problem* using the parallel-strip criterion They improved complexity of constructing graph G to $O(N^2)$. They have also shown that for closed polygonal curve the *min*-# *problem* can be solved in $S(N)$ time, where $S(N)$ denotes the time complexity for solving all-pairs shortest path problem of $G_\varepsilon(P)$. They also presented algorithm of linear time complexity for *min*-# problem for the convex curves (see Table 3.5.1).

Table 3.5.1: Summary of the time complexities of the polygonal algorithms of Chan and Chin [Chan-Chin'96].

Shape type	General		Convex	
	Open	Closed	Open	Closed
Min-# problem	$O(N^2)$	S(N)	$O(N)$	$O(N^2)$
Min-ε problem	$O(N^2\log N)$	S(N)log$N$	$O(N^2)$	$O(N^2)$

Based on the *infinite beam criterion*, Toussaint [Toussaint'85] solved the *min*-# *problem* in $O(N^2\log N)$ time and $O(N^2)$ space. Imai and Iri gave an $O(N^2\log^2 N)$ time and $O(N^2)$ space for the *min*-ε *problem*. Eu and Toussaint [Eu-Toussaint'94] published another algorithm for *min*-# *problem* under the infinite beam criterion that was claimed to take $O(N^2)$ time, and $O(N^2\log N))$ for *min*-ε problem. The space complexity is $O(N^2)$. Eu and Toussaint [Eu-Toussaint'94] also used the infinte beam criterion based on L_1 and L_∞ distance metrics. Using L_2 distance metric, the *min*-# *problem* can be solved in $O(N^3)$ time, and *min*-ε problem in $O(N^3\log N))$ time.

Ray and Ray [Ray-Ray'93] determined least number of segments with the minimum possible error by maximizing an objective function comprised of the length of line segments and the sum of absolute errors between the line segments and the digital curve (L_1 error measure).

Pikaz and Dinstein [Pikaz-Dinstein'95] found *min*-# approximation of the polygonal curve where the city-block metric is used to measure distance between the

approximation and the input curve. For the city-block the distance the complexity of the algorithm is $O(N^2)$.

Chen and Daescu [Chen-Daescu'98, Chen-Daescu'03,] presented a number of efficient algorithms for the 2-D *min*-# and *min*-ε *problems* in the same time bounds as [Chan-Chin'96, Eu-Toussaint'94], but using only $O(N)$ space in comparison with $O(N^2)$. Also they applied the technique to a special case of the 3-D *min*-# and *min*-ε *problems*.

Zhu and Seneviratne [Zhu-Seneviratne'97] have shown that cone-intersection algorithm [Sklansky-Gonzalez'80] has the problem that some peaks will disappear from the curve when the cone angle is the only evaluation factor and introduced modification to Sklansky and Gonzalez method. Based on the modified algorithm for they proposed new version of the algorithm for *min*-# *problem*.

Katsaggelos *et al*. [Katsaggelos-Kondi-Meier-Osterman-Schuster'98, Schuster-Katsaggelos'98] used polygonal approximation for lossy encoding with minimum rate and given distortion bounds. Actually, Katsaggelos *et al*. solved more general problem, than just a *min*-# one. In this case, the techniques based on geometrical computations, cannot be used. Because of the brute-force method applied for construction of the graph $G_\varepsilon(P)$, time complexity of their algorithm is $O(N^3)$.

Hosur and Ma [Hosur-Ma'99] following the approach of Katsaggelos *et al*. for *min*-# problem proposed cone intersection method for reducing the graph construction time, which actually has been introduced early in [Sklansky-Gonzalez'80] and have already been used in other algorithms for *min*-# *problem* [Papakonstantinou'85, Dunham'86].

Recently Agarwal *et al*. [Agarwal–Har-Peleg–Mustafa–Wang'02] proposed *min*-# algorithm for Hausdorff and Frechet error measure of $O(N^{4/3+\delta})$ time complexity, where $\delta>0$. Barequet *et al*. [Barequet-Chen-Daescu-Goodrich-Snoeyink'02] presented agorithms for approximating polygonal curves in 3D and higher dimensional spaces under the tolerance zone criterion.

Table 3.5.2: Summary of *min*-# and *min*-ε results from [Barequet-Chen-Daescu-Goodrich-Snoeyink'02].

	3 dimensions		$d\geq4$ dimensions	
Metric	*Min*-#	*Min*-ε	*Min*-#	Min-ε
L_2	$O(N^2\log N)$	$O(N^2\log^3 N)$	$O(N^{3-2/(\lfloor d/2\rfloor+1)}\text{polylog}N)$	$O(N^{73}\text{polylog}N)^*$
L_1 & L_∞	$O(N^2)$	$O(N^2\log N)$	$O(N^2)$	$O(N^2\log N)$

*For d=4 only.

With DP algorithm of Perez and Vidal the *min*-# *problem* for measure L_2 can be solved in $O(N^3)$ time (see also [Tseng-Juan-Chang-Lin'98]). Salloti proposed fast optimal algorithm for the problem based on the A*-search algorithm, and he introduced for *min*-ε *problem* with L_2 measure [Salotti'00, Salotti'01]. Using the heuristic error estimations to stop the search, he reduced complexity of Perez-Vidal algorithm from $O(N^3)$ to $O(N^2)$ time.

3.5.2 Problem of multiple solutions

As we can see, the error measure L_∞ is widely used in heuristic and optimal algorithms for *min-# problem*. The error measure L_∞ is preferred to other measures for obvious reasons. But careful study of these *min-#* algorithms based on the error measure L_∞ shows that all the algorithms suffer from an essential drawback, namely visible distortion of approximation curve (see Fig. 3.5.2, left)

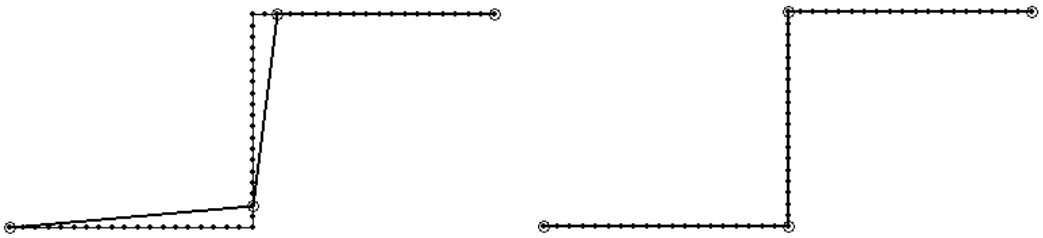


Figure 3.5.2: Examples of *min-#* approximation of 59-vertex test shape for error tolerance ε=2 with L_∞ error metrics: by algorithm based on the shortest path in digraph (left); by the proposed method [**P5**] with L_2 optimization. Vertices of the input curve are labeled with dots; the approximation points are labeled with circles.

Firstly, the drawback was noted by Papakonstantinou [Papakonstantinou'94] who has been studying problem of data reduction of ECG signals by polygonal approximation with minimum number of line segments for measure L_∞. As it was mentioned above, he wrote, that so called optimal solution is not unique. Actually, the number of the equivalent optimal solutions (having the same minimal number of line segments) can be extremely large. The reason of the effect is as follows: with the error measure L_∞ we can control only the maximum deviation, but not the deviations for all the vertices of the approximated curve. Formally, the solutions satisfy the constraint on maximum deviation, but the solutions have visible shape distortion.

To overcome the problem, Papakonstantinou proposed how to select the best one among the formally optimal solutions, with the minimal integral square error. In other word, he proposed to use error measures L_∞ and L_2 jointly to obtain solution, which satisfies the condition on maximum deviation $D(P) \le \varepsilon$ and provides minimum to the approximation error $E(P)$ with measure L_2. The complexity of the modified *min-#* algorithm is defined by the complexity of the base algorithm but the processing time is bigger because of L_2-optimization. The proposed method provides good results. It is easy to understand why the effect has been discovered by studying approximation of ECG signals. For smooth curves the distortion effect is small, whereas for curves with sharp corners, such as ECG signals, the effect is clearly visible.

Haugland *et al*. [Haugland *et al*.'97a, Nygaard-Haugland'97] attacked the problem by joint use of measures L_∞ and L_2 from another side. They solved problem of ECG data reduction with a given data compression ratio by algorithm for *min-*ε *problem* with measure L_2. Proximity of the approximation curve to the input one is measured by the sum of squared errors, but the deviation (error value) for every individual vertex is beyond the control of the algorithm [Haugland *et al*.' 97b]. To overcome the problem, they proposed modified version of the base DP algorithm for *min-*ε *problem*, introducing additional check if the local deviation less than the tolerance level. To

reduce the processing time for the computation of the L_∞ approximation error, a fast algorithm based on convex hull construction has been used.

Figure 3.5.3: The test 2900-vertex shape "Australia": all shortest paths in the graph $G_\varepsilon(P)$ for a given error tolerance $\varepsilon=1$.

Authors formulated a new version of *min-ε problem* using two conditions: a) the number of segments should be less than a given value: $M \le M_0$, and b) the maximum deviation should be below the tolerance level: $D(P) \le \varepsilon$. As we can see, the conditions are contradictious: if the given number of segments M_0 is too small, the constraint on maximum deviation cannot be satisfied with any number of segments. The problem can be solved with constraint on the maximum error, or on the number of segments, but not on both of them.

Pikaz and Dinstein in [Pikaz-Dinstein'95] solved *min-# problem* as the shortest path on digraph with city-block distance metrics. The obtained polygonal approximation is minimal with respect to the number of vertices under a given maximal error. As in the mentioned above case, there might be several different polygonal approximations with the same minimal number of vertices. Pikaz and Dinstein offered to select solution that is optimal according to the criterion of *minimal maximal* distance between the approximation and the original curve. Although the proposed method allows reducing the maximum deviation, the local deviations (smaller than tolerance level) are still out of control.

3.5.3 Selection of the best solution with reduced search

In [**P5**] we apply the iterative reduced search for solving the *min-# problem*. At first, the initial solution approximation for a given error tolerance ε is obtained with any algorithm for *min-# problem*, then a bounding corridor is constructed along the reference path, and finally the solution is searched in the bounding corridor (see Figs. 3.5.2 and 3.5.4).

Experiments have shown, that although the integral squared error $E(P)$ is reduced after L_2-optimization, the maximum deviation $D(P)$ can be bigger than a given

tolerance level in a few outliers. The problem can be solved by additional iterations with reduced search to increase the number of linear segments M. Another solution is to perform L_2-optimization with L_∞ constraints on the approximation error by cost of higher time complexity because of L_∞ approximation error calculation.

In high quality vectorization tasks, the increase of the maximum deviation is about 10% and takes place for an outliers only. In our opinion, in practical applications, the given constraint of maximum deviation ε can be treated in non-strong way. In other words, to reduce processing time we can use the iterative reduced search to optimize location of vertices. The obtained approximation points satisfies the given constraint on the deviation, excluding may be a few outliers.

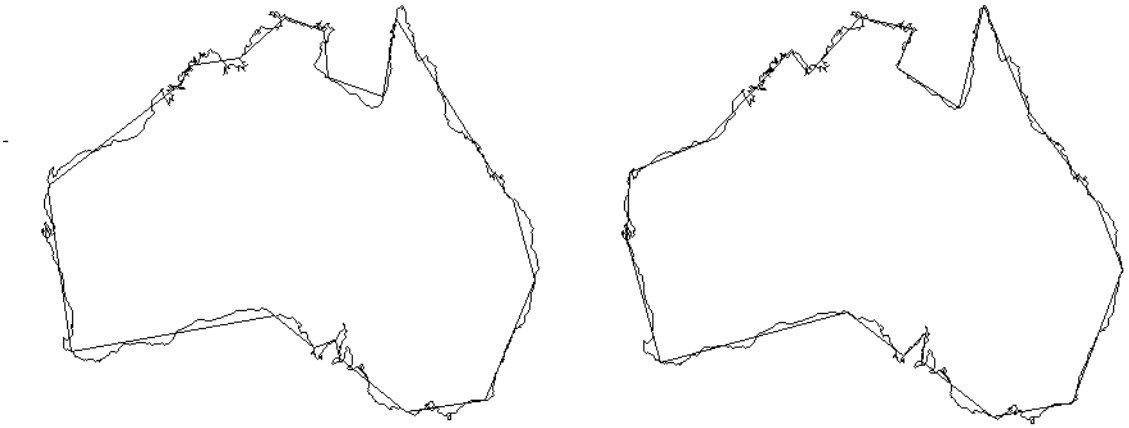


Figure 3.5.4: The *min*-# approximation for ε = 10: the solution as one of the shortest path in graph $G_\varepsilon(P)$: processing time: $D_1(P)$=1, E_1=629, T_1=4.5 s (left); this solution after processing with reduced search DP algorithm: E_2=298, maximum distance $D_2(P)$=1.08; additional processing time T_2=0.5 s (right).

Summary

We have proposed to use iterative reduced search approach to improve the quality of *min*-# approximation solutions obtained. The initial min-# solution can be found with any algorithm with L_∞ error measure. Then we optimize location of the approximation vertices with L_2 error measure using iterative reduced search. The proposed algorithm is tailored for high-quality vectorization of digitized curves.

3.6 Approximation of closed contours

3.6.1 Problem formulation

A closed *N-vertex polygonal curve P* in 2-dimensional space is represented as the ordered set of vertices $P=\{p_1, \ldots, p_N \mid p_N \equiv p_1\}$, when the last vertex is equal to the first one. The approximation curve Q consists of $(M+1)$ vertices: $Q=\{q_1, \ldots, q_{M+1} \mid q_{M+1} \equiv q_1\}$, where the set of vertices q_m is a subset of P. In the case of *closed* contours, we have to find optimal allocation of all approximation vertices including the starting point (see Fig. 3.6.1).

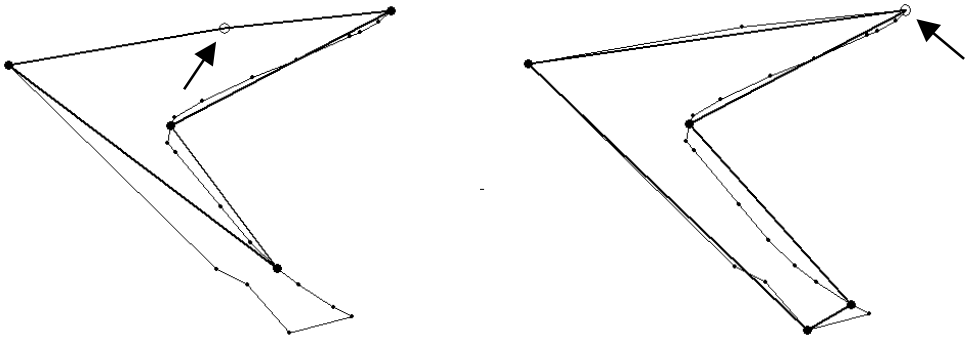


Figure 3.6.1. Examples of *min*-ε approximation of the closed contour #1 by M=5 linear segments: with non-optimal starting point (left); with the optimal starting point (right). The starting points are marked with arrow.

3.6.2 Survey of solutions

Imai and Iri [Imai-Iri'88] wrote that the closed curve *min*-# (resp. *min*-ε) problem can be solved by solving the N ordinary open curve *min*-# (resp. *min*-ε) problems. If complexity of algorithm for open curve *min*-# *problem* is $O(N^2)$, the complexity for closed contour approximation is $O(N^3)$ time. Perez and Vidal [Perez-Vidal'94] also wrote that the complexity of the straightforward *min*-ε algorithm for closed contour is $(N-M)$ times that of the algorithm for open curve. Taking into account the complexity of modified full-search DP algorithm [Perez-Vidal'94, **P4**] for closed contours *min*-ε problem it gives time complexity of $O(M(N-M)^3)$.

It was shown in [Chan-Chin'96] that the *min*-# *problem* for closed curve can be solved in $S(N)$ time, where $S(N)$ is the time for solving the all-pairs shortest path problem for a graph of N vertices.

There also exist a number of heuristic approaches for selecting the starting point. Sato [Sato'92] chooses the farthest point from the center of gravity as a starting point.. Ray and Ray [Ray-Ray'93] in their formulation of the approximation problem specify neither the error nor number of line segments as cost function, but a ratio of the local L_1 error measure to the length of the segments. For the case of the closed contours they proposed to extend the sequential search beyond the current starting point until to the first approximation vertex to revise the choice of the starting point.

Pikaz and Dinstein [Pikaz-Dinstein'95] considered *min-# problem* with city-block error metrics, and proposed an algorithm based on the shortest path problem, including the search of optimal starting point by an algorithm of complexity $O(N^2)$. Zhu and Seneviratne [Zhu-Seneviratne'97] considered the approximation problem with L_∞ metrics, and proposed an $O(N^2)$ time algorithm for open curves. An iterative procedure for optimizing the choice of the starting point was also suggested for the case of closed curve. Schroeder and Laurent [Schroeder-Laurent'99] in near-optimal algorithm for *min-# problem* perform preliminary approximation until the first approximation vertex is reached, and use this vertex as a new starting point for the second iteration.

Horng and Li [Horng-Li'02] proposed a two-step method to select the starting point: at the first step, optimal approximation with any starting point is performed. At the second step, the approximation is performed again using the $\lfloor M/2 \rfloor$-th vertex as the new starting point.

To sum up, the proposed heuristic approaches for closed curves are sub-optimal whereas the optimal choice of the starting point is time consuming. Thus, the problem has not been solved satisfactory. One of the heuristic approaches is to perform preliminary approximation using one of the approximation vertices as a new starting point for final approximation [Ray-Ray'93, Pikaz-Dinstein'95, Zhu-Seneviratne'97, Horng-Li'02]. With this approach for starting point selection it is possible to obtain good results but, in general, the optimality of solution cannot be guaranteed (see Fig. 3.6.2).

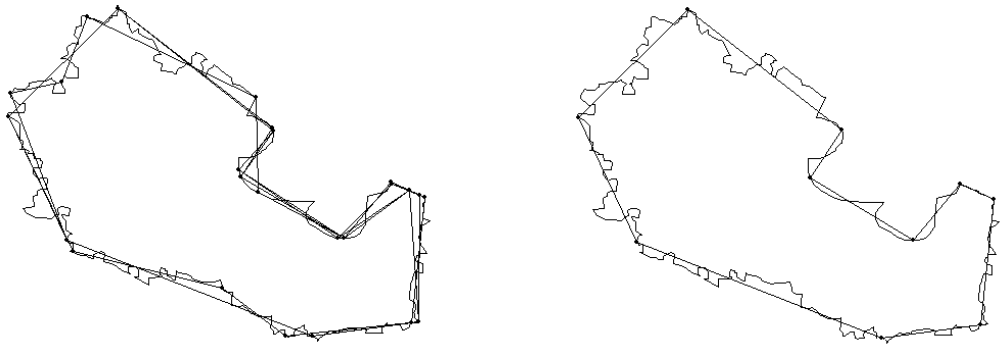


Figure 3.6.2: Results of *min*-ε approximation of 417-vertex closed contour # 2 by M=10 linear segments for all the possible initial starting points: by heuristic method [Horng-Li'02] (left); by the proposed method for optimal staring point allocation [**P6**].

3.6.3 Analysis of state space for *min*-ε *problem*

In the mentioned heuristic algorithms after the first iteration, however, the search starts from scratch and loses the information of the previous run. The idea of the approach we proposed in [**P6**] is to extend dynamic programming search in special state space beyond the last vertex of contour in cyclic way until the last point will be reached again. Solutions for sub-tasks in the state space are analyzed to find the best possible starting point.

We call two states on the sub-path in state space *conjugate* if the sub-path with the states corresponds to approximation of the input contour with the same start and end points. Approximation error for the sub-path is defined as the difference of the

cost function values of the conjugate states. To find the optimal starting point we have to find two conjugate states, which provide minimum of the approximation error $E(m)$ for sub-path defined for all goal states in the state space (see Fig. 3.6.3). As we know, in DP algorithm for *open* curve solutions are constructed under the condition $q_{M+1}=p_N$. Propagating DP search in state space cyclically beyond the end point of P, we remove the restriction and make the approximation vertex q_{M+1} "free". Then we analyze solutions of all relevant sub-tasks with the free vertex q_{M+1} to find the best location for the starting point. If the original starting point gives the optimal solution, the correspondent path to goal state (N,M) is limited by conjugate states.

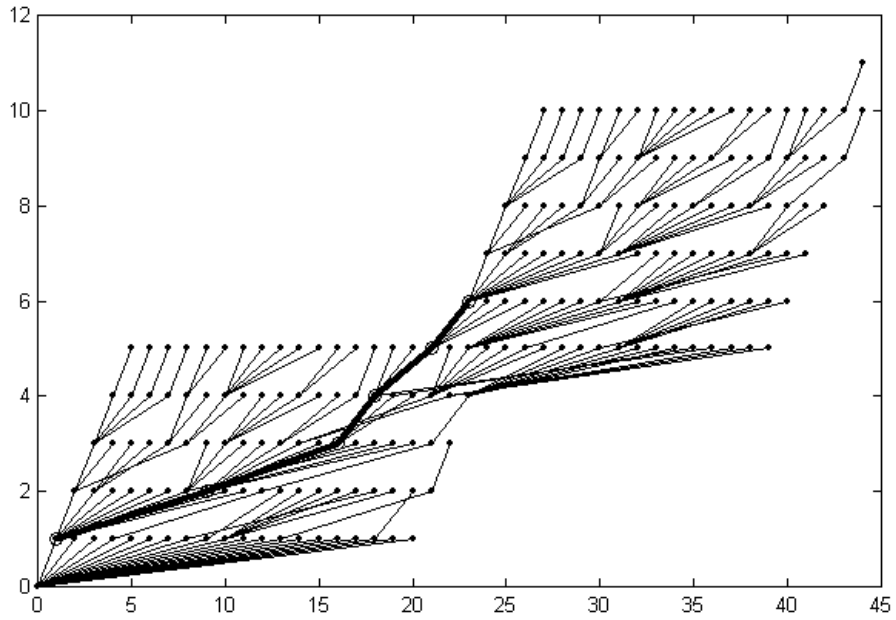


Figure 3.6.3: State space Ω_2 for the closed contour #1. Optimal sub-path in the space Ω_2 for the conjugate states with minimum cost function is emphasized with thick line and circles. Approximation with the optimal starting point that corresponds the sub-path is on Fig. 3.6.1, right.

Generally speaking, global optimality of the found starting point cannot be guaranteed. Moreover, the sub-path with conjugate states may be does not exist in the state space Ω_2. Sometimes it happens in the case of coarse approximation when the number of vertices is big ($N \sim 1000$) and the number of vertices is relatively small ($M < 10$). We can extend DP search cyclically to the next runs along the curve in attempt to find better solution in the state space Ω_k, where $k \geq 3$. Anyway, the state space contains solution for the original starting point.

3.6.4 *Min*-# approximation of closed contours

To solve the *min-# problem* for closed curves, we have to find approximation of the closed contour P by another closed contour Q with minimal number of line segments with an approximation error within given error tolerance level: $D(P) \leq \varepsilon$. The approximation error $D(P)$ with measure L_∞ is given as the maximum Euclidean distance from the vertices of P to the approximation linear segments.

To solve the *min-# problem* a digraph is constructed on the vertices of the input contour P. In the digraph, a pair of vertices p_i and p_j are connected with an edge if the

approximation error of the curve segment between the vertices is less than a given error tolerance: $d(p_i,p_j)\leq\varepsilon$. The optimal solution is then a given by solving the shortest path in the digraph. This can be solved by using DP algorithm for the shortest path problem in the digraph.

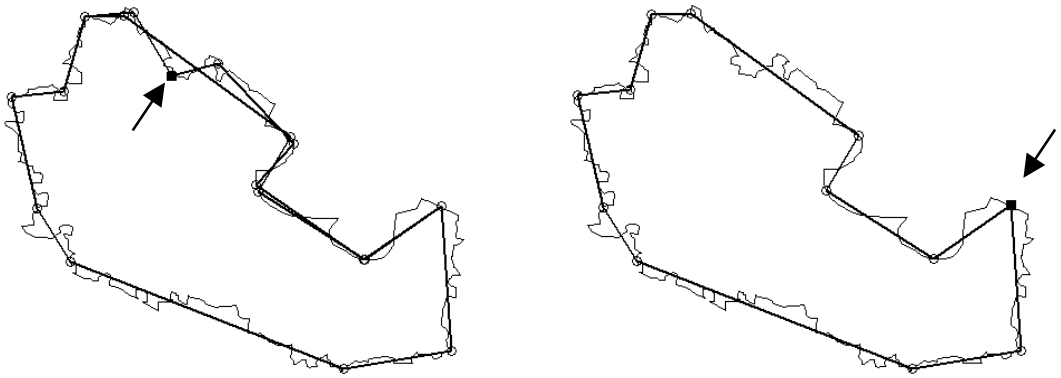


Figure 3.6.4. Result of *min*-# approximation of double-size curve P_2 for error tolerance ε=25 (left). *Min*-# approximation of the input closed contour #2 with optimal starting point obtained by analysis of DP solutions for P_2 (right). Starting points are marked with arrow.

To find the optimal approximation for closed contour we shall follow the approach introduced in [**P6**]: perform approximation of the wrapped double-size closed contour P_2 and then analyze the state space. In the case of min-# problem, the analysis of the space is reduced to the analysis of the solutions for the relevant sub-problems: we have to found the part of the approximation polygonal curve Q_2, which have the same start and end points (see Fig. 3.6.4). As in the previous case, the initial approximation of the input curve can be extended cyclically to find better solution. Even if such a solution with less number of segments is not found, the obtained solution for the original starting point anyway can be used.

3.6.5 Summary

We have introduced a new approach for *min*-ε and *min*-# approximation of closed contours based on dynamic programming method for open curves. It performs approximation of the cyclically extended double-size contour and then makes analysis of the state space to select the best starting point. The processing time is double of that of the approximation of the corresponding open curve. The time complexity of the algorithms is defined by the complexity of approximation algorithms for open curves in use. For solving the *min*-ε *problem* the suggested method can be used along with iterative reduced search algorithm with time complexity between $O(N)$ and $O(N^2)$.

3.7 Multiple-object polygonal approximation

3.7.1 Problem formulation

The problem of polygonal approximation of a single curve can be extended to the case of multiple curves (see Fig. 3.7.1) as follows:

> ***Multiple object min-# problem***: Given K polygonal curves $P_1, P_2, ..., P_K$, approximate it by set of K another polygonal curves $Q_1, Q_2, ..., Q_K$ with the minimum total number of segments M so that the approximation error does not exceed a given maximum tolerance ε.
>
> ***Multiple object min-ε problem***: Given K polygonal curves $P_1, P_2, ..., P_K$, approximate it by set of K another polygonal curves $Q_1, Q_2, ..., Q_K$ with a given total number of segments M so that the total approximation error is minimized.

Solution for the *multiple-object min-# problem* depends on the error measure in use. In the case of L_∞ error measure, the problem reduces to the *single-object min-# problem* as the optimization can be solved for every object independently [Schuster-Katsaggelos'98]. In the case of additive error measures (L_1, L_2, *etc.*), on the other hand, the problem is not trivial. Fortunately, in practical applications we mostly have to deal with error measure L_∞ because the use of additive error measures (L_1 or L_2) is not practical in the case of *min-# problem*.

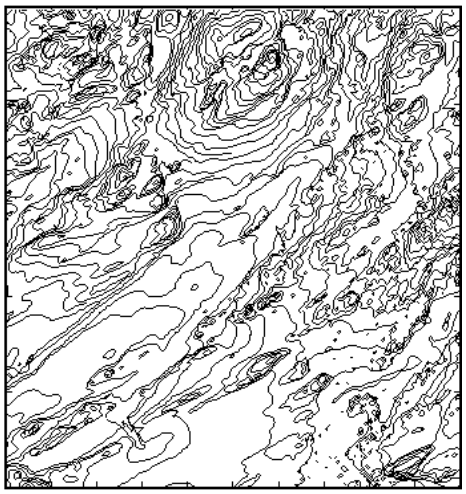


Figure 3.7.1: Example of multiple-object vector map: "Elevation map" [NLS], the total number of vertices N=38924, the number of objects K=569.

The case of *min*-ε approximation of *multiple objects* (with any error measure) is more complicated. The optimal approximation cannot be obtained by solving the approximation of each individual objects separately because the given total number of approximation segments should be optimally distributed among all objects. For example, uniform allocation of the segments can assign too many segments to the less complicated objects and, respectively, lacking the segments for more complicated objects. This situation is illustrated in Fig. 3.7.2.

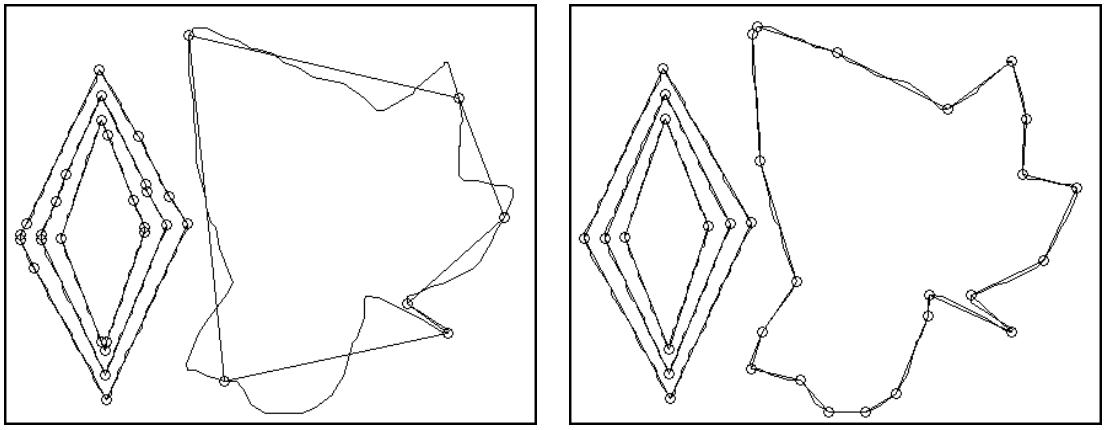


Figure 3.7.2: Example of multiple object approximation with uniform allocation of the segments numbers ($M_k \approx N_k M/N$) (left), and with optimal allocation of the segments number (right). The number of points in the objects are $N_D = 3\times121$ ("Diamond"), and $N_L = 82$ ("Leaf"). The corresponding number of segments are $M_D=3\times9$ and $M_L=6$ with uniform allocation of the segments number, and $M_D=3\times4$ and $M_L=21$ with the optimal allocation of the segments number.

The *multiple-object min-ε problem* can be formulated as the following optimization tasks for the total approximation error $E(P_1, \ldots, P_K, K)$ for K objects $\{P_1, \ldots, P_K\}$:

$$E(P_1,\ldots,P_K,M) = \min_{\{M_k\}} \min_{\{q_m\}} \sum_{k=1}^{K} \sum_{m=1}^{M_k-1} e^2\left(q_{k,m}, q_{k,m+1}\right) \text{ subject to } : \sum_{k=1}^{K} M_k \le M,$$

where $e^2(q_{k,m}, q_{k,m+1})$ is approximation error with measure L_2 of curve segment $\{p_i, \ldots, p_j\}$ of P_k by the correspondent line segment $(q_{k,m}, q_{k,m+1})$ of Q_k; M is a given total number of segments and M_k is the number of segments in object P_k.

3.7.2 Survey of solutions

Shuster and Katsaggelos considered problem of lossy encoding of object boundaries in rate-distortion sense [Schuster-Katsaggelos'98]. The digitized contour is approximated by polygon Q, which leads to the smallest distortion for a given rate (number of bits). Actually, the case when the rate is proportional to the number of line segments corresponds to the *multiple-object min-ε problem*. Schuster and Katsaggelos [Schuster-Katsaggelos'98] considered two different classes of distortion measures: L_∞ and L_2. For the first class (measure L_∞), they used scheme based on the shortest path algorithm for a weighted directed acyclic graph. For the second class (measure L_2) they proposed two algorithms.

The first algorithm is based on the Lagrangian multipliers method, which uses the DP algorithm for the shortest path in a directed acyclic graph. The time complexity of the Lagrangian approach for a fixed multiplier λ is the same as for the base shortest path algorithm. The shortest path algorithm is invoked several times by the bisection algorithm to find the optimal λ^* and, hence, the time complexity is a function of the number of required iterations. Thus, the complexity of the first algorithm is $O(N^2 \log N)$ because it is defined by the complexity of the shortest path algorithm and the number of bisection iterations. Since the Lagrangian multiplier method can only

find solutions on the convex hull of the operational rate-distortion functions, they also proposed a tree-pruning based algorithm. This method is a one pass variant algorithm with the complexity of $O(N^2)$, but the efficiency of the pruning scheme cannot be guaranteed in general.

Algorithms with the complexity of higher than $O(N^2)$ can be used for relatively small input. This can be suitable for the encoding of object contours for MPEG-4 standard [Katsaggelos-Kondi-Meier-Osterman-Schuster'98] but it can be too slow in the case of large vector maps.

3.7.3 Reduced search algorithm

In the **P7**, the *min*-ε *problem* of optimal approximation of multiple-object vector data was considered. We have introduced two algorithms for solving the problem based on dynamic programming: full search and iterative reduced search. Full search algorithm of complexity $O(N^3)$ has been introduced to represent the DP approach for joint optimization the number of segments and the approximation of the individual objects. At first, the rate-distortion functions are computed for all the objects as minimal approximation error for all possible number of segments. Then problem of optimal allocation of constrained resource is solved by DP algorithm. Finally the optimal approximation for every object is calculated for the found optimal distribution of segment numbers.

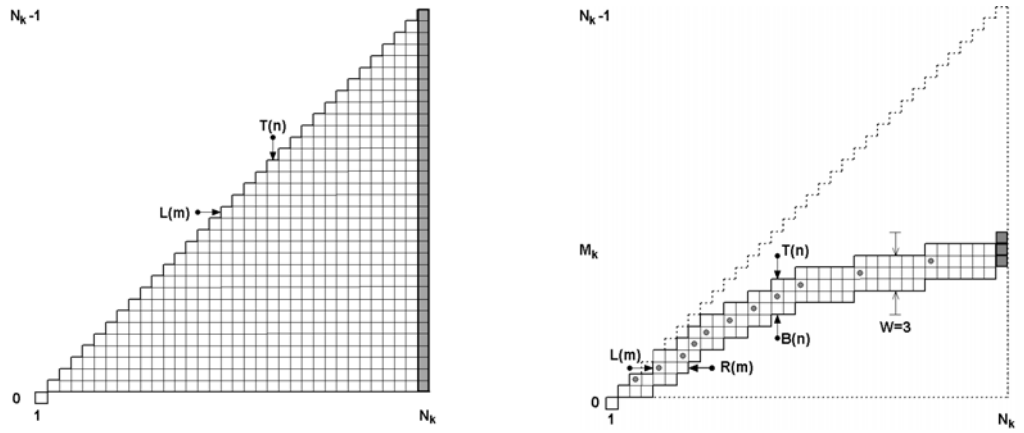


Figure 3.7.3: Illustration of the *multiple-goal* state space Ω_k for full search DP algorithm (sample problem of N_k=34 and M_k=12) (left), and the *multiple-goal* bounding corridor of width W=3 in the state space for reduced search DP algorithm (sample problem of N_k=34 and M_k=12) (right). The reference path $H(m)$ in the corridor is marked with dark gray circles, and the goal states with gray squares.

Because the time and space complexity of the full search algorithm is high, the iterative reduced search approach was generalized to the problem under consideration. We follow the main idea of the reduced search by reducing the search space by a given preliminary solution for the approximation, and then perform the search in the reduced space iteratively (see Fig. 3.7.3). Main difference to the full search is that a smaller search area is needed, which makes the algorithm faster. The iterative reduced search algorithm has time complexity of $O(N)$–$O(N^2)$. This is significantly smaller than the $O(N^3)$ of the full search, or the $O(N^2\log(N))$ of [Schuster-Katsaggelos'98]. The reduced search approach is also applicable for very large data sets with reasonable memory requirements. Although the optimality of the algorithm cannot be

guaranteed in general, the experiments indicate that the method is capable of finding the optimal solution even in the case of very large data sets (see Fig. 3.7.4). The algorithm can also be tuned for obtaining very fast sub-optimal solutions by reducing the number of iterations and corridor width.

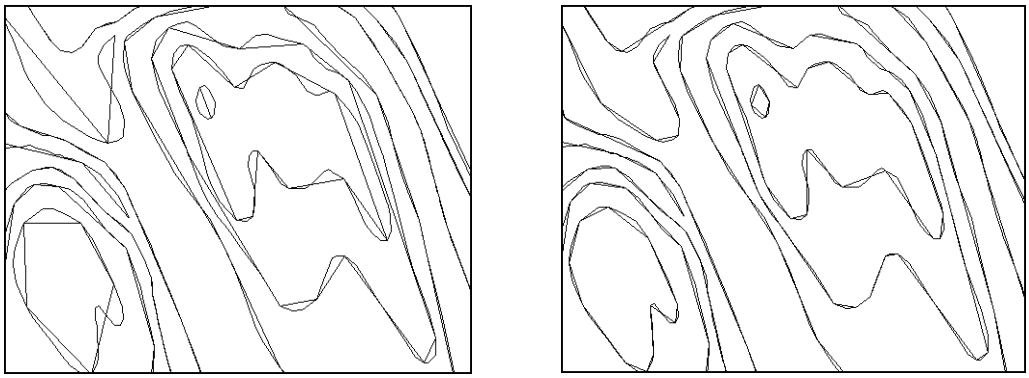


Figure 3.7.4: Approximation results for test data "Elevation map": solution (fragment) for straightforward approach, Douglas-Peucker approximation with uniform allocation of the segments numbers (left): $E_0 = 892158$, $T_0 = 1.7$s (left); final result (fragment) of optimal approximation with optimal number of segments after 20 iterations; $E_{20} = 124093$, $T_{20} = 22$ s (right). The vector data reduction ratio is 5:1 (N=38924, M=7784). Processing time for full search algorithm is 157 s.

3.7.4 Summary

We have introduced two algorithms for solving the problem based on dynamic programming: full search and iterative reduced search. The algorithms optimize the number of segments and the approximation of the individual objects jointly. Experimental results indicate that the proposed algorithm reaches the optimal solution in all cases tested even though the optimality cannot be guaranteed in general. The iterative reduced search algorithm has time complexity of $O(N)$–$O(N^2)$ depending on the given the data reduction ratio.

4 Conclusions

4.1 The main results

We have presented vectorization system, which uses locally adaptive binarization algorithm. Efficient implementation of distance transform based thinning algorithm for large binary images was also proposed. The developed algorithm can be used in multiprocessor parallel machines as well as in single-processor computers. The developed system was applied for feature-based filtration of binary images as a part of context-based image compression.

In the second part of the thesis, we introduced a new paradigm of bounding corridor in the state space. Using the iterative reduced search approach in the corridor we introduced fast algorithm for the *min*-ε *problem*, and also for the *min*-# *problem* with joint use of error measures L_2 and L_∞. Furthermore, we extended the idea to the case of closed contours. Finally, we generalized the iterative reduced search approach to the case of multiple-object *min*-ε *problem* for vector data reduction.

4.2 Main conclusions

To bridge the gap between slow optimal and fast heuristic algorithms for *min*-ε *problem*, we introduced paradigm of bounding corridor and iterative reduced search approach. We have released condition of solution optimality to reduce the time complexity of the original full search algorithm. With the proposed iterative reduced search algorithm we have again approached to the 100% optimality very closely. The paradigm was used for solving other approximation problems, including *min*-# *problem* for open curves, *min*-ε and *min*-# approximation of closed contours, *min*-ε approximation of multiple objects.

4.3 Open problems

Although in practice, we can achieve optimal result with iterative reduced search the problem of optimality of the solutions in general case is still open. The analogous problem concerns the optimality of solutions obtained with analysis of the state space for closed contours with multi-run cyclical DP search.

4.4 Topics for the future research

The future research area includes the following tasks in digital cartography. Using the developed fast algorithms for polygonal approximation we can perform better context modeling of input vector data for lossless compression of vector maps using Minimum Description Length (MDL) approach.

Other important area of research in digital cartography is map simplification with preserving of topology. Currently the topology-preserving approximation is performed with non-optimal heuristic algorithms. With the proposed approach quality of polygonal approximation solution can be improved by the same time cost.

References

[1] Ablameyko S., Aparin G., Bereishik V., Homenko M., Knowledge-based interpretation of roads and correlated objects on map drawing, *Proc. Int. Conf. Control, Automation, Robotics, and Computer Vision-ICARVC'94*, Singapore, pp.343-347, 1994.

[2] Ablameyko S.V. Bereishik V., Frantkevich O., Paramonova N., Melnik E., Homenko M., Interpretation of engineering drawings: Techniques and experimental results, *Pattern Recognition and Image Analysis*, 5: 380-401, 1995.

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

[4] Ageenko E., *Context-based Compression of Binary Images*, Ph.D. Thesis, CS Dept, University of Joensuu, Joensuu, Finland, 2000.

[5] Agarwal A., Klawe M.M., Moran S., Shor P., Wilber R., Geometric applications of a matrix-searching algorithm, *Algorithmica*, 2: 195-208, 1987.

[6] Agarwal P.K., Har-Peleg S., Mustafa N., Wang Y., Near linear-time approximation algorithms for curve simplification. *Proc. 10th Annual European Sympos. on Algorithms*, 29-41, 2002.

[7] Anderson I.M., Bezdek J.C., Curvature and tangential deflection of discrete arcs: A theory based on the commutator of scatter matrix pairs and its application to vertex detection in planar shape data, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6; 27-40, 1984.

[8] Ansari N., Delp E.J., On detecting dominant points, *Pattern Recognition*, 24: 441-450, 1991.

[9] Aoki Y., Shio A., Arai H., Odaka K., A prototype system for interpreting hand-sketched floor plans, *Proc. Int. Conf. on Pattern Recognition-ICPR'96*, 1996.

[10] Aoyama H., Kawagoe M., A piecewise linear approximation method preserving visual feature points of original figures, *Comput. Vision, Graphics and Image Process*.: *Graphical Models and Image Process*, 53: 435-446, 1991.

[11] Arcelli C., Cordella L., Levialdi S., Parallel thinning of binary pictures, *Electronic Letters*, 11: 148-149, 1975.

[12] Arcelli C., Ramella G., Finding contour-based abstraction of planar patterns, *Pattern Recognition*, 26:1563-1577, 1993.

[13] Arcelli C., Sanniti di Baja G., A width-independent fast thinning algorithm, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 7(4): 463-474, 1985.

[14] Arcelli C., Sanniti di Baja G., A one-pass two-operation process to detect the skeletal pixels on the 4-distance transform, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(4): 411-414, 1989.

[15] Asada H., Brady M., The curvature primal sketch, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(1):2-14, 1986.

[16] Attneave F., Some informational aspects of visual perception, *Physical Review,* 61:1833-193, 1954.

[17] Baek J.H., Teague K.A., Parallel thinning on a distributed memory computer, *Proc. 5th on Distributed Memory Computing Conf.-DMCC5*, Charleston, USA, 72-75, April 1990.

[18] Baek Y.S., Teague K.A., Chang Y.S., Parallel extended local feature extraction on distributed memory computer, *Proc. Int. Conf. on Multisensor Fusion and Integration for Intelligent System-MFI'94*, Las Vegas, USA, October 2-5, 1994.

[19] Ballard D.H., Strip trees: A hierarchical representation for curves, *Comm. ACM*, 24(5): 310-321, 1981.

[20] Barequet G., Chen D.Z., Daescu O., Goodrich M.T., Snoeyink J., Efficiently approximating polygonal paths in three and higher dimensions, *Algorithmica*, 33(2): 150-167, 2002.

[21] Beau V., Singer M., Reduced resolution and scale space for dominant feature detection in contours, *Pattern Recognition*, 34: 287-297, 2001.

[22] Baumgart B.G., *Geometric Modeling for Computer Vision*, PhD Thesis, CS Dept, Stanford University, STAN-CS-74-463, 1974.

[23] Beffert H., Shinghal R., Skeletonizing binary patterns on the homogeneous multiprocessor, *Pattern Recognition and Artificial Intelligence*, 3(2): 207-217, 1989.

[24] Bellman R., *Dynamic Programming*, Princeton University Press, New Jersey, 1957.

[25] Bellman R., On the approximation of curves by line segments using dynamic programming, *Comm. ACM*, 4(6): 284, 1961.

[26] Bernsen J., Dynamic thresholding of grey-level images, *Proc. Int. Conf. on Pattern Recognition-ICPR'86*, 1251-1255, 1986.

[27] Borgefors G., Distance transformations in arbitrary dimensions, *Computer Vision, Graphics, and Image Processing*, 27(3): 321-345, 1984.

[28] Borgefors G., Distance transformations in digital images, *Computer Vision, Graphics, and Image Processing*, 34(3): 344-371, 1986.

[29] Boxer L., Chang C.-S., Miller R., Rau-Chaplin A., Polygonal approximation by boundary reduction, *Pattern Recognition Letters*, 14: 111-119, 1993.

[30] Cantoni A., Optimal curve fitting approximation with piecewise linear functions, *IEEE Trans. Computers*, 20: 54-67, 1971.

[31] Cederberg R.L.T., An iterative algorithm for angle detection on digital curves, *Proc. 4th Int. Joint Conf. on Pattern Recognition*, Kyoto, Japan, 576-578, 1978.

[32] Chai I., Dori D., Orthogonal Zig-Zag: An efficient method for extracting lines from engineering drawings, in *Visual Form*, C.Arcelli, L.P. Cordella, G. Sanniti di Baja, (Eds.), Plenum Press, New York and London, pp. 127-136, 1992.

[33] Chan W.S., Chin F., On approximation of polygonal curves with minimum number of line segments or minimum error, *Int. J. Comput. Geom. Appl.*, 6: 59-77, 1996.

[34] Chandran S., Kim S.K. Mount D.M., Parallel computational geometry of rectangles, *Algorithmica*, 7: 25-49, 1992.

[35] Chhabra A.K., Dori D., (Eds.), *Graphics Recognition-Recent Advances*, Lecture Notes in Computer Science, Vol.1941, 3-19. Springer Verlag, September 2000.

[36] Chen D.Z., Daescu O., Space-efficient algorithms for approximating polygonal curves in two-dimensional space, *Proc. 4th Annual Int. Conf. on Computing and Combinatorics*, 55-64, 1998.

[37] Chen D.Z., Daescu O., Space-efficient algorithms for approximating polygonal curves in two-dimensional space, *Int. J. of Comput. Geometry & Applications*, 13(2): 95-111, 2003.

[38] Chen Y.S., Hsu W.H., A modified fast parallel algorithm for thinning digital patterns, *Pattern Recognition Letters*, 7: 99-106, 1993.

[39] Chen L.H., Liao H.Y., Wang J.Y., Fan K.C., Hsieh C.C., An interpretation system for cadastral maps, *Proc. Int. Conf. on Pattern Recognition-ICPR'96*, 1996

[40] Chen J.-M., Ventura J.A., Wu C.H., Segmentation of planar curves into circular and line segments, *Image Vision and Computing*, 14: 71-83, 1996.

[41] Chhabra A.K., Dori D., (Eds.), *Graphics Recognition-Recent Advances*, Lecture Notes in Computer Science, Vol.1941, Springer Verlag, 2000.

[42] Chia T.-L., Wang K.-B., Chen Z., Lou D.-C., Parallel distance transforms on a linear array architecture, *Information Processing Letters*, 82: 73-81, 2002.

[43] Chiang J.Y., Tue S.C., Leo Y.C., A new algorithm for line image vectorization, *Pattern Recognition*, 31(10): 1541-1549, 1998.

[44] Chinnasarn K., Rangsanseri Y., Thitimajshima P., Removing salt-and-pepper noise in text/graphics images, *Proc of the Asia-Pacific Conf. on Circuits and Systems-APCCAS'98*, 459-462, 1998.

[45] Chung J-W., Lee J.-H., Moon J.-H. , Kim J.-K., A new vertex-based binary shape coder for high coding efficiency, *Signal Processing*: *Image Communications*, 15: 655-684, 2000.

[46] Chung Y., Prasanna V.K., Parallelization of perceptual grouping on distributed memory machines, *Proc. of Conf. on Computer Architectures for Machine Perception Conf.*, Corno, Italy, 323-330, 1995.

[47] Chung Y., Prasanna V.K., Wang C.-L., A fast asynchronous algorithm for linear feature extraction on IBM SP-2, *Comm. ACM*, 30(2), 1987.

[48] Chung K.-L., Yan W.-M., Chen W.-Y., Efficient algorithms for 3-D polygonal approximation based on LISE criterion, *Pattern Recognition*, 35: 2539-2548, 2002.

[49] Computational Geometry Impact Force, *Application Challenges to Computational Geometry*, Technical Report, TR-521-96, 1996.

[50] Cordella L.P., Dettori G., On $O(n)$ algorithm for polygonal approximation, *Pattern Recognition Letters*, 3: 93-97, 1985.

[51] Cormen T.H., Leiserson C.E., Rivest R.L., *Introductions to Algorithms*, The MIT Press, 1990.

[52] Cornic P., Another look at the dominant point detection of digital curves, *Pattern Recognition Letters*, 18 (1): 13-25, 1997.

[53] Cox M.G., Curve fitting with piecewise polynomials, *J. Inst. Math. Appl*., 8: 36-52, 1971.

[54] Davis L., Understanding shape: Angles and sides, *IEEE Trans. Computers*, 26: 236-242, 1977.

[55] Davis L.S., Rosenfeld A., Curve segmentation by relaxation labeling, *IEEE Trans. Computers*, 26: 1053-1057, 1977.

[56] Davis L.S., Rosenfeld A., Hierarchical relaxation for waveform parsing, *Proc. Int. Workshop on Computer Vision Systems*, 101-109, 1978.

[57] Deguchi K., Aoki S., Regularized polygonal approximation for analysis and interpretation of planar contour figures, *Proc. Int. Conf. on Pattern Recognition-ICPR'90*, B: 865-869, 1990.

[58] van Dijk S., Thierens D., de Berg M., *Using Genetic Algorithms for Solving Hard Problems in GIS*, Technical Report, TR-2000-32, Utrecht University, 2000.

[59] Dijkstra E.W., A note on two problems in connexion with graphs, *Numerische Mathemaitk*, 1: 269-271, 1959.

[60] Dori D., Orthogonal Zig-Zag: An algorithm for vectorizing engineering drawings compared to Hough transform, *Advances in Engineering Software*, 28(1): 11-24, 1997.

[61] Dori D., Liu W., Sparse pixel vectorization: An algorithm and its performance evaluation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(3): 202-215, 1999.

[62] Dorigo M.,*Optimization, Learning, and Natural Algorithms*, PhD Thesis, Dip. Elettronica e Informazione, Politecnico di Milano, Italia, 1992.

[63] Dosch P., Tombre K., Ah-Soon C., Masini G., A complete system for analysis of architectural drawings. *Int. J. Document Analysis and Recognition*, 3(2): 102-116, 2000.

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

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

[66] Douglas D.H., Peucker T.K., Algorithm for the reduction of the number of points required to represent a line or its caricature, *The Canadian Cartographer*, 10 (2): 112-122, 1973.

[67] Duda R.O., Hart P.E., *Pattern Classification and Scene Analysis*, New-York: Wiley, 1972, pp.328-339.

[68] Dunham J., Optimum uniform piecewise linear approximation of planar curves, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(1): 67-75, 1986.

[69] Dunn M.E., Joseph S.H., Processing poor quality line drawings by local estimation of noise, *Proc. Int. Conf. on Pattern Recognition-ICPR'88*, 153-162, 1988.

[70] Eivkil L., Taxt T., Moen K., A fast adaptive method for binarization of document images, *Proc. Int. Conf. on Document Analysis and Recognition-ICDAR'9I*, 435-443, 1991.

[71] Espelid R., Jonassen I., A comparison of splitting methods for the identification of corner-points, *Pattern Recognition Letters*, 12: 79-83, 1991.

[72] Eu D., Toussaint G.T., On approximating polygonal curves in two and three dimensions, *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing*, 56 (3): 231-246, 1994.

[73] Fahn C.-S., Wang J.-F., Lee J.-Y., An adaptive reduction procedure for the piecewise linear approximation of digitized curves, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11: 967-973, 1989.

[74] Fishler M.A., Bolles R.C., Perceptual organization and curve partitioning, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8: 100-105, 1986.

[75] Fränti P., Ageenko E., Kolesnikov A., Chalenko I., Compression of line-drawing images using vectorizing and feature-based filtering, *Proc Int. Conf. on Computer Graphics and Visualization-GraphiCon'98*, Moscow, Russia, pp. 219-224, June 1998.

[76] Fränti P., Ageenko E., Kolesnikov A., Vectorizing and feature-based filtering for line-drawing image compression, *Pattern Analysis & Applications*, 2(4): 285-291, 1999.

[77] Fränti P., Ageenko P., Kälviäinen H., Kukkonen P., Compression of line-drawing images using Hough transform for exploiting global dependencies, *Proc. 4th. Int. Conf. on Information Sciences-JSIC'98*, 4: 433-436, 1998.

[78] Fränti P., Ageenko P., Kukkonen P., Kälviäinen H., Using Hough transformation for context-based image compression in hybrid raster/vector applications, *J. Electronic Imaging*, 11(2): 236-245, 2002.

[79] Freeman H., Davis L.S., A corner-fitting algorithm for chain-coded curves, *IEEE Trans. Computers*, 26: 297-303, 1977.

[80] Fujiwara A., Inoue M., Masuzawa T., Fujiwara H., A simple paralle algorithm for weighted distance transforms, *Proc. IEEE 2nd Int. Conf. on Algorithms and Architectures for Parallel Processing*, 1-8, 1996.

[81] Fujiwara A., Inoue M., Masuzawa T., Fujiwara H., A simple parallel algorithm for the medial axis transforms, *Proc. Int. Parallel Processing Symp*. 407-411, 1997.

[82] Garrido A., de la Blanca N.P., Garcia-Silvente M., Boundary simplification using a multiscale dominant-point detection algorithm, *Pattern Recognition,* 31: 791-804, 1998.

[83] Glasbey C.A., An analysis of histogram-based thresholding algorithm, *Graphical Models and Image Processing*, 55(6): 532-7, 1993.

[84] Glover F., Laguna M., *Tabu Search*, Kluwer Academic Publisher, 1997.

[85] Gluss B., Further remarks on line segment curve-fitting using dynamic programming, *Comm. ACM*, 5: 441-443, 1962.

[86] Goldberg D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, 1989.

[87] Gritzali F., Papakonstantinou G., A fast piecewise linear approximation algorithm, *Signal Processing*, 5: 221-227, 1983.

[88] Guo Z., Hall R.W., Parallel thinning with two-subiteration algorithms, *Comm. ACM*, 32(3): 359-373, 1989.

[89] Hall R.W., Fast parallel thinning algorithms: parallel speed and connectivity preservation, *Comm. ACM*, 32(1): 123-131, 1989.

[90] Haugland D., Heber J., Husøy J., Optimization algorithm for ECG compression, *Medical & Biological Engineering & Computing*, 35: 420-424, 1997.

[91] Haugland D., Heber J., Husøy J., Compression data by shortest path methods, in *Operations Research Proceedings 1996,* Springer, Berlin, 145-150, 1997.

[92] Hayat L., Naqvi A.A., Sandler M.B., Comparative evaluation of fast thinning algorithms on a multiprocessor architecture, *Image and Vision Computing*, 10: 210-218, 1992.

[93] Heber J., Haugland D., Husøy J., An efficient implementation of an optimal time-domain ECG coder, *Proc. IEEE Conf. on Engineering in Medicine and Biology*, 1384-1385, 1997.

[94] Heckbert P.S., Garland M., Survey of polygonal surface simpification algorithms, *Technical Report*, CS Dept., Carnegie Mellon University, 1997.

[95] Heijmans H., *Morphological Image Operators*, Academic Press, Boston, 1994.

[96] Held A., Abe K., Arcelli C., Towards a hierarchical contour description via dominant point detection, *IEEE Trans. on Systems, Man, and Cybernetics*, 24(6): 942-949, 1994.

[97] Hershberger J., Snoeyink J., Speeding up the Douglas-Peucker line simplification algorithm, *Proc. 5th Int. Symp. on Spatial Data Handling*, 134-143, 1992.

[98] Hershberger J., Snoeyink J., An O(n log n) implementation of the Douglas-Peucker algorithm for line simplification. In *Proc. 10th Annu. ACM Symp. on Comput. Geom.*, 383-384, 1994.

[99] Hershberger J., Snoeyink J., Cartografic line simplification and polygon CSG formulæ in O(n log* n) time. In *Computational Geometry: Theory and Applications*, 11(3-4): 175-185, 1998.

[100] Heydorn S., Widner P., Optimization and performance analysis of thinning algorithms on parallel computers, *Parallel Computing*, 17:17-27, 1991.

[101] Ho S.-Y., Chen Y.-C., An efficient evolutionary algorithm for accurate polygonal approximation, *Pattern Recognition*, 34: 2305-2317, 2001.

[102] Hori O., Tanigawa O., Raster-to-vector conversion by line fitting based on contours and skeletons, *Proc. Int. Conf on Document Analysis and Recognition-ICDAR'93*, pp.353-358, 1993.

[103] Horng J.-H., Improving fitting quality of polygonal approximation by using the dynamic programming technique, *Pattern Recognition Letters,* 23(4): 1657-1673, 2002.

[104] Horng J.-H., Li J.-T., An automatic and efficient dynamic programming algorithm for polygonal approximation of digital curves, *Pattern Recognition Letters*, 23(1-3): 171-182, 2002.

[105] Horst J.A., Beichl I., A simple algorithm for efficient piecewise linear approximation of space curves, *Proc. Int. Conf. Image Processing-ICIP'97*, 2: 744-747, 1997.

[106] Horst J., Beichl I., Efficient piecewise linear approximation of space curves using chord and arc length, *Proc. Conf. on Machine Vision-96*, 1996.

[107] Hosur P.I., Ma K.-K., Optimal algorithm for progressive polygonal approximation of discrete planar curves, *Proc. Int. Conf. Image Processing-ICIP'99*, 1: 16-20, 1999.

[108] Hu J., Yan H., Polygonal approximation of digital curves based on the principles of perceptual organization, *Pattern Recognition*, 30(5): 701-718, 1997.

[109] Huang S.-C., Sun Y.-N., Polygonal approximation using genetic algorithms, *Pattern Recognition*, 32: 1409-1420, 1999.

[110] Imai H., Iri M., Computational-geometric methods for polygonal approximations of a curve, *Computer Vision, Graphics and Image Processing*, 36: 31-41, 1986a.

[111] Imai H., Iri M., An optimal algorithm for approximating a piecewise linear function, *J. of Information Processing*, 9(3): 159-162, 1986b.

[112] Imai H., Iri M., Polygonal approximations of a curve (formulations and algorithms), in *Computational Morphology*, G.T.Toussaint, (Ed), 71-86, North-Holland, Amsterdam, 1988.

[113] Iñesta J.M., Buendia M., Sarti M.A., Reliable polygonal approximations of imaged real objects through dominant point detection, *Pattern Recognition*, 31: 685-697, 1998.

[114] Ishijama M., Chin S.B., Hostetter G.H., Sklansky, Scan-along polygonal approximation for data compression of electrocardiograms, *IEEE Trans. Biomedical Engineering*, 30(11): 723-729, 1983.

[115] Janssen R.D.T., *The Application of Model-based Image Processing to the Interpretation of Maps*, PhD Thesis, Delft University of Technology, The Netherlands, 1995.

[116] Janssen R.D.T., Vossepoel A.M., Adaptive vectorization of line drawing images, *Computer Vision and Image Understanding*, 65(1): 38-56, 1997.

[117] Jawahr C., Biswas P., Ray A., Analysis of fuzzy thresholding schemes, *Pattern Recognition*, 33(8): 1339-1349, 2000.

[118] Jenq, J.F., Sahni, S., Serial and parallel algorithms for the medial axis transform, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14: 1218-1224, 1992.

[119] Jin, X.C., Ong, S.H., Jajasooriah, A domain operator for binary morphological processing, *IEEE Trans. Image Processing*, 4(7): 1042-1046, 1995.

[120] Jolt C.M., Stewart S., Clint M., Perrott R.H., An improved parallel thinning algorithm, *Comm. ACM*, 29(3): 239-242, 1987.

[121] Kanungo T., Haralick R., Dori D., Understanding engineering drawings: A survey, *Proc. Int. Workshop on Graphics Recognition*, Pennsylvania, USA, pp.119-130, August 1995.

[122] Kasturi, R., O'Gorman, L., Document Image Analysis: A Bibliography, *Machine Vision and Applications*, 5, pp. 231-243, 1992.

[123] Kasturi R, Trivedi M.M., (Eds.), *Image Analysis Applications*, Marcell Dekker, New York, 1990.

[124] Kasturi R, Tombre K., (Eds.), *Graphics Recognition–Methods and Applications*, Lecture Notes in Computer Science, Vol. 107, pp.49-56, Springer-Verlag, 1996.

[125] Katsaggelos A.K., Kondi L.P., Meier F.W., Osterman J., Schuster G.M., MPEG-4 and rate-distortion-based shape-coding techniques, *Proc. of IEEE*, 86(6): 1126-1154, 1998.

[126] Kolesnikov A., Belekhov V., Chalenko I., Vectorization of the raster images, *Pattern Recognition and Image Analysis*, 6(4): 784-794, 1996.

[127] Kolesnikov A., Fränti P., A fast near-optimal *min*-# polygonal approximation of digitized curves, *Proc. IASTED Int. Conf. on Automation, Control and Information Technology-ACIT'02*, Novosibirsk, Russia, pp.418-422, June 2002.

[128] Kolesnikov A., Fränti P., A fast near-optimal algorithm for approximation of polygonal curves, *Proc Int. Conf. on Pattern Recognition-ICPR'02*, Quebec City, Canada, 4: 335-338, August 2002.

[129] Kolesnikov A., Fränti P., Polygonal approximation of closed contours, in *Lecture Notes in Computer Science*, Vol.2749, G.Goos, J.Hartmanis, J. van Leeuwen, (Eds.): Proc. of the Scandinavian Conf. on Image Analysis-SCIA'2003, Göteborg, Sweden, pp. 778-785, June 2003.

[130] Kolesnikov A., Fränti P., Reduced-search dynamic programming for approximation of polygonal curves, *Pattern Recognition Letters*, 24(14): 2243-2254, 2003.

[131] Kolesnikov A., Fränti P., Fast algorithm for multiple-objects *min*-ε *problem*, *accepted to Int. Conf. on Image Processing-ICIP'2003*, Barcelona, Spain, September 2003.

[132] Kolesnikov A., Fränti P., *Data Reduction of Large Vector Graphics*, Research Report, A-2003-2. 2003.

[133] Kolesnikov A., Trichina E., Parallel algorithm of image skeletonization, *Proc. 5th Workshop Distributed Processing of the Information*, Novosibirsk Russia, October 1995. (*in Russian*)

[134] Kolesnikov A., Trichina E., A parallel algorithm for thinning of binary images, *Optoelectronics, Instrumentation and Data Processing*, 6: 7-13, 1995.

[135] Kolesnikov A., Trichina E., Parallel implementation of width-independent fast thinning algorithm, *Proc. 1st Int. Conf. Visual Information Systems-VISUAL'96*, Melbourne, Australia, pp. 384-393, February 1996.

[136] Koski A., Juhola M., Segmentation of digital signals based on estimated compression ratio, *IEEE Trans. Biomedical Engineering*, 43(9): 928-938, 1996.

[137] Koskinen L., Astola J., Soft morphological filters: a robust morphological filtering method, *J. Electron Imaging*, 3: 60-70, 1994.

[138] Kossentini F., Smith M.J.T., A fast PNN design algorithm for entropy-constrained residual vector quantization, *IEEE Trans. Image Processing*, 7: 1045-1050*,* 1998.

[139] Ku K.M., Chiu K.M., Polygonal approximation of digital curve by graduate iterative merging, *Electronics Letters*, 31: 444-446, 1995.

[140] Kurozumi Y., Davis W.A., Polygonal approximation by the minimax method, *Computer Vision, Graphics and Image Processing*, 19: 248-264, 1982.

[141] Lam L., Lee S.-W., Suen C.Y., Thinning methodologies – A comprehensive survey, *IEEE Trans. on PAMI*, 14(9): 869-885, 1992.

[142] Lam L., Lee S.-W., Suen C.Y., A systematic evaluation of skeletonization algorithms, *Int. J. Pattern Recognition and Artificial Intelligence*, 7(5): 1203-1225, 1993.

[143] Latecki L.J., Lakämper R., Polygon evolution by vertex deletion, *Computer Vision and Image Understanding*, 73: 441-454, 1999.

[144] Lawson C.L., Characteristics properties of the segmentation rational minimax approximation problem, *Numer. Math.*, 6: 293-301, 1964.

[145] Lee K.H., Cho S.B., Choy Y.C., Automated vectorization of cartographic maps by a knowledge-based system, *Engineering Applications of Artificial Intelligence*, 13: 165-178, 2000.

[146] Lee J.-H., Chung J.-W., Kim J.-K., Optimal vertex adjustment method using a Viterbi algorithm for vertex-based shape coding, *Electronic Letters*, 35 (9): 706-707, 1999.

[147] Lee S.U., Chung S.Y., Park R.H., A comparative performance study of several global thresholding techniques for segmentation, *Computer Vision, Graphics, and Image Processing*, 52: 171-90, 1990.

[148] Lee Y.H., Horng S.J., Fast parallel chessboard distance transform algorithms, *Proc. Int. Conf. Parallel and Distributed Systems-ICPADS'96*, 488-493, 1996.

[149] Lee Y.H., Horng S.J., Optimal computing the chessboard transform on parallel processing systems, *Computer Vision and Image Understanding*, 73(3): 374-390, 1999.

[150] Leu J.G., Chen L., Polygonal approximation of 2d shapes through boundary merging, *Pattern Recognition Letters*, 7(4): 231-238, 1988.

[151] Lladós J., *Combining Graph Matching and Hough Transform for Hand-Drawn Graphical Document Analysis. Application to Architectural Drawings* , PhD Thesis, Universitat Autònoma de Barcelona, November 1997.

[152] Loce R.P., Mean-absolute-error representation and optimization of computational-morphological filters, *CVGIP: Graphical Model and Image Processing*, 57(1): 27-37, 1995.

[153] Manzanera A., Bernard T.M., Prêteux F., Longuet B, Ultra-fast skeleton based on the isotropic fully parallel algorithm, *Proc. of Discrete Geometry for Computer Imagery-DGCI'99*, Springer-Verlag, Lecture Notes in Computer Science, 1568: 313-324, 1999.

[154] Manzanera A., Bernard T.M., Metrical properties of a collection of 2D parallel thinning algorithms, *Proc. Int. Workshop on Combinatorial Image Analysis*, Palermo, Italy, May 2003, Electronic Notes on Discrete Mathematics, vol. 12, Elsevier Science, 2003.

[155] Marji M., Siy P., A new algorithm for dominant points detection and polygonalization of digital curves, *Pattern Recognition*, 2003. (in press)

[156] Mazzariol M., *Computer-Aided Parallelization of Applications*, Ph.D. Thesis, No 2431 Computer Science Department, Ecole Polytechnique Fédérale de Lausanne, Switzerland, 2001.

[157] Melkman A., O'Rourke J., On polygonal chain approximation, in *Computational Morphology*, G.T. Toussaint, (Ed), 87-95, North-Holland, Amsterdam, 1988.

[158] Mikheev A., Vincent L., Faber V., High-quality polygonal contour approximation based on relaxation, *Proc. Int. Conf. Document Analysis and Recognition-ICDAR'01*, 361-365, 2001.

[159] Milgram M., de Saint Pierre T., Boundary detection and skeletonization with a massively parallel architecture, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(1): 74-78, 1990.

[160] Monagan G., Roosli M., Appropriate base representation using a run graph, *Proc. Int. Conf. Document Analysis and Recognition-ICDAR'93*, Tsukuba, Japan, pp.623-626, 1993.

[161] Montanari U., A note on a minimal length polygonal approximation to digitized curve, *Comm. ACM*, 13: 41-47, 1970.

[162] Montoya M.D.G., Garcia I., Implementation of parallel thinning algorithms on multicomputers: analysis of the work load balance, *Proc. 6th Euromicro Workshop on Parallel and Distributed Processing*, 257-263, 1998.

[163] Mori K., Wada K., Toraichi K., Function approximated shape representation using dynamic programming with multi-resolution analysis, *Proc. Int. Conf. on Signal Process. and Applications & Technology-ICSPAT'99*, Orlando, USA, 1999.

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

[165] Neumann R., Teisseron G., Extraction of dominant points by estimation of the contour fluctuations, *Pattern Recognition*, 35: 1447-1462, 2002.

[166] Niblack N., *An Introduction to Image Processing*, Prentice Hall, pp. 115-116, 1986.

[167] Niblack C.W., Gibbons P.B., Capson D.W., Generating skeletons and centerlines from the distance transform, *Computer Vision, Graphics and Image Processing: Graphical Models and Image Processing*, 54(5): 420-437, 1992.

[168] Nikiel S., Noise suppression based on the fractal dimension estimates, *Proc. Int. Conf. on Image Processing-ICIP'96*, Losanne, Switzerland, pp. 193-196, 1996.

[169] Novikov Yu.L., *An efficient algorithms for raster images vectorization and its realization in GIS*. PhD Thesis, Tomsk State University, Russia. (*in Russian*)

[170] Nygaard R, *Shortest Path Methods in Representation and Compression of Signals and Image Contours*. PhD Thesis, The Norwegian Institute of Technology, July 2000.

[171] Nygaard R., Haugland D., Multiple error measure in ECG data compression, *Proc. Norwegian Signal Processing Symp.*, Tromsø, Norway, 134-139, 1997.

[172] Nygaard R., Melnikov G., Katsaggelos A.K., Rate-distortion optimal ECG coding algorithm, *Proc. Int. Conf. Image Processing-ICIP'99*, Kobe, Japan, 348-351, October 1999.

[173] Nygaard R., Melnikov G., Katsaggelos G., A rate-distortion optimal ECG coding algorithm, *IEEE Trans. Biomedical Engineering*, 48(1): 28-40, 2001.

[174] Nygaard R., Husøy J., Haugland D., Compression of image contours using combinatorial optimization, *Proc Int. Conf. Image Processing-ICIP'98*, 1: 266-270, 1998.

[175] Ogier J.M., Mullot R., Labiche J., Lecourtier Y., Multilevel approach and distributed consistency for technical map interpretation: Application to cadastral maps, *Computer Vision and Image Understanding*, 70(3): 438-451, 1998.

[176] O'Gorman L., Kasturi R., (Eds.), *Document Image Analysis*, IEEE Computer Society, Los Alamitos, California, 1994.

[177] Otsu N., A thresholding selection method from gray-level histogram, *IEEE Trans. on Systems, Man, and Cybernetics*, 9(1): 62-66, 1979.

[178] Pal S.K., Pal S.K., A review on image segmentation techniques, *Pattern Recognition*, 26(9): 1277-94, 1993.

[179] Papakonstantinou G., Optimal polygonal approximation of digital curves, *Signal Processing*, 8: 131-135, 1985.

[180] Papakonstantinou G., Tsanakas P., Manis G., Parallel approaches to piecewise linear approximation, *Signal Processing*, 37: 415-423, 1994.

[181] Park R.-H., Jee Y.H.., Multistep polygonal approximation, *J. Electronic Imaging*, 3(3): 232-244, 1994.

[182] Pavlidis T.,*Structural Pattern Recognition*, Springer, Berlin, 1977.

[183] Pavlidis T., A vectorizer and feature extractor for document recognition, *Computer Vision, Graphics and Image Processing*, 35: 111-127, 1986.

[184] Pavlidis T., Horovitz S.L., Segmentation of plane curves, *IEEE Trans. Computers*, 23(8): 860-870, 1974.

[185] Pei S.-C., Horng J.-H., Optimum approximation of digital planar curves using circular arcs, *Pattern Recognition*, 3: 383-388, 1996.

[186] Perez J.C., Vidal E., Optimum polygonal approximation of digitized curves, *Pattern Recognition Letters*, 15: 743-750, 1994.

[187] Pfaltz J.L., Rosenfeld A., Computer representation of planar regions by their skeletons, *Comm. ACM*, 10: 119-125, 1967.

[188] Phillips Y., Rosenfeld A., An ISODATA algorithm for straight line fitting, *Pattern Recognition Letters*, 7: 291-297, 1988.

[189] Pikaz A., Dinstein I., Optimal polygonal approximation of digital curves, *Pattern Recognition*, 28(3): 371-379, 1995.

[190] Pikaz A., Dinstein I., An algorithm for polygonal approximation based on iterative point elimination, *Pattern Recognition Letters*, 16 (6): 557-563, 1995.

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

[192] Piper J, Efficient implementation of skeletonization using interval coding, *Pattern Recognition Letters*, 3: 389-2397, 1985.

[193] Poty V., Miguet S., A new 2-D and 3-D thinning algorithms based on successive border generations, *Research Report*, No.94-20, Ecole Normale Superieure de Lyon, Lyon, France, 1994.

[194] Poty V., Miguet S., Data allocation strategies for parallel image processing algorithms, *Int. J. Pattern Recognition and Artificial Intelligence*, 9(4): 615-634, 1995.

[195] Poty V., Ubeda V., A parallel thinning algorithm using $k \times k$ masks, *Int. J. Pattern Recognition and Artificial Intelligence*, 7: 1183-1202, 1993.

[196] Prasanna V.K., Wang L., Scalable parallel implementations of perceptual grouping on connection machine CM-5, *Proc. Int. Conf. on Pattern Recognition-ICPR'94*, 229-233, 1994.

[197] Pridmore T.P., Ablameyko SV., A generalized distance transform for line patters, *Pattern Recognition and Image Analysis*, 6: 545-554, 1996.

[198] Ragnemalm I, Ablameyko S.V., On the distance transform of line patterns, *Proc. Scandinavian Conf. on Image Analysis*, 1357-1363, 1993.

[199] Ramer U., An iterative procedure for polygonal approximation of plane curves, *Computer Graphics and Image Processing*, 1: 244-256, 1972.

[200] Randolph T.R., Smith M.J.T., Enhancement of fax documents using a binary angular representation, *Proc. Int. Symp. on Intelligent Multimedia, Video and Signal Processing*, 125-128, 2001.

[201] Ramer U., An iterative procedure for the polygonal approximation of plane curves, *Computer Graphics and Image Processing*, 1: 244-256, 1972.

[202] Ray B.K., Ray K.S., An algorithm for detection of dominant points and polygonal approximation of digitized curves, *Pattern Recognition Letters*, 13: 849-856, 1992.

[203] Ray B.K., Ray K.S., Detection of significant points and polygonal approximation of digitized curves, *Pattern Recognition Letters*, 13: 443-452, 1992.

[204] Ray B.K., Ray K.S., Determination of optimal polygon from digital curve using L_1 norm, *Pattern Recognition*, 26(4): 505-509, 1993.

[205] Ray B.K., Ray K.S., A non-parametric sequential method for polygonal approximation of digital curves, *Pattern Recognition Letters*, 15: 161-167, 1994.

[206] Ray B.K., Ray K.S., A new split-and-merge technique for polygonal approximation of chain coded curves, *Pattern Recognition Letters*, 16: 161-169, 1995.

[207] Recatalá G., Iñesta J.M., Polygonal approximation through genetic algorithms, *Proc. 11th Scandinavian Conf. on Image Analysis*, 707-714, June 1999.

[208] Reumann K., Witkam A.P.M., Optimizing curve segmentation in computer graphics, *Proc. Int. Comput. Sympos.*, A.Gunther, B.Levrat and H.Lipps, (Eds.), New-York: Elsevier, 467-472, 1974.

[209] Rhee P.K., Lee C.W., Performance evaluation of parallel thinning algorithms based on PRAM model, *Proc. Int. Conf. on Parallel and Distributed Methods for Image Processing*, *Proceedings SPIE*, Vol. 3166, 1997.

[210] Roberge J., A data reduction algorithm for plain curves, *Computer Vision*, *Graphics and Image Processing*, 29: 128-195, 1985.

[211] Röösli M., Monagan G., A high quality vectorization combining local quality measures and global constraints, *Proc. Int. Conf. on Document Analysis and Recognition-ICDAR'95*, 243-248, 1995.

[212] Rosenfeld A., Johnston E., Angle detection on digital curves, *IEEE Trans. Computers*, 22: 875-878, 1973.

[213] Rosenfeld A., Pfaltz J.L., Sequential operations in digital picture processing, *J. Ass. Comput. Math.*, 13: 471-494, 1966.

[214] Rosenfeld A., Pfaltz J.L., Distance functions in digital pictures, *Pattern Recognition*, 1: 33-61, 1968.

[215] Rosenfeld A., Weszka J.S., An improved method of angle detection on digital curves. *IEEE Trans. Computers*, 24(9): 940-941, 1975.

[216] Rosin P.L., Techniques for assessing polygonal approximations of curves, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14 (6): 659-666, 1997.

[217] Rosin P.L., Assessing the behaviour of polygonal approximation algorithms, *Pattern Recognition*, 36: 505-518, 2003.

[218] Rosseel M., Comments on paper by Saigal: A constrained shortest route problem, *Operations Research*, 16: 1232-1234, 1968.

[219] Rutkowski W.S., Peleg S., Rosenfeld A., Shape segmentation using relaxation, *IEEE Trans. Pattern Analysis and Machine Intelligence*: 368-375, 1981.

[220] Saigal R., A constrained shortest route problem, *Operations Research*, 16: 205-209, 1968.

[221] Sahoo P.K., Soltani S., Wang A.K.C., A survey of thresholding techniques, *Computer Vision, Graphics, and Image Processing*, 41: 233-60, 1988.

[222] Salotti M., Improvement of Perez and Vidal algorithm for the decomposition of digitized curves into line segments, *Proc. Int. Conf. on Pattern Recognition-ICPR'00*, 2, 882-886, 2000.

[223] Salotti M., An efficient algorithm for the optimal polygonal approximation of digitized curves, *Pattern Recognition Letters*, 22: 215-221, 2001.

[224] Salotti M., Un algorithme efficiace pour l'approximation polygonale optimale, *Proc. 13ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle-RFIA'2002*, Angers, France, 1: 11-18, 2002.

[225] Salotti M., Optimal polygonal approximation of digitized curves using the sum of square deviations criterion, *Pattern Recognition*, 35(2): 435-443, 2002.

[226] Sanchiz J.M., Iñesta J.M., Pla F., A neural network-based algorithm to detect dominant points from the chain-code of a contour, *Proc. Int. Conf. on Pattern Recognition-ICPR'98,* 4: 325-329. 1998.

[227] Sanchiz J.M., Pla F., Iñesta J.M., Using neural networks to detect dominant points in chain-coded contours, *Int. J. Pattern Recognition and Artificial Intelligence*, 12: 661-675, 1998.

[228] Sankar P.V., Sharma C.V., A parallel procedure for detection of dominant points on digital curve, *Comput. Graphics and Image Processing*, 7: 403-412, 1978.

[229] Sarkar D., A simple algorithm for detection of significant vertices for polygonal approximation of chain-coded curves, *Pattern Recognition Letters*, 14: 959-964, 1993.

[230] Sato Y., Piecewise linear approximation of planar curves by perimeter optimization, *Pattern Recognition*, 25(12): 1535-1543, 1992.

[231] Sauvola J., *Document Analysis Techiques and System Components with Applications in Image Retrieval*, PhD Thesis, University of Oulu, Oulu, Finland, 1997.

[232] Sauvola J., Pietikäinen M., Adaptive document image binarization, *Pattern Recognition*, 33: 225-236, 2000.

[233] Schwarzkopf O., Parallel computation of distance transforms, *Algorithmica*, 6: 685-697, 1991.

[234] Schonfeld D., Goutsias J., Optimal morphological pattern restoration from noisy binary images, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(1): 14-29, 1991.

[235] Schroeder K., Laurent P., Efficient polygon approximations for shape signatures, *Proc. Int. Conf. on Image Processing-ICIP'99*, 2: 811-814, 1999.

[236] Schuster G.M., Katsaggelos A.K., An optimal polygonal boundary encoding scheme in the rate-distortion sense, *IEEE Trans. Image Processing*, 7(1): 13-26, 1998.

[237] Schuster G.M., Melnikov G., Katsaggelos A.K., Operationally optimal vertex-based shape coding, *IEEE Signal Processing Magazine*, 15 (6): 91-108, 1998.

[238] Serra J., *Image Analysis and Mathematical Morphology*, Academic Press, London, 1992.

[239] Sharaiha Y.M., Christofides N., An optimal algorithm for the straight segment approximation of digital arcs, *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing*, 55(5): 397-407, 1993.

[240] Shih C.C., Pu C.C., A skeletonization algorithm by maxima tracking on Euclidean distance transform, *Pattern Recognition*, 28(3): 331-341, 1995.

[241] Sklansky J., Chasin R.L., Hansen B.J., Minimum perimeter polygons of digitized silhouettes. *IEEE Trans. Computers*, 23: 1355-1364, 1972.

[242] Sklansky J., Gonzalez V., Fast polygonal approximation of digitized curves. *Pattern Recognition*, 12: 327-331, 1980.

[243] Sohn K., Kim J.H., Alexander W.E., Mean field annealing approach to robust corner detection, *Trans. System, Man and Cybernetics*, (B28), 82-90, 1998.

[244] Song J. *Research on Progressive-Simplification Based Vectorization Model and Graphics Recognition Methods*, PhD Thesis, Nanjing University, Nanjing, June, 2001.

[245] Song J., Su F., Chen J., Cai S., A knowledge-aided line network oriented vectorization method for engineering drawings, *Pattern Analysis and Applications*, 3: 142-152, 2000.

[246] Song J., Su F., Li F., Cai S., Raster-to-vector conversion of construction engineering drawings, *Automation in Construction*, 11: 597-605, 2002.

[247] Song J., Su F., Tai C.-L., Cai S., An object-oriented progressive simplification-based vectorization system for engineering drawings: Model, algorithm, and performance, *IEEE Trans Pattern Analysis Machine Intelligence*, 24(8): 1048-1060, 2002.

[248] Stefanelli R., Rosenfeld A., Some parallel algorithms for digital pictures, *J. of ACM*, 18: 255-264, 1971.

[249] Stone H., Approximation of curves by line segments, *Math. Comput.*, 15: 40-47, 1961.

[250] Suen C.Y., Wang P.S.P., (Eds.) Special issue on thinning methodologies for pattern recognition, *Int. J. Pattern Recognition and Artificial Intelligence*, 7(5): 965-1308, 1993.

[251] Sun Y.-N., Huang S.-C., Genetic algorithms for error-bounded polygonal approximation, *Int. J. Pattern Recognition and Artificial Intelligence*, 14(3): 297-314, 2000.

[252] Suzuki S., Abe K., Sequential thinning of binary pictures using distance transformation, *Proc. Int. Conf. on Pattern Recognition-ICPR'86*, 289-292, 1986.

[253] .Suzuki S., Abe, Binary pictures thinning by an iterative parallel two-sub-cycle operations. *Pattern Recognition*, 10(3): 297-307, 1987.

[254] Taxt T., Flynn P.J, Jain A.K., Segmentation of document images, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(12): 1322-29, 1989.

[255] Teh C.H., Chin R.T., On the detection of dominant points on digital curves, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(8): 859-872, 1989.

[256] Tombre K., Analysis of engineering drawings: State of the art and challenges. In K. Tombre and A.K. Chhabra (Eds.), *Graphics Recognition – Algorithms and Systems*, *Lecture Notes in Computer Science*, Vol. 1389, 257-264, Springer-Verlag, 1998.

[257] Tombre K., Ah-Soon C., Dosch P., Habed A., Masini G., Stable, robust and off-the-shelf methods for graphics recognition, *Proc. Int. Conf. on Pattern Recognition-ICPR'98*, 1: 406-408, 1998.

[258] Tombre K., Ah-Soon C., Dosch P., Masini G., Tabbone S., Stable and robust vectorization: How to make the right choices. In A.K. Chhabra and D. Dori, (Eds.), *Graphics Recognition–Recent Advances*, Lecture Notes in Computer Science, Vol.1941, 3-19. Springer Verlag, 2000.

[259] Tombre K., Chhabra A.K. (Eds.), *Graphics Recognition– Algorithms and Systems*, Lecture Notes in Computer Science, Vol. 1389, 257-264, Springer-Verlag, 1998.

[260] Toriwaki J.I., Yokoi S., Distance transformations and skeletons of digitized pictures with applications, in *Progress in Pattern Recognition*, 1: 187-264, North Holland, Amsterdam, 1981.

[261] Toussaint G.T., On the complexity of approximating polygonal curves in the plane, *Proc. IASTED Int. Symp. on Robotics and Automation*, Lugano, Switzerland, 1985.

[62] Traver V.J., Recatalá G., Iñesta J.M., Exploring the performance of genetic algorithms as polygonal approximators, *Proc. Int. Conf. on Pattern Recognition-ICPR'2000*, 3: 770-773, 2000.

[262] Trier Ø.D., *A Data Capture System for Hydrographic Maps*, PhD Thesis, University of Oslo, Norway, 1996.

[263] Trier Ø.D., Jain A.K., Goal-directed evaluation of binarization methods, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17: 1191-1201, 1995.

[264] Trier Ø.D., Taxt T., Evaluation of binarization methods for document images, *IEEE Trans. Pattern Analysis and Machine Intelligence*17: 312-15, 1995.

[265] Tseng C.-C., Juan C.-J., Chang H.-C., Lin J.-F., An optimal line segment extraction algorithm for online Chinese character recognition using dynamic programming, *Pattern Recognition Letters*, 19: 953-961, 1998.

[266] Turner K.J., *Computer Perception of Curved Objects Using a Television Camera*, PhD Thesis, U. of Edinburgh, Scotland, 1974.

[267] Ubeda S., Pyramidal thinning algorithm for SIMD parallel machines, *Pattern Recognition*, 12: 1993-2000, 1995.

[268] Vallone U., Bidemensional shapes polygonalization by ACO, *Proc. 3rd Int. Workshop on Ants Algorithms*, Brussels, Belgium, 296-297, September 2002.

[269] Ventura J.A., Chen J.M., Segmentaion of two-dimensional curve contours, *Pattern Recognition*, 25(10):1129-1140, 1992.

[270] Visvalingam M., Whyatt J., The Douglas-Peucker algorithm for line simplification: re-evaluation through visualisation, *Computer Graphics Forum*, 9(3): 213-228, 1990.

[271] Visvalingam M., Whyatt J., Cartographic algorithms: problems of implementation and evaluation and the impact of digitizsing errors, *Computer Graphics Forum*, 10(3): 225-235, 1991.

[272] Visvalingam M., Whyatt J., Line generalization by repeated elimination of points, *Cartographic Journal*, 30(1): 46-51, 1993.

[273] Vossepoel A.M., Schutte K., Delanghe C., Memory efficient skeletonisation of utility maps, *Proc. Conf. on Document Analysis and Recognition- ICDAR'97*, Ulm, Germany, pp.797-800, August 1997.

[274] Wall K., Danielsson P.E., A fast sequential method for polygonal approximation of digitized curves, *Comput. Vision, Graphics and Image Process*, 28: 220-227, 1984.

[275] Wan W., Ventura J.A., Segmentation of planar curves into straight-line segments and elliptical arcs, *Graphics Model and Image Processing*, 59: 484-494, 1997.

[276] Wang P.S.P., Zhang Y.Y., A fast and flexible thinning algorithm, *IEEE Trans. Comput.*, 38(5): 741-745, 1989.

[277] Wenyin L., Dori D., From raster to vectors: Extracting visual information from line drawings, *Pattern Analysis & Applications*, 22: 10-21, 1999.

[278] Weszka J.S., Survey of threshold selection techniques, *Computer Graphics and Image Processing*, 7(2): 259-265, 1978.

[279] Weszka J.S., Nagel R.N., Rosenfeld A., A threshold selection technique, *IEEE Trans. Computers*, 23: 1322-1326, 1974.

[280] Williams C.M., An efficient algorithm for the piecewise linear approximation of planar curves, *Computer Graphics and Image Processing*, 8: 286-293, 1978.

[281] Wilson G.V., Assessing the usability of parallel programming systems: The Cowichan problems, *Proc. IFIP WG10.3 Working Conf. Programming Environments for Massively Parallel Distributed Systems*, Monte Verita, Ascona, Switzerland, April 1994.

[282] Wilson G.V., Bal H.E., An empirical assessment of the usability of Orca using the Cowichan problems, *IEEE Trans. Parallel and Distributed Technology*, 4(3): 36-44, 1996.

[283] Wu W.-Y., An adaptive method for detecting dominant points, *Pattern Recognition*, 36(10): 2231-2237, 2003.

[284] Wu J.-S., Leou J.-J., New polygonal approximation schemes for object shape representation, *Pattern Recognition*, 26: 471-484, 1993.

[285] Wu R.-Y., Tsai W.-H., A new one-pass parallel thinning algorithm for binary images, *Pattern Recognition Letters*, 13: 715-723, 1992.

[286] Wu W.-Y., Wang, Detecting the dominant points by the curvature-based polygonal approximation, *Computer Vision, Graphics and Image Processing: Graphical Models and Image Process*. 55: 79-88, 1993.

[287] Wuescher D.M., Boyer K.L., Robust contour decomposition using a constant curvature criterion, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(1): 41-51, 1991.

[288] Xiao Y., Zou J.J., Yan H., An adaptive split-and-merge method for binary image contour data compression, *Pattern Recognition Letters*, 22: 299-307, 2001.

[289] Xu C. C., Lau F.C.M., Decentralized remapping of data parallel computations with the generalized dimension exchange method, *Proc. 1994 Scalable High-Performance Computing Conf.*, 411-421, May 1994.

[290] Xu C.- C., Lau F.C.M., *Load Balancing in Parallel Computers: Theory and Practice*, Kluwer Academic Publisher, 1997.

[291] Xu C. C., Lau F.C.M., Diekmann R., Decentralized remapping of data parallel applications in distributed memory multiprocessors, *Concurrency: Practice and Experience*, 9(12): 1351-1376, 1997.

[292] Yin P.-Y., Algorithms for straight-line fitting using *k*-means, *Pattern Recognition Letters*, 19: 31-41, 1998.

[293] Yin P.-Y., A new method for polygonal approximation using genetic algorithms, *Pattern Recognition Letters* 19: 1017-1026, 1998.

[294] Yin P.-Y., Genetic algorithms for polygonal approximation of digital curves, *Int. J. Pattern Recognition and Artificial Intelligence* 13:1061-1082, 1999.

[295] Yin P.-Y., A tabu search approach to the approximation of digital curves, *Int. J. Pattern Recognition and Artificial Intelligence* 14: 243-255, 2000.

[296] Yin P.-Y., Ant colony search algorithms for optimal approximation of plane curve, *Pattern Recognition*, 36(8): 1783-1797*,* 2003.

[297] Yokoi S., Toriwaki J.I., Fukumura J.I., An analysis of topolgical properties of digital binary pictures using local features, *Comput. Graphics and Image Processing*, 4:63-73, 1975.

[298] Yu Y., *A System for Engineering Flow Drawing Understanding*, PhD Thesis, Univ. Nebraska-Lincoln, August 1994.

[299] Yun B.-J., Lee S.-W., Kim S.-D., Vertex adjustment method using a geometric constraint for polygon-based shape coding, *Electronic Letters*, 37 (9): 754-755, 2001.

[300] Zhang H., Guo J., Optimal polygonal approximation of digital planar curves using meta-heuristics, *Pattern Recognition*, 34, 1429-1436, 2001.

[301] Zhang, Mahgoub, A parallel VLSI-implementable thinning algorithm, *Proc. 8th Int. Conf. on Parallel and Distributed Computing Systems*, 254-S, 1995.

[302] Zhang T.Y., Suen C.Y, A fast parallel algorithm for thinning digital patterns, *Comm. ACM*, 27(3): 236-239, 1984.

[303] Zhang Y.Y., Wang P.S.P., A new parallel thinning methodology, *Int. J. Pattern Recognition and Artificial Intelligence*, 8: 999-1011, 1994.

[304] Zhao Z., Saalfeld A., A linear-time sleeve-fitting polyline simplification algorithms. *Proc. 13th Int. Conf. AutoCarto, ACSM/ASPRS'97 Technical Papers,* 5: 214-223, 1997.

[305] Zheng Q., Kanungo T., *Morphological degradation models and their use in document image restoration*, University of Maryland, USA, Technical Report, LAMP-TR-065 CAR-TR-962 N660010028910/IIS9987944, 2001.

[306] Zhou X., Wei J., Li F., Woo P.Y., A new algorithm for parallel thinning and its hardware realization, *Int. J. Imaging Systems and Technology*, 10(4): 318-322, 1999.

[307] Zhu P., Chirlian P.M., On critical point detection of digital shapes, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17: 737-748, 1995.

[308] Zhu Y., Seneviratne L.D., Optimal polygonal approximation of digitized curves, *IEE Proc.-Vis. Image Signal Process*, 144(1): 8-14, 1997.