A Survey of Digital Map Processing Techniques

YAO-YI CHIANG, University of Southern California
STEFAN LEYK, University of Colorado, Boulder
CRAIG A. KNOBLOCK, University of Southern California

Maps depict natural and human-induced changes on earth at a fine resolution for large areas and over long periods of time. In addition, maps—especially historical maps—are often the only information source about the earth as surveyed using geodetic techniques. In order to preserve these unique documents, increasing numbers of digital map archives have been established, driven by advances in software and hardware technologies. Since the early 1980s, researchers from a variety of disciplines, including computer science and geography, have been working on computational methods for the extraction and recognition of geographic features from archived images of maps (digital map processing). The typical result from map processing is geographic information that can be used in spatial and spatiotemporal analyses in a Geographic Information System environment, which benefits numerous research fields in the spatial, social, environmental, and health sciences. However, map processing literature is spread across a broad range of disciplines in which maps are included as a special type of image. This article presents an overview of existing map processing techniques, with the goal of bringing together the past and current research efforts in this interdisciplinary field, to characterize the advances that have been made, and to identify future research directions and opportunities.

Categories and Subject Descriptors: A.1 [**Introduction and Survey**]; H.2.8 [**Database Management; Database Applications**]: Spatial Databases and GIS

General Terms: Design, Algorithms

Additional Key Words and Phrases: Map processing, geographic information systems, image processing, pattern recognition, graphics recognition, color segmentation

ACM Reference Format:
Yao-Yi Chiang, Stefan Leyk, and Craig A. Knoblock. 2014. A survey of digital map processing techniques. ACM Comput. Surv. 47, 1, Article 1 (April 2014), 44 pages.
DOI: http://dx.doi.org/10.1145/2557423

1. INTRODUCTION

This article presents an overview of the techniques for *digital map processing* (or simply *map processing*), which refers to computational procedures aimed at the automatic or semiautomatic extraction and/or recognition of geographic features contained in images (usually scanned) of maps. Digital map processing is a relatively young research field that grew out of image processing, document analysis, graphics recognition, and digital cartography. Over the past 40 years, researchers have become increasingly

Authors' addresses: Yao-Yi Chiang, University of Southern California, Spatial Sciences Institute, 3616 Trousdale Parkway, AHF B55, Los Angeles, CA 90089-0374; email: yaoyic@usc.edu; Stefan Leyk, University of Colorado at Boulder, Department of Geography, 260 UCB, Boulder, Colorado 80309-0260, USA; email: stefan.leyk@colorado.edu; Craig A. Knoblock, University of Southern California, Information Sciences Institute and Department of Computer Science, 4676 Admiralty Way, Marina del Rey, CA 90292; email: knoblock@isi.edu.

DOI: http://dx.doi.org/10.1145/2557423

interested in the methodological aspects of computational map processing (mostly in scanned maps) for the purposes of retrieval, extraction, and integration of geographic data [Freeman and Pieroni 1982]. This increasing attention results not only from the parallel advances in technologies (e.g., digital image analysis, recognition, and Geographic Information Systems [GIS]) but also can certainly be linked to the fact that computational map processing methods enable the preservation of unique (historical) maps and the utilization of the contained geographic information in modern analytical environments (e.g., in a GIS).

Maps can cover large areas over long periods of time for many regions in the world. This makes maps unique documents witnessing places, human activities, and natural features in the past for which no or only limited alternative information sources exist. Processing map images to extract and recognize geographic information results in spatially referenced data (i.e., map data) that can then be accessed, processed, and maintained in a GIS environment (in other words, "unlocking" geographic information from map documents). Incorporating map data into a GIS environment (i.e., map data can be used for spatial analyses and overlaid with other spatial data) creates unprecedented opportunities for multitemporal and multicontextual spatial analyses, such as analyzing the changes in built-up areas over large regions across long time periods, and investigating how these changes interact with other geographic features, such as vegetation or wetland areas. In addition, digital map processing can expedite the processes of comparing map contents from different map series/editions (e.g., contemporary maps vs. historical maps), to update current map series, and to create thematic maps or new map series.

The need for computational solutions with higher degrees of automation for map processing becomes evident if one considers that millions of map documents have already been scanned and stored in digital archives. For example, the U.S. Geological Survey (USGS) continues to scan and release all editions of more than 200,000 historic topographic map pages of the United States that cover the time period from 1884 to 2006. The GIS Center in Academia Sinica, Taiwan, has scanned and archived more than 160,000 historical maps. Such digital archives can only be fully used after the archived maps have been converted to a GIS usable format. However, manually processing these maps not only generates nonreproducible data that can suffer from a high degree of inaccuracy and introduced subjectivity but also does not scale well for handling large numbers of maps. For example, manually digitizing one sounding label on a nautical chart includes two steps: drawing a minimum bounding box to label the sounding location and typing in the sounding value. If these two steps take a total of 6 seconds per sounding label, for a typical nautical chart that has more than 4,000 sounding labels, the digitization process takes more than 6 hours. In contrast, using digital map processing techniques can dramatically reduce the time (and cost) to fully utilize the geographic information locked in these maps. For example, Chiang and Knoblock [2011] developed a map processing package, Strabo, which only requires 1 minute of the user's time for operating the software to recognize 1,253 labels from an area in a nautical chart with 83% precision and 80% recall. For this particular area in the nautical chart, the amount of user time for verifying and correcting the recognition results from Strabo is less than 1 hour, which is around 30% of the time that would be needed when performing the entire text recognition task manually. Without such computational solutions for map processing, large portions of the spatial data in maps remain inaccessible unless they are manually converting to spatial datasets.

General techniques for document analysis and graphics recognition (e.g., Cordella and Vento [2000], Lladós et al. [2002]) cannot be directly applied to map processing because maps often pose particular difficulties for recognition due to various graphical quality issues and the complexity of the map contents. The graphical quality of scanned

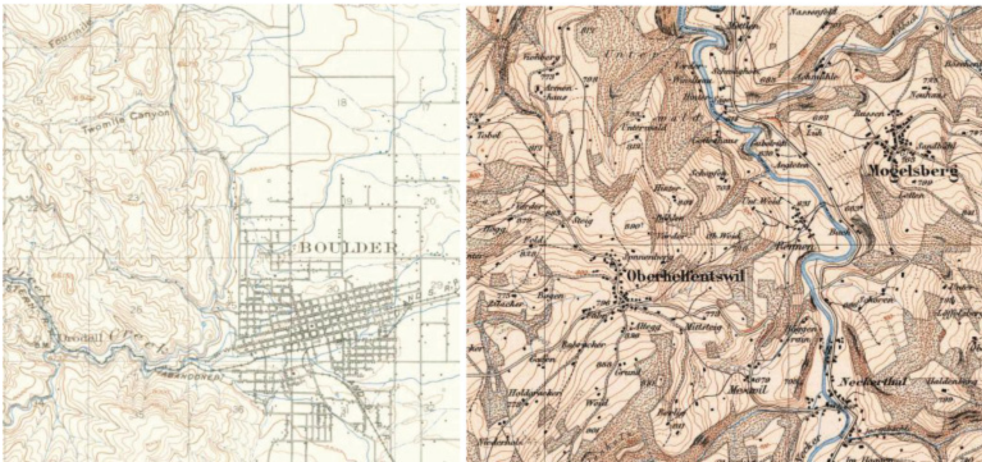


Fig. 1. Subsections of two examples of scanned (historical) maps of inferior graphical quality due to aging and bleaching effects: USGS topographic map of Boulder, Colorado (1902), at a scale of 1:62,500 (left), and Swiss National Topographic map (Siegfried map), page 220, Brunnadern (1879) at a scale of 1:25,000. Reproduced by permission of swisstopo (BA13123).

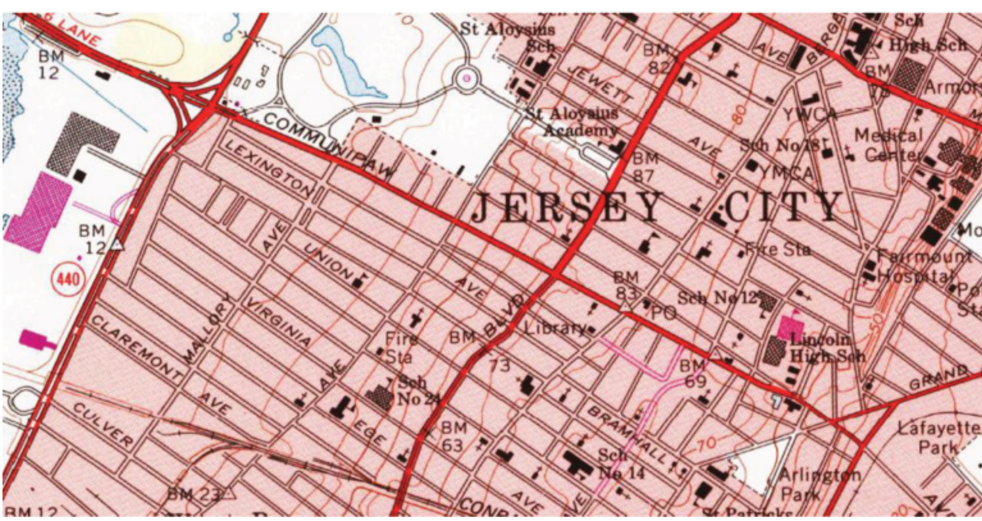


Fig. 2. An example map subsection that has multioriented labels: USGS topographic map of Jersey City, New Jersey (1967), at a scale of 1:24,000. Credit: U.S. Geological Survey.

maps can be affected by scanning or image compression processes. In addition, the stored and archived map materials suffer from aging and bleaching effects, whereas original reproduction materials (e.g., copper plates) have often been destroyed or lost. The final scanned images inherit these graphical properties. Figure 1 shows examples of two scanned historical maps of inferior graphical quality due to the imperfections (including manually drawn features) of the original maps. Furthermore, as seen in the sample map in Figure 2, layers of geographic features, such as roads, contour lines, and labels, often overlap with each other, which increases the degree of content complexity and color mixing.

Various types of maps were produced manually until recently (e.g., using copper plate or stone engraving techniques), resulting in high variations of symbol appearances and colors occurring on papers and in the final scans. Graphical guidelines for maps are often map specific and rule based but usually do not satisfy the quality requirements for machine drawings most often described and processed in document analysis and recognition work. This impedes processing map images within a general document recognition framework. For example, state-of-the-art commercial optical character recognition (OCR) tools can achieve high recognition rates in documents containing text lines of the same orientation, but recognizing map labels (text) in scanned maps is still challenging [Nagy et al. 1997; Chiang and Knoblock 2011]. The main reason is that map labeling is based on specific underlying semantics and follows cartographic rules, which results in map labels of varying orientations that can appear curved and can often overlap with other graphical elements.

Although outcomes are of the same type, extraction and recognition techniques used in digital map processing are significantly different from the techniques used in remote sensing, document analysis, and other recognition applications. For example, road extraction from scanned maps has to deal with fundamentally different input data (i.e., manmade graphics) than road extraction from satellite imagery where photography-type or reflectance data are used. Accordingly, road extraction from scanned maps aims at removing the noise from scanning, compression, and imperfection of the original material, accurately detecting and converting the pixel-based road information carefully depicted by cartographers into vector format [Bin and Cheong 1998; Itonaga et al. 2003; Bucha et al. 2006; Chiang and Knoblock 2013]. In comparison, road extraction from satellite imagery identifies and extracts features from spectral reflectance data that could cartographically be represented as road areas and road centerlines in a GIS [Hickman et al. 1995; Steger et al. 1997; Hodgson et al. 2004].

Due to the impediments and challenges in map processing presented earlier, most approaches described in the literature are rather specific to a particular series of maps and a lack of more generic solutions still persists. One significant consequence is that, to date, research on map processing shows slower rates of progress in developing general, automated, and robust solutions when compared to related fields in image processing and pattern recognition. Interestingly, the map processing literature appears to be abundant but highly dispersed across a range of fields such as image processing, document analysis and recognition, machine learning, data integration, and geoinformatics/digital cartography. Although this is an interesting development, it also reflects an increasing need to bring together existing efforts to provide a thorough overview of what has been done in various fields and show how these efforts are related. Such an overview will help establish an objective outlook of the most promising research directions to increase the rate of progress in the field of digital map processing. As such, this survey synthesizes existing research in map processing and presents an overview that covers the diverse disciplines where this research has been published. Research on map processing greatly benefits from interdisciplinary approaches using the strengths of disciplines that naturally participate in this area, such as computer science, Geographic Information Science (GIScience), and cartography. Given the large body of research on map processing published over the past 40 years, we felt that it would be most useful to focus on the current state of the art instead of attempting to write a complete history of this field. As such, we have drawn on our combined 30 years of research experience on map processing to select and present the most important and promising techniques.

The remainder of this survey is organized as follows. Section 2 outlines a brief history of map processing research and defines the scope of this survey. Section 3 describes the relevant basic techniques in document analysis and pattern recognition that support map processing. Section 4 reviews past and ongoing research on map processing for extracting and recognizing geographic information from scanned maps. Section 5 concludes with a discussion and directions for future work.

2. A BRIEF HISTORY OF MAP PROCESSING AND THE SCOPE OF THIS SURVEY

Efforts of map processing or information extraction from maps have been going on for four decades [Freeman and Pieroni 1982] and show an increasing intensity as scanning technologies have improved, storage capabilities have increased, and processing speed has increased. Research on map processing has been conducted on different types of maps, including cadastral or land register maps (e.g., Boatto et al. [1992], Di Zenzo et al. [1996], Katona and Hudra [1999], Raveaux et al. [2008]), road maps (e.g., Bin and Cheong [1998], Itonaga et al. [2003], Dhar and Chanda [2006], Bucha et al. [2007], Chiang et al. [2009], Chiang and Knoblock [2013]), hydrographic maps (e.g., Trier et al. [1997]), city maps (e.g., Chen et al. [1999]), and utility maps (e.g., Den Hartog et al. [1996]), as well as topographic or other survey maps (e.g., Morse [1969], Yamada et al. [1993], Yamamoto et al. [1993], Khotanzad and Zink [1996], Frischknecht and Kanani [1998], Arrighi and Soille [1999], Ogier et al. [2001], Bessaid et al. [2003], Miyoshi et al. [2004], Chen et al. [2006], Leyk et al. [2006], Xin et al. [2006], Henderson et al. [2009]). Mostly, the described systems were unable to process different types of maps and thus the focus was rather narrow (e.g., extraction of geographic-feature layers from USGS topographic maps [Henderson et al. 2009]). Most studies focus on particular features, symbols, or map layers; therefore, efforts to extract map contents have been highly map specific, not applicable to a broader range of map products.

One important question becomes apparent: why is the literature on map processing so dispersed over various scientific disciplines? This question can be answered by looking at the early years of map processing when this research direction was considered part of image or document analysis. At that time, the main actors were computer scientists interested in technological aspects around digitization and information extraction in general; they developed specific hardware for manual map digitization [Leberl and Olson 1982], and they considered maps as just one kind of document [Cofer and Tou 1972; Ejiri et al. 1984] without exploiting many of the cartographic principles to process maps. As soon as the first GIS tools became more established and available as desktop solutions, the interest in digital map data and recognition tools increased considerably. GIS users, companies, and institutions quickly realized that maps in analog (paper) format represented one important source for generating digital spatial information that could be accessed, stored, and managed in GIS environments. Naturally, the issue of recognition in map documents gained more interest in the scientific community for two main reasons. First, the large number of paper maps stored in archives called for the need of higher degrees of automation because manual digitization was (and is) a highly labor-intensive, expensive process. Second, it was soon realized that such extracted spatial data would enable unprecedented GIS-based research efforts in a variety of disciplines such as landscape ecology (e.g., Kienast [1993]) or land-cover change analysis (e.g., Petit and Lambin [2002], Kozak et al. [2007]) as well as in studies on population dynamics and urbanization (e.g., Dietzel et al. [2005]), potentially covering large areas and long periods of time. Thus, the growing interest in map processing is driven by the increasing topical or substantive potential to use map data and also by the lack of general methods and techniques to overcome the technological challenges in map processing.

Today, map production in the Western world is mainly digital, and thus map processing focuses primarily on historical map documents (historical in a sense of predating the switch to digital map production), which were originally published on paper. In addition, map processing also focuses on map documents of which the original reproduction materials do not exist or are difficult to access. Therefore, the motivation of map processing is twofold. On one side, the scanning, digitization, registration, and referencing of maps represent an important way to preserve existing paper maps in a digital format. On the other side, map processing aims at the generation of spatial data that can be used in a GIS for spatial analyses and can be combined with different spatial data from various points in time. As such, map processing can also be seen as an advanced approach to derive and preserve unique historical information that provides a way to generate a better understanding of the complex landscape and its changes over long periods of time.

Maps are widely used in our daily life, and for centuries cartographers have been producing maps. In this survey, we focus on the map processing techniques that process map images underlying common cartographic principles and leave out other types of maps that would require different processing techniques for information extraction; two examples are sketch maps [Broelemann and Jiang 2011] and tourist maps, which commonly do not have an underlying map projection and are intentionally biased or skewed to improve information representation. Map processing is an active research area that spans many research fields, such as geography, pattern recognition, computer vision, image processing, and document analysis and recognition; therefore, the terminology used can be slightly different depending on the publication venue. The terminology used in this survey is briefly defined as follows:

—A *scanned* map is a digital image of a paper map generated using an image scanner. For example, the USGS digital raster graphics (DRG) is a series of scanned maps from the USGS standard topographic map series.
—An *electronic* map is an image of a map generated directly from computer systems, in most cases, GIS. Electronic maps represent a large portion of map images on the Internet (where maps can be found by keyword search using search engines [Goel et al. 2011]), such as snapshots of Google Maps or copies of the TIGER/Line maps. The source and georeferencing information of these map images is often missing or unknown, and digital map processing techniques can be used to georeference the maps [Chen et al. 2008; Chiang et al. 2009] or identify the map source [Chiang and Knoblock 2009].
—A *map image* is either a scanned map or an electronic map.
—*Digitizing* or *digital encoding* a map is the process of converting a map image into spatial datasets (i.e., points, lines, and polygons in vector format or extracted layers in image [raster] format). For example, the result from map digitizing can be any of the commonly available spatial vector data types (e.g., Esri shapefiles) where linked nodes and vertices can be used to represent features such as road lines.
—A *geographic feature* or *map feature* can be a set of linear objects, such as road lines or contour lines, a set of character objects, a set of area objects (area features), or a set of map symbols.
—A *geographic-feature layer* (feature layer or map layer) is a bilevel image that contains a number of pixels representing a set of geographic features in a map.
—*Extracting* a geographic-feature layer or a geographic feature is the process of extracting the set of pixels that represents an individual geographic feature in a map.
—*Recognizing* a geographic feature is the process of converting the geographic feature into a machine-editable format, such as a road vectorization process in which the recognized data are of known semantic meaning.
—An *attribute* is a characteristic that can be used to characterize features (e.g., shape descriptors) or colors (color attributes). The term *feature* is often used for *attribute* in the image processing and computer vision literatures. However, for the purpose of consistency in this survey, we use feature as described previously and strictly rely on the term *attribute* for characteristics or descriptors of such features and/or colors.

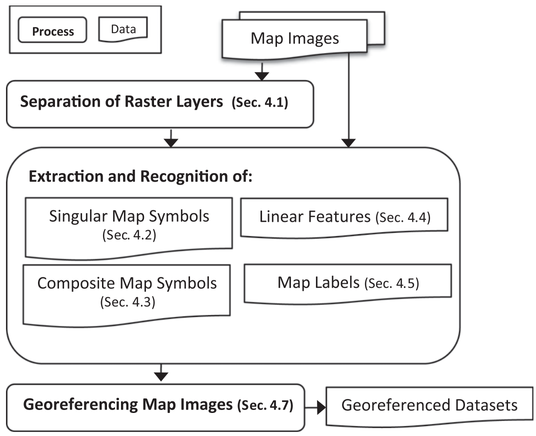


Fig. 3. An example map processing workflow and the corresponding sections in this survey.

3. BASIC TECHNIQUES FOR DOCUMENT ANALYSIS AND PATTERN RECOGNITION IN MAP PROCESSING TASKS

The overall goal of map processing is to generate either image (raster) or vector layers of geographic features from map images for subsequent use in a GIS. One common way to reach this goal is to first separate geographic-feature layers from map images by grouping image pixels of similar colors. An image (raster) layer can then be converted to vector data by recognizing individual features of a geographic type from each of the separate layers. Other approaches compute (geometric) attributes (e.g., shape descriptors) from training samples of desired geographic features and search for similar patterns in the map to directly recognize individual geographic features. Figure 3 shows a typical workflow for map processing where each process maps to a subsection in this survey. The entire workflow can be fully automatic, require some preprocessing/postprocessing, or be interactive (Section 4.6). In the remainder of this section, we briefly describe the background knowledge of printing and scanning techniques and then introduce the basic techniques of color image segmentation (CIS) and feature recognition that are relevant for map processing. These technical details will lay the foundation for then reviewing the different contributions on map processing in the subsequent sections of this survey.

Before any of these techniques can be applied to map documents, which are in general printed on paper, these documents have to be scanned. This scanning process is a crucial step to produce map images and determines the image quality of the scanned maps. Scanning parameters such as spatial sampling rate (or resolution), sampling spot size (or blur), bit depth of the resulting data, and brightness and contrast settings, as well as scanner calibration procedures, are determinants of the properties of the resultant image, and subsequent extraction techniques heavily depend on these properties. One of the most important parameters on a consumer-grade scanner is the scanning resolution. A rule of thumb is that for OCR, the scanning resolution needs to be at least 300 dots per inch (DPI). Figure 4 demonstrates two sets of maps scanned at three different scanning resolutions.[1]

Although a quantitative measurement of the scanning quality is possible (e.g., using signal-to-noise ratio to compare multiple scans of the same images), judging the quality

[1]Figures 4 and 5 were created by Dr. Reid Priedhorsky for his review on the print quality of topographic maps, which can be found at http://reidster.net/trips/maps/.

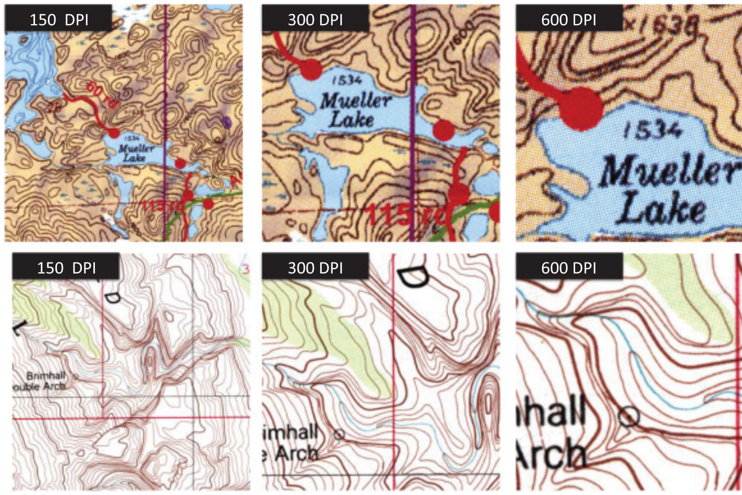


Fig. 4. Maps scanned at various resolutions but with the same image dimension where lower DPI values allow the image to cover a larger area (but fewer image details) than greater DPI values.

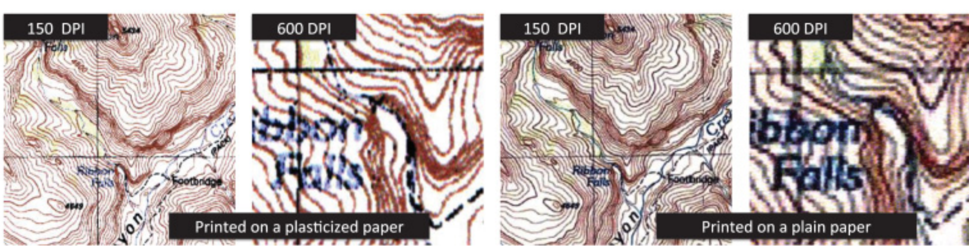


Fig. 5. Two scans of the same map extents (at 150 and 600 DPI, respectively) printed on two different materials.

of a scanned map for its suitability for digital map processing is subjective and can vary among different geographic features. Figure 5 shows the same map scanned from two different materials (paper types). For the purpose of contour line extraction, the scanning results on the left of Figure 5, which originate from plasticized paper, are favorable since the contour lines appear crisp in the final image. On the other hand, for the purpose of character recognition, the scanning results from plain paper (on the right of Figure 5) are more suitable due to the solid, uniform-colored characters. In this survey, we do not include specifics of the scanning process and instead focus on the map processing techniques (see Stoffel and Moreland [1981] and Sharma and Trussell [1997] for more details on image scanning and digital color imaging). Additionally, the printing method used for map production and materials of the original maps largely affects the results of the final scans. For example, maps were often printed using a halftoning process, which uses limited colors to create a uniform perception of numerous color tones. A typical result from the halftoning process is a series of dotted pixel patterns (texture) in the scanned image. The green and blue areas in Figure 4 show these dotted patterns. Noise removal filters, such as the median filter, are usually applied to the raw scanned image before the color segmentation step to suppress the halftone texture [Fung and Parker 1996].

A common image format to store the raster image (bitmap) produced from scanning is the tagged image file format (TIFF). The TIFF is a flexible format that supports both lossy and lossless compression methods, various image types (e.g., grayscale, RGB, or indexed color), and user-defined color spaces. An extension to the TIFF called *GeoTIFF* allows the TIFF image to store georeferencing information. In contrast to the fact that the TIFF stores only raster data, a format that stores both raster and vector data is the portable document format (PDF). In addition, the GeoPDF built on the PDF is a container that has been specifically designed to host geospatial data. It can store multiple layers of raster and vector data and their georeferencing information. For example, the digital USGS topographic maps are now distributed in GeoPDF format with a number of data layers, including orthoimagery, transportation, geographic names, and contour lines.

3.1. Color Image Segmentation

In map processing, similar to most other image analysis and pattern recognition tasks, image segmentation represents a crucial preprocessing step of which the outcome directly influences subsequent processing steps [Fu and Mui 1981; Pal and Pal 1993]. The basic intention of image segmentation is to divide an image into exclusive regions such that each region is homogeneous, but the union of any two neighboring regions is not [Cheng and Sun 2000].

Most map images contain color (i.e., are encoded in a color space such as RGB, HSI, or CIE L^*u^*v), not just grey values. In such cases, CIS can be applied in this preprocessing step [Cheng et al. 2001; Lucchese and Mitra 2001]. Most maps contain thematic layers represented in predefined colors (e.g., blue for hydrography, red for elevation, or green for vegetation), and different color spaces have been utilized in the past to perform CIS. The main goal is hence identifying color-homogeneous regions and reducing the number of color values found in the original map images to the number of existing map color layers. This allows for characterizing CIS in color maps as a typical classification problem based on color and spatial attributes [Cheng et al. 2001], which means that resulting homogeneous regions must show spatial contiguity and that connectivity has to be preserved.

Numerous existing approaches for CIS have been tested separately or in combination in map processing efforts, and the basic techniques are briefly explained next. More detailed overviews of these techniques can be found in Cheng et al. [2001] or Lucchese and Mitra [2001].

Histogram thresholding is an attribute-space–based technique that uses histograms of color values found in an image to identify values or value ranges that occur frequently and thus could represent possible color classes [Sahoo et al. 1988; Pal and Pal 1993; Cheng et al. 2001]. This approach searches for global and local maxima in frequency distributions of color values and assumes that peaks in a histogram correspond to color classes thought to be occurring spatially adjacent. Spatial contiguity is not explicitly accounted for in histogram thresholding. Histograms are usually created separately for each color space dimension, and results are combined in different ways. If the underlying image is a grayscale image, the histogram-based approach uses only one frequency distribution. Figure 6 shows an example of grayscale histogram thresholding. The original map (left) is first converted to a grayscale image (middle). Then the grayscale histogram (bottom) of the grayscale image is used to determine the grayscale range of the background pixels for separating the map features from the map background. The bilevel image (right) is the thresholding result when the pixels that have grayscale values between 199 and 255 (which is the map background corresponding to the blue portion toward the right in the histogram) are removed. Homogram-based approaches derive from histogram-based thresholding and attempt

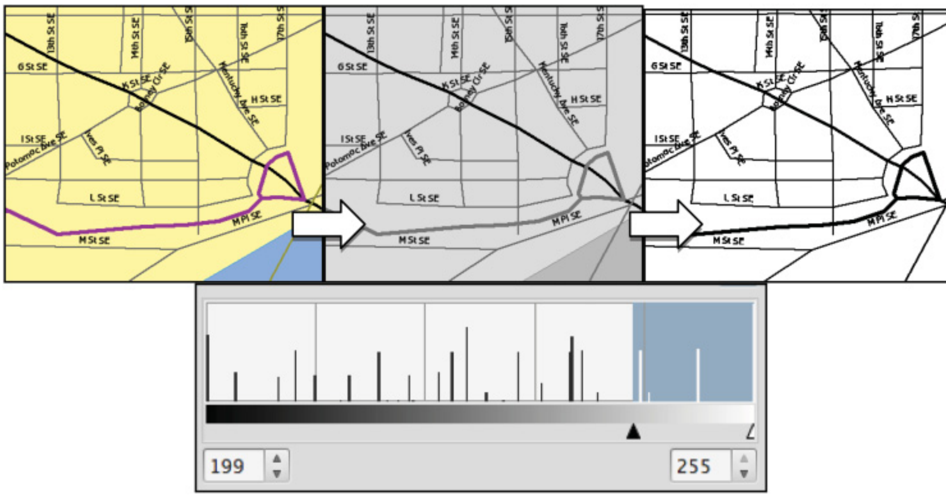
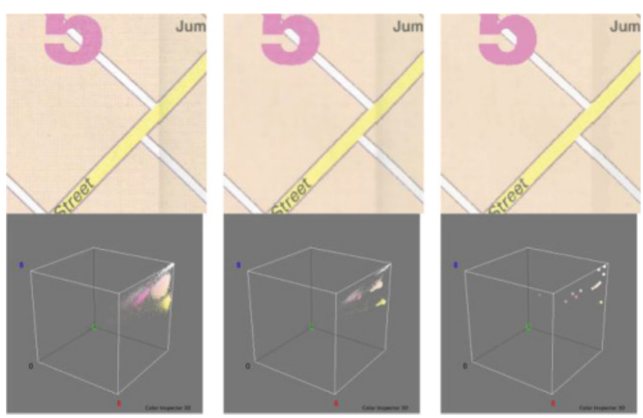


Fig. 6. An example of grayscale thresholding.



(a) An example tile (b) Mean-shift results (c) K-means results (K=16)

Fig. 7. The principle of color space clustering. Each pixel is assigned to one of the classes based on distances between its corresponding position and different cluster centers in color space. The resulting images (top) appear similar even though the numbers of colors in the color space (bottom) are significantly reduced after applying each clustering algorithm.

to overcome the lack in contiguity by implementing homogeneity criteria and applying histogram-based thresholding using locally homogeneous pixel locations [Cheng et al. 1998, 2002].

Color space clustering is an unsupervised classification approach to identify classes of objects without any prior knowledge [Bow 1992]; it can be seen as the multidimensional extension of thresholding methods [Fu and Mui 1981]. Clustering makes use of all dimensions of the selected color space and attempts to identify statistically significant clusters in this color space, assuming that high frequencies of similar color value combinations potentially describe main color classes (Figure 7). Existing algorithms that have been frequently applied and adjusted in different ways in the context of map processing are K-means (or C-means) [McQueen 1967], the fuzzy set–based alternative, fuzzy C-means (applied to CIS, as in Wu et al. [1994] and Zheng et al. [2003]), or Iterative Self-Organizing Data Analysis (ISODATA) [Ball and Hall 1965]. The basic

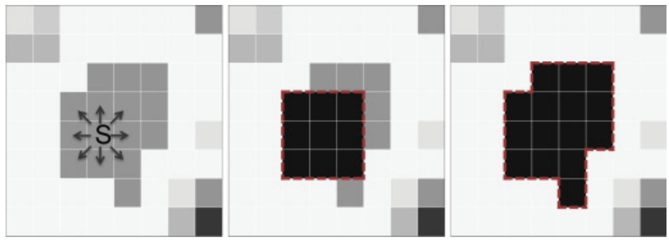


Fig. 8. The principle of region growing is to identify a seed pixel for a region of interest and to search the local environment for similar pixel values and, if homogeneity criteria are fulfilled within this local environment, the region is grown. This step is then repeated until no further pixels can be added.

idea of these algorithms is that n observations are partitioned in k clusters in that each observation belongs to exactly one cluster (K-means) or to different clusters at varying degrees (fuzzy C-means). The decision as to which class each observation is assigned depends on distance measures (e.g., the Euclidean distance) in a color space between a cluster center and the observation. Similar to the thresholding approach, spatial contiguity of color values is not explicitly accounted for without consideration of homogeneity in a local plane. In addition, the number of clusters in the underlying image has to be specified, potentially introducing subjectivity into the process. To overcome this limitation, nonparametric clustering methods, such as the Mean-Shift algorithm [Comaniciu and Meer 2002], can be used in color segmentation without prior knowledge of the number of resulting clusters (i.e., the number of colors in the quantized image). Figure 7(b) shows the CIS results of the Mean-Shift algorithm, which merges similar colors if they are spatially close in the image plane [Chiang and Knoblock 2013].

Region-based approaches are based on the principle of grouping pixels into homogeneous groups applying techniques such as region growing, region splitting, or region merging. These techniques are often called and classified as image-domain–based techniques [Lucchese and Mitra 2001] because they take spatial contiguity (or compactness) of homogeneous regions as well as connectivity between these regions directly into account. Region growing approaches typically start with the definition and placement of so-called seeds—pixels that have to fulfill predefined criteria of color types and local homogeneity [Rosenfeld and Kak 1982]. These seeds are the starting positions for testing their surrounding pixels for similarity to the seed color values and for homogeneity. If the surrounding pixels fulfill the predefined conditions, they are added to the region. In this way, regions (i.e., groups of similar color occurrences) around these seeds are built (grown) sequentially until the image is partitioned into homogeneous groups (Figure 8), followed by a merging process for small regions to maximize the level of overall homogeneity.

Split-and-merge techniques start with the partitioning of the image into regions and then examining these regions for homogeneity. If the regions are found to be nonhomogeneous, they are further partitioned (split) until homogeneous subregions are found—similar to quadtree data representations (Figure 9) [Bow 1992; Jain et al. 1995].

After this splitting phase, small regions are merged under the constraint of maintaining homogeneity criteria. Region-based approaches are more robust against noise especially where contiguity and connectivity criteria are important for image processing, but they are computationally more expensive. In addition, region-growing results depend on the location of seeds or seed regions as well as on the order and direction of pixel examination. To overcome some of these limitations, region-based approaches are

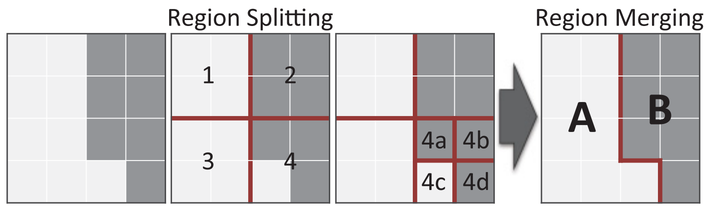


Fig. 9. The principle of merge-and-split approaches. Region splitting searches for homogeneous regions in an image by hierarchically partitioning the area into quadrants (as in the quadtree approach). This partitioning process continues until there are no heterogeneous regions left. Region merging merges adjacent subregions if the homogeneity criterion can still be warranted.

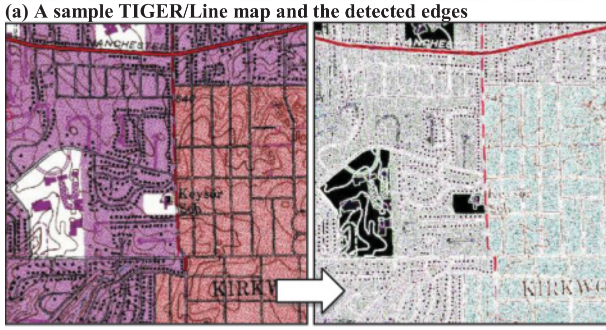
often combined with color space–based approaches such as clustering or histogram-based techniques in a preliminary step to identify a number of seed candidates.

Edge detection techniques work on the principle of detecting contrasts in color values between regions and thus identifying discontinuities—that is, abrupt changes in color values using local filtering operators [Fu and Mui 1981; Pal and Pal 1993]. Complete segmentation results can only be achieved in concert with other techniques such as region-based approaches. Traditionally, edge detection has been performed in grayscale images and can be classified in parallel and sequential techniques [Pal and Pal 1993]. Parallel means that the decision of one location to be part of an edge is independent of the decisions made for other pixels; thus, differential operators to detect edges can be applied simultaneously anywhere in the image. In sequential techniques for edge detection, the decision whether or not a pixel is part of an edge depends on pixels examined previously; therefore, the process highly depends on the starting point. Edge detection in color images is based on the same approaches described earlier for monochromatic (or grayscale) images but makes use of the richer color information. Principally, discontinuities in color space including all color channels are detected by calculating gradients in single color channels as in grayscale images, and the results are then combined in different ways. In order to increase robustness of the process, some uniformity constraints of detected edges are added. Edge detection techniques show limitations in images with poor contrasts or disconnected features as well as when noise is present due to the small spatial extent of the local differential operators. Figure 10 shows edge detection results from a TIGER/Line map and a USGS topographic map. The results from the topographic map are quite noisy due to the poor graphical quality of the scanned image.

Finally, *artificial neural networks* have a long history of being implemented to solve classification and clustering problems in pattern recognition tasks including CIS in map images. In short, neural networks consist of numerous computational elements called *processors*, which are connected by links with variable weights [Bow 1992]. Such a neural network allows for representing highly complex sets of interdependencies that can be of particular advantage in CIS if the image suffers from blurring or mixed pixels, making class assignment a more complex task. One example is the Hopfield network [Bow 1992], whose network architecture is derived from the energy function and applied to CIS (e.g., Campadelli et al. [1997]). Here, image segmentation is considered as the problem of minimizing a suitable energy function. Several other neural network–based CIS approaches exist, such as self-organizing maps, back propagation, or learning vector quantization, but have not been applied to map processing tasks. Typical problems of neural network–based approaches to image segmentation are the extended training time and high specificity to the image of interest (lack of generality).

CIS, not only in map processing tasks, is frequently carried out by making use of different combinations of earlier described segmentation techniques [Den Hartog et al.

(a) A sample TIGER/Line map and the detected edges



(b) A sample USGS topographic map (the purple areas are the updated features since the original survey, and the white areas represent the areas with scant or no vegetation) and the detected edges

Fig. 10. Edge-based image segmentation attempts to calculate gradient functions in order to find discontinuities that define edges between regions in the image (black pixels in the images to the right represent zero gradient as a result of the local gradient function).

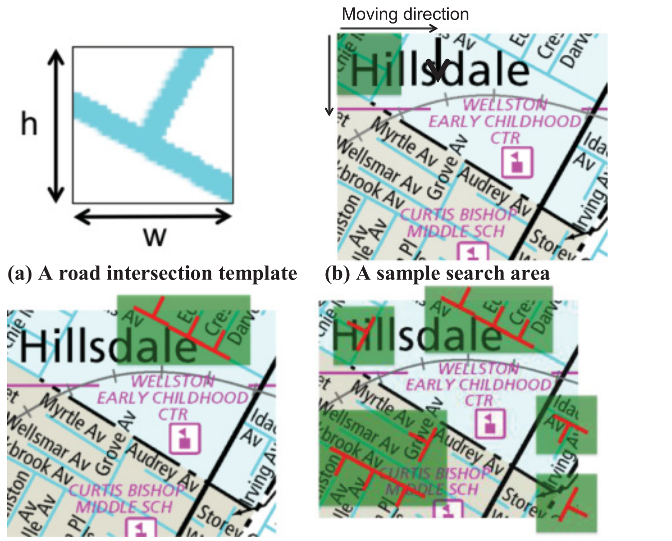
1996; Centeno 1998; Santos et al. 1998; Cordeiro and Pina 2006; Leyk and Boesch 2010; Leyk 2010]. However, if maps suffer from low graphical quality and excessive complexity, CIS represents a difficult process in which preservation of connectivity and shapes of map objects is particularly crucial. Relevant approaches that aim at overcoming such problems in map images will be reviewed in more detail in Section 4.1.

3.2. Geographic Feature Extraction and Recognition

The process of extracting and recognizing geographic features from scanned maps utilizes and combines a variety of techniques that stem from the fields of document analysis and image processing and fall into three major categories: template matching, morphological operators, and shape descriptors.

Template matching techniques are widely used in computer vision, pattern recognition, and document analysis research. A detailed survey of various template matching algorithms and their applications can be found in Brunelli [2009]. In general, the main goal of these template matching algorithms is to find objects in the document of interest that are similar in shape and size to an object given as a template. The most commonly used algorithm in map processing is correlation-based template matching [Rosenfeld and VanderBrug 1977], which evaluates the correlation between the pixel values of a template and the pixel values of the target image within a window of the same size as the template to find the objects of interest. Here we briefly introduce the principles of correlation-based template matching exemplified on a road intersection detector recently used in map processing work [Chen et al. 2008; Chiang et al. 2009].

Given a road intersection template, T, of the size w by h pixels as shown in Figure 11(a), the road intersection detector searches the input map shown in Figure 11(b) within a moving (or sliding) window, S, of the size w by h pixels. The



(a) **A road intersection template** (b) **A sample search area**

(c) **Template matching results** (d) **Rotation-invariant matching results**

Fig. 11. Using template matching to identify a specific type of road intersections.

matching algorithm generates a similarity value correlating the contents of the map document within the moving window with the contents of the template. For example, a normalized cross correlation can be computed [Chen et al. 2008] as:

$$Normalized\,Cross\,Correlation(T_{x,y}, S_{X,Y}) = \frac{\sum_{y=1}^{h}\sum_{x=1}^{w} T(x, y)S(X+x, Y+y)}{\sqrt{\sum_{y=1}^{h}\sum_{x=1}^{w} T(x, y)^2 \sum_{y=1}^{h}\sum_{x=1}^{w} s(X+x, Y+y)^2}},$$

where $T(x, y)$ equals 1, if the pixel at position (x, y) in the template is a foreground pixel (i.e., is part of the road intersection template); otherwise, $T(x, y)$ equals zero. $S(X+x, Y+y)$ equals 1 if the map pixel at (x, y) in the moving window matches the color of the pixel at (x, y) in the template; otherwise, $S(X+x, Y+y)$ equals zero. The location (X, Y) is the current window shift position. Each matching interaction shifts the window position by one pixel in either the horizontal or vertical direction (overlapping the last window). The normalized cross correlation measure can range from 0 to 1, where 0 means that the map contents within the moving window are completely different from the template, and 1 means a perfect match. By adjusting a threshold on the correlation measure, we can identify map objects that are geometrically similar to the template (i.e., a high threshold is used to identify map objects that match most of the template). Figure 11(c) shows an example of the road intersections (red intersections in the green areas) that can be identified using the described template matching algorithm based on a correlation threshold.

There are more sophisticated template matching algorithms that detect objects with a similar shape but different sizes (scale invariant) or in different orientations (rotation invariant). Rotation-invariant template matching algorithms often use shape descriptors (explained at the end of this section) instead of using every pixel in a moving window to compute the similarity values and are more computationally intensive. Figure 11(d) shows the detected road intersections (red intersections in the green areas) when a rotation-invariant template matching algorithm is used.

Morphological operators are commonly used to modify image contents to facilitate and improve the extraction and recognition of point, linear, and area features. In more complex settings, morphological operators have been applied for identifying parallel linear features [Yamada et al. 1993; Chiang et al. 2009], reconnecting broken linear features [Chiang and Knoblock 2013], separating linear features from text [Luo and Kasturi 1998], removing noise objects for recovering large area features [Angulo and Serra 2003], and converting areas of pixels to vector format [Doermann 1998; Tombre et al. 2000]. Basic morphological operators are binary operators that deal with images with two colors (i.e., bilevel images) and can be extended to process grayscale and color images. These traditional morphological operators are based on the hit-or-miss transform using masks of various sizes and shapes (see Pratt [2001] for a detailed description of common morphological operators). A hit-or-miss transform scans every pixel and its local environment (e.g., a square mask with 3×3 pixels) in an input bilevel image. A "hit" condition arises if the mask matches a pixel and the pixel's neighborhood pattern—that is, the local morphological conditions of interest; otherwise, it is a "miss" condition. Each morphological operator employs a different mask to perform the hit-or-miss transform and triggers specific actions in response to a hit or miss condition. This hit-or-miss transform is a type of template matching algorithm that generates binary results (either a hit or a miss) instead of similarity measures (in the range [0, 1]) as described earlier in the road intersection detection example.

The most commonly used morphological operators also in map processing tasks are binary operators (in contrast to the multilevel morphological operators that apply to grayscale or color images) because the results of CIS can be conceptualized as a set of bilevel images representing individual map layers [Agam and Dinstein 1996]. Binary morphological operators that are often employed in combination include the dilation, erosion, closing, opening, and thinning (skeletonizing) operators. The aim of a dilation operation is to expand the region of foreground pixels if the hit condition is fulfilled. In map processing, a dilation operator is commonly applied to fill gaps of broken lines or holes in presumably homogenous areas. For example, if a background pixel is found with one or more foreground pixels within its 3×3 pixel neighborhood (i.e., a hit), a binary dilation operator with a 3×3 square mask converts the background pixel to a foreground pixel (i.e., the action in response to a hit), thus expanding the area of foreground pixels. Figure 12 shows an example of applying a binary dilation operator iteratively to merge road edges.

The idea of a binary erosion operator is to reduce the region of foreground pixels. For a foreground pixel with one or more background pixels within the 3×3 pixel neighborhood (i.e., a hit), a binary erosion operator with a 3×3 square mask converts the foreground pixel to a background pixel (i.e., the action in response to a hit), thus reducing the area of foreground pixels. Figure 12 also shows an example of applying an erosion operator iteratively to reduce road areas.

The closing operator is a dilation operator followed by an erosion operator, and the opening operator is an erosion operator followed by a dilation operator. The closing operator is used to remove holes and smooth region boundaries without expanding region areas. For example, if a closing operator with a 3×3 square mask is applied, the background areas that are smaller than 3×3 will be first conveyed to the foreground by the binary dilation operator and the foreground area will grow in all directions. Next, the erosion operator will shrink the foreground area in all directions to offset the expansion of the foreground area from the dilation operator. In contrast, if an opening operator with a 3×3 square mask is applied, the foreground objects that are smaller than 3×3 will be first converted to background pixels and the foreground area will shrink in all directions. Next, the binary dilation operator will expand the foreground area in all directions to offset the shrinking of the foreground area from the binary erosion operator.

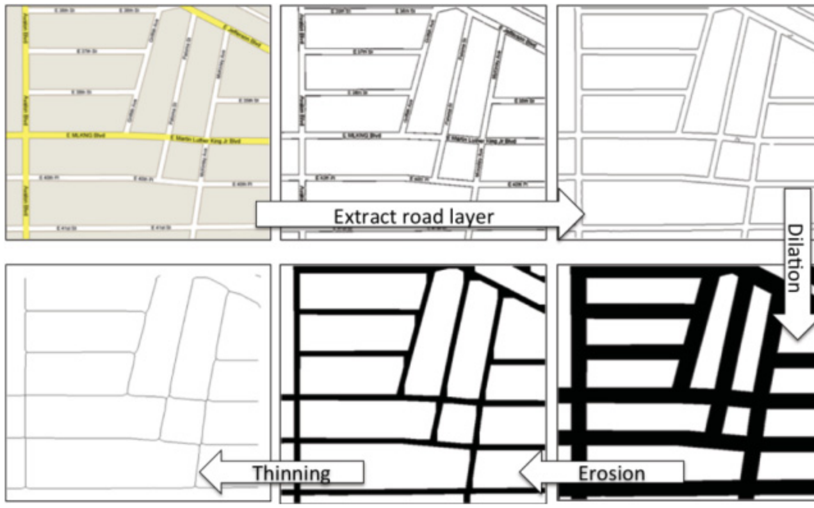


Fig. 12. Using binary morphological operators to extract road centerlines [Chiang et al. 2009].

Due to its simplicity, the thinning operator is widely used in raster-to-vector conversion to produce a so-called skeleton of a group of foreground pixels (see Hori and Okazaki [1992], Röösli and Monagan [1995], and Janssen and Vossepoel [1997] for more sophisticated, high-quality vectorization techniques). The thinning operator is a conditional erosion operator with an additional verification step. The first step of the thinning operator is to label every foreground pixel that connects to one or more background pixels (i.e., the same idea as the binary erosion operator) as a candidate pixel that may be converted to a background pixel. Next, in the verification step, each candidate pixel is examined to determine if the conversion of that pixel causes any disruption of original line branches or crossings. This verification ensures that the original structure of the considered object is not compromised. One common critique of the thinning operator is that the resulting skeletons do not accurately represent the original shape of the object in terms of geometry and topology [Tombre et al. 2000]. Hence, the results of the thinning operator often require manual editing or subsequent shape adjustment to correct the thinning results. Figure 12 presents an example of applying a generic thinning operator to identify the road centerlines.

Shape descriptors have commonly been used to identify text and symbolic features in map processing. Shape descriptors range from simple metrics of height, width, area, compactness, or the bounding box of a connected component to computationally more complex ones, such as descriptors based on the discrete Fourier transform for estimating texture representations [Adam et al. 2000; Chiang and Knoblock 2006] or the Hough transform for identifying imperfect linear objects (e.g., broken lines) [Kasturi and Alemany 1988; Chen and Wang 1997; Dhar and Chanda 2006; Chiang and Knoblock 2013].

Simple shape descriptors (e.g., geometric measures of height, width, or area) usually rely on a connected component analysis. First, connected components (i.e., regions or groups of pixels that are defined by certain connectivity and adjacency criteria) are identified and labeled in the input image. Adjacency (and thus connectivity) can be conditional on the existence of pixels next to the pixel of interest either only in horizontal or vertical directions (von Neumann or "diamond-shaped" neighborhood) or in both horizontal/vertical and diagonal directions (Moore or "square-shaped" neighborhood). Every pair of foreground pixels that fulfills the underlying adjacency condition belongs

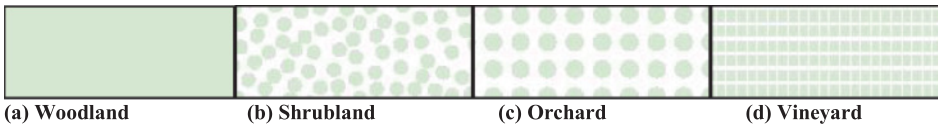


Fig. 13. Textures used to represent different vegetation/land use types in USGS topographic maps. Credit: U.S. Geological Survey (USGS topographic map symbols).

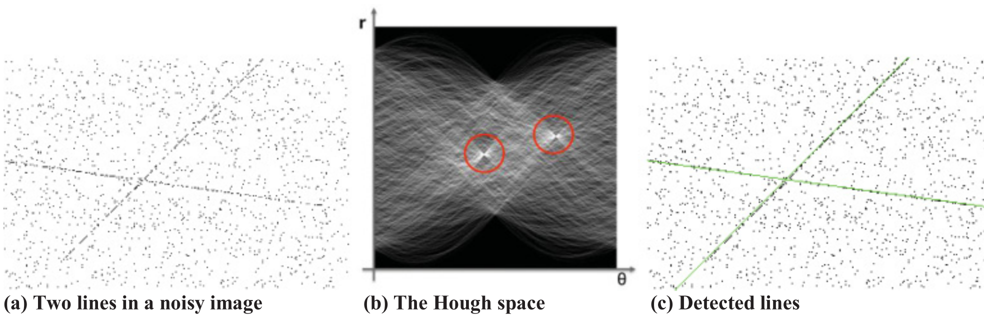


Fig. 14. The Hough transform.

to the same connected component. Next, sets of attributes—shape descriptors—of the individual connected components are calculated; these are subsequently used to identify specific features such as text objects or various symbols (see Salmon et al. [2007] for a detailed description of common shape descriptors). For example, given a scanned map that contains text labels and road lines, various descriptors (or attributes) such as width, height, and compactness of the text characters can be empirically determined in the map and subsequently used to filter out road lines and retain character-like objects (or connected components) for recognition [Li et al. 2000; Cao and Tan 2002].

Shape descriptors based on the discrete Fourier transform [Brigham 1988] have played an important role in multiple texture classification applications due to their ability to generate distinct attributes for different texture representations [Randen and Husoy 1999]. The discrete Fourier transform maps an image into the frequency domain where the strength of each frequency is represented by one of the estimated Fourier coefficients. Since the texture representations of the same map features are usually similar within a local environment (but vary across different map features), the Fourier transform (or the Fourier-related transform, such as the discrete cosine transform) can be used to distinguish between map features [Chiang and Knoblock 2006]. For example, Figure 13 shows four green patterns that are used to depict different vegetation/land use types in USGS topographic maps.

Another technique that is extensively used in map processing is the Hough transform. The Hough transform is a technique frequently used to identify imperfect lines (e.g., broken lines or dashed lines) [Duda and Hart 1972]. For example, Figure 14(a) shows two lines in a noisy image (white pixels are background); parts of the line pixels are missing. To detect the two lines in Figure 14(a), the Hough transform first transforms the image to the Hough space (Figure 14(b)) (black pixels are the background). In the Hough space, a sinusoidal curve represents a set of straight lines that pass through a point in the image plane. A point in the Hough space represents a straight line in the image plane, which is described by two parameters: r and θ. As shown in Figure 15, in the image plane, the parameter r is the distance between the straight line and the point of origin, and θ is the angle of the shortest line between the point of origin and the straight line. Using the two parameters, r and θ, a straight line of certain orientation

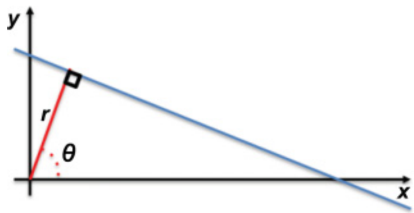


Fig. 15. Using r and θ to represent a straight line.

is described as

$$r = x\,cos\ \theta + y\,sin\ \theta.$$

For the set of straight lines that pass through a point, $P\ (x_p, y_p)$, in the image plane, the sinusoidal curve in the Hough space is

$$r(\theta) = x_p\,cos\ \theta + y_p\,sin\ \theta.$$

When two sinusoidal curves generated from two points, $P(x_p, y_p)$ and $Q(x_q, y_q)$, intersect in the Hough space, the intersection point represents a straight line in the image plane that passes through two points, $P(x_p, y_p)$ and $Q(x_q, y_q)$. Hence, by finding the local maxima in the Hough space where many of the sinusoidal curves intersect, we can detect possible straight lines in the image plane even if the lines are constituted from imperfect objects. Figure 14(c) shows the detected straight lines in the image plane. Since identifying the local maxima in the Hough space requires only a simple grayscale thresholding in the Hough space, the Hough transform is a robust and straightforward choice in map processing for detecting linear features such as road lines [Yamada et al. 1993; Dhar and Chanda 2006] or for identifying straight map labels [Chen and Wang 1997; Chiang and Knoblock 2013].

4. A REVIEW OF MAP PROCESSING EFFORTS FOR THE EXTRACTION AND RECOGNITION OF CARTOGRAPHIC INFORMATION

Earlier approaches to map processing pursued different objectives. Some systems were designed as general recognition systems to extract multiple features from a map document [Wigand 1988; Ebi et al. 1994; Frischknecht et al. 1998; Bessaid et al. 2003; Dhar and Chanda 2006], whereas others are highly specialized approaches recognizing one well-defined set of features in the map [Fayek and Wong 1996; Watanabe and Zhang 1997; Miyoshi et al. 2004; San et al. 2004; Xin et al. 2006]. Typically, various preprocessing steps have to be conducted, such as image cleaning and filtering, followed by separating the feature of interest from other overlapping map symbols (e.g., Gamba and Mecocci [1999], Cao and Tan [2002], Bajcsy [2003], Leyk and Boesch [2009]), which increases the complexity of the recognition process. This complexity can be seen as one reason for the lack of automation in many recognition tasks.

We describe the recent approaches to map processing organized by the geometric properties of the features of interest (i.e., line or area features). Here, first we describe how the various thematic or map layers (geographic features) are represented (Figure 16):

—*Road features* are commonly represented as line features—single (road centerlines) or parallel (road edges)—or as area features (e.g., Maderlechner and Mayer [1995], Bin and Cheong [1998], Itonaga et al. [2003], Chen et al. [2004], Dhar and Chanda [2006], Bucha et al. [2007], Chiang et al. [2012], Chiang and Knoblock [2013]). Road intersections are often extracted and represented as point features because they

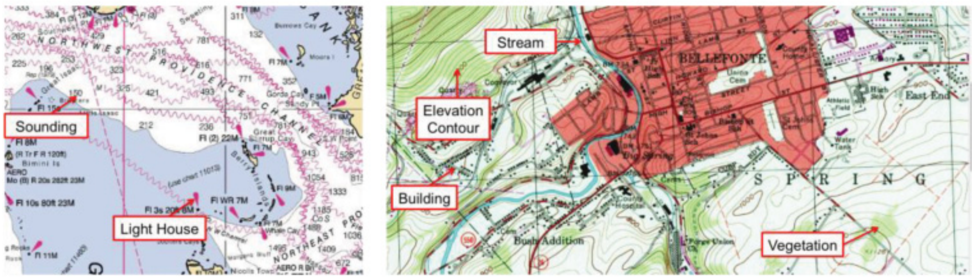


Fig. 16. Various thematic layers found in maps for possible extraction. Left: A nautical chart (National Oceanic and Atmospheric Administration [NOAA]). Right: A topographic map (USGS).

provide distinct characteristics in road networks (e.g., Habib et al. [1999], Liu [2002], Chiang et al. [2009], Henderson and Linton [2009]).

—*Vegetation features* often are represented by simple background filling areas but can also appear in the form of vegetation symbols [Ebi et al. 1994; Frischknecht et al. 1998] or symbol compositions with or without linear features as boundaries [Dhar and Chanda 2006; Leyk et al. 2006; Leyk and Boesch 2009].

—*Text* or *text labels* are mostly part of the black layer and appear in different styles, font sizes, and orientations and can even appear curved; recognition approaches have to deal with these difficulties (e.g., Yamada et al. [1993], Deseilligny et al. [1995], Chen and Wang [1997], Chen et al. [1999], Gllavata et al. [2004], Dhar and Chanda [2006], Pezeshk and Tutwiler [2008, 2010a, 2010b, 2011]). In addition, text objects are often overlapping with other features in the map (e.g., Kasturi and Alemany [1988], Luo and Kasturi [1998], Yin and Huang [2001], Cao and Tan [2002], Chiang and Knoblock [2011]).

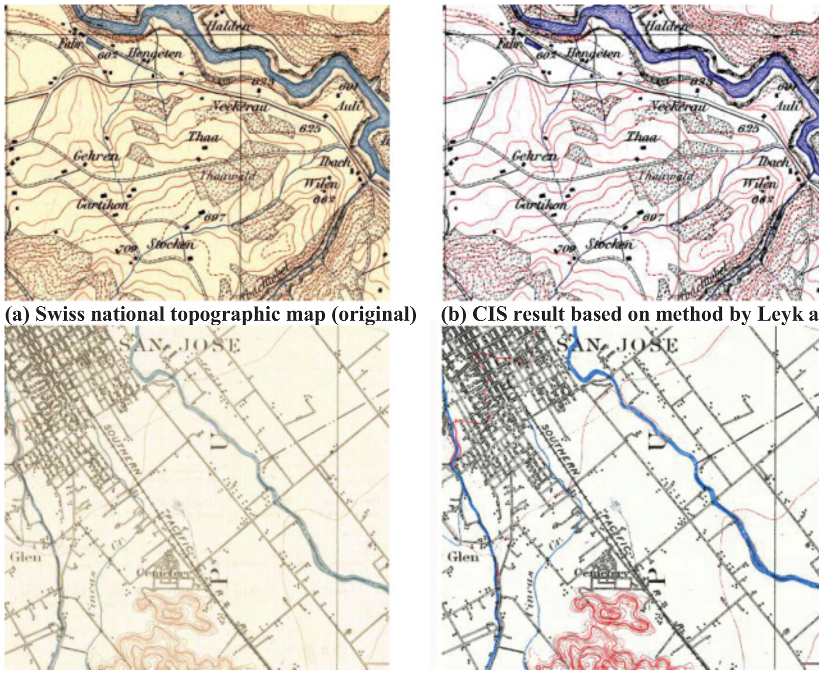
—*Hydrographical map features* (in most cases the blue layer) include streams that are commonly represented as linear or area features depending on the map scale [Ebi et al. 1994; Angulo and Serra 2003; Dhar and Chanda 2006; Henderson et al. 2009] or wetland areas often shown as a spatial distribution of single or clustered symbols [Leyk and Boesch 2010].

—*Building symbols* are in most cases small rectangular area features or can be elongated shapes (in urban areas with higher building densities); typically, they are part of the black map layer (e.g., Fayek and Wong [1996], Frischknecht and Kanani [1998], Liu [2002], Angulo and Serra [2003]). Building symbols often overlap with other map features, such as roads or text.

—*Elevation contours* are linear features that belong in most cases to the red or brown map layer and follow certain geometric properties—that is, they must not intersect each other (e.g., Yamamoto et al. [1993], Khotanzad and Zink [1996, 2003], Dupont et al. [1998], Salvatore and Guitton [2004], San et al. [2004], Chen et al. [2006], Pezeshk and Tutwiler [2008], Samet and Hancer [2012], Miao et al. [2013]).

—*Other cartographic symbols* that are of possible interest in map processing include features from nautical charts [Trier et al. 1997], symbols for landmarks, political boundaries [Gamba and Mecocci 1999] or churches [Frischknecht and Kanani 1998], and other legend-based symbols [Samet and Soffer 1998].

The following subsections describe and review map processing efforts in more detail focusing on the most relevant studies found in the literature, sorted by steps typically done in map processing: separating raster layers (segmentation), extracting area features, extracting composite features and other map symbols, line detection, text recognition, interactive map processing, and map registration. In numerous studies,



(a) Swiss national topographic map (original) **(b) CIS result based on method by Leyk and Boesch [2010]**

(c) USGS historical topographic map **(d) CIS result based on method by Leyk [2010]**

Fig. 17. Two exemplary CIS results from historical topographic maps. The results in (b) and (d) are shown in pure colors in quantized images, blue for hydrography, red for contours, black for the black layer, and white for background. Reproduced by permission of swisstopo (BA13123).

the reader can find accuracy measures for proposed map processing techniques such as percent correctly classified, recall, precision, or the kappa coefficient of agreement. However, objective measures are difficult to compute due to the lack of sufficient ("complete") reference data and inherent subjectivity—that is, it is always difficult to claim that the extraction of a cartographic symbol is perfect or acceptable.

4.1. Separation of Raster Layers (Segmentation)

Section 3.1 described the most commonly used general techniques for CIS applied to map processing tasks, which has the main goal of separating the different color layers found in a map in order to improve the performance of subsequent processing steps. This section reviews specialized image segmentation approaches developed for more complex and graphically inferior map images (Figure 17), which pose particular challenges. Numerous map processing studies are designed specifically for noise removal [Mello et al. 2012] conducted on well-conditioned maps, and segmentation often represents a single preprocessing step based on histogram thresholding or color space clustering. These approaches are excluded here. In this section, existing CIS approaches are grouped into three different categories according to the information domain that they use (color space and image plane): (1) approaches that use *only color space*, (2) approaches that use *color space and the local context in the image plane implicitly*, and (3) approaches that use *color space and local spatial relations in the image plane explicitly*.

Some recent CIS techniques in maps are *solely based on color space*. For example, in a study on Indian survey maps, image enhancement is applied in a first preprocessing step to change each pixel value to one of the possible RGB color space extremes such as {255,0,0} or {0,0,255} representing initial clusters in color space [Dhar and Chanda 2006]. Next, the K-means algorithm is applied to allocate each original pixel in the map to one of the initial clusters based on minimum distance in color space. The RGB values of the cluster centers are then calculated based on normalized feature vectors of all pixels that have been allocated to the corresponding initial color cluster. In a final separation step, each pixel is allocated to the nearest final (adjusted) prototype, and for each feature layer one bilevel image is generated. The approach contributes to high recognition rates in subsequent map recognition stages but assumes high levels of homogeneity within feature layers and well-defined color values.

Another CIS approach that works entirely in color space aims at the separation of color objects in topographic maps. Based on tonal values, Cordeiro and Pina [2006] define different cut levels for the identification of color objects in RGB color space. For each color layer, sampled color values define cut-level values for sections within the color space cube. These sections are then either intersected in color space to define noise *regions* (background) or are unified to create so-called *co-regions* in color space that define the color objects of the different colored map layers. The process of isolating the color objects is performed layer wise after already isolated layers were eliminated from the scanned image. This approach seems to work on maps with clearly defined color values, but it needs to be tested on maps with higher variability in color values and on maps with false colors. In addition, defining the cut levels introduces some subjectivity.

Some approaches to CIS in maps make *implicit use of the local spatial context in the image plane in addition to information from the color space*. One example refers to a segmentation process in utility maps. The process starts with a sharpening step to unblur the image using Laplacian linear filters [Den Hartog et al. 1996]. Next, a global grey value thresholding step with *hysteresis* is done. This means that they first define a grey value range and then identify pixels with values above and below this range as definite background and object pixels, respectively. In a second step, the pixels between the two thresholds are classified as object pixels only if they are directly connected to an already classified object pixel (adjacency in the image plane). This global segmentation is input to a complex recognition process. If inconsistencies can be detected in the recognition result, a resegmentation (or top-down segmentation) is performed to correct the errors made in the global segmentation. Although integrating prior knowledge improves the segmentation and subsequent recognition, Den Hartog et al. [1996] describe a highly laborious and complex technique that requires intensive user intervention and is specific to grayscale utility maps.

A second example is a study that uses hue histograms (from the HSV color space) to define thresholds of feature layers in different subregions of a scanned topographic map image [San et al. 2004]. This approach is able to adjust for varying cut-off values to identify map layers by evaluating bimodal subregional histogram distributions. In order to remove isolated small regions (noise), they perform an edge-preserving smoothing technique. Their approach incorporates an innovative step in partitioning the image to adjust for color variability, but it has not been tested on different maps.

A third study that belongs to this category is based on Eigenvector Line-Fitting in RGB color space to segment USGS topographic maps. In this work, Khotanzad and Zink [1996] propose a segmentation technique that overcomes problems of scanner-related false coloring and aliasing. They use a local window (7×7 pixels in the image plane) to classify transitional pixels between objects and background. Their approach is later applied and adjusted by Chen et al. [2006], who developed a technique called *local window segmentation*. This method applies a line-tracing algorithm to repair gaps and thick (i.e., merged) contour lines locally to improve the overall segmentation result.

This approach showed acceptable results on well-conditioned (or more recent) USGS topographic maps but has never been tested on other maps.

The third category of CIS approaches includes techniques that use *color space information for initial color prototype definition and make explicit use of spatial relations* to perform the image segmentation. For example, in a histogram-based approach on German topographic maps, color cluster centers are detected in the histogram of the *u'v'* chromaticity plane ($L^*u^*v^*$ color space). They apply a peak detection technique adjusted to this two-dimensional cumulative frequency distribution [Ebi et al. 1994]. These cluster centers are then used to perform the segmentation of map layers using chromaticity and lightness criteria. Finally, they employ region growing based on containment tests and structural texture analysis to fill unclassified areas in the map. Although this approach has been demonstrated for maps with well-defined color layers, the use of spatially explicit relations in the region growing step provides greater flexibility to process maps with lower graphical quality.

Other work that can be grouped in this third category combines color space, frequency domain, and the image plane to improve CIS in low-quality Swiss historical topographic maps. In this approach, only locally homogeneous pixels are included in the frequency distribution of the map image [Leyk and Boesch 2010]. The most frequent values (peak) and similar values (similarity is defined by distance in color space) are assigned to the first color layer and removed from the frequency distribution. The same process is performed to identify the remaining (initially hidden) color prototypes. These prototypes are then input to a seeded region growing process to identify connected regions based on color similarity and spatial adjacency in the local image plane. The method (Figure 17(a) and (b)) produces acceptable results but requires user-specified parameters; it also suffers from excessive processing time. This approach provided the basis for a two-stage CIS approach in historical USGS topographic maps [Leyk 2010] to overcome color variability found in features of the same map layer. In the first step, a sample of pixel values is collected from the image to determine global color prototypes, initially based on the distance in color space to the color extremes representing the map layers (e.g., {255,0,0} in RGB for red). The prototype positions in color space are adjusted sequentially with each pixel value added until stabilized. The prototypes are then used to define homogeneous core regions of map layers. In the second step, the color prototypes are recomputed based on a second sample drawn from the edges of homogeneous regions to account for color variability in map layers at transitions to background, thus improving the subsequent spatial expansion of regions (Figure 17(c) and (d)). This approach shows good results and shorter processing time on low graphical quality maps but still needs to be tested on other maps.

Summarizing the efforts mentioned previously, current challenges that can be identified for CIS in scanned maps are preserving the spatial context of segmentation results, ensuring the connectivity of homogeneous map elements, and maintaining the shapes of map objects while accounting for color variability of map contents within one map page and across maps. This variability is caused by finite-size sampling spots, which results in blurring of color boundaries and mixed-color pixels. Thus, even high-quality paper maps with uniformly colored features will yield scanned images with considerable variation in color. All of these aspects are crucial for generating topologically correct cartographic information for input into a GIS. Furthermore, more generic CIS approaches that would make it possible to process different maps or types of maps are in urgent demand. Such tests are rarely (if at all) done. In general, it can be stated that the complexity of CIS efforts increases with decreasing quality of the map image, which often relates to the age of the scanned map document. Accordingly, the degree of automation is lower if the methodological steps become more complex and the solution is often highly map or map series specific. However, the most promising strategy seems

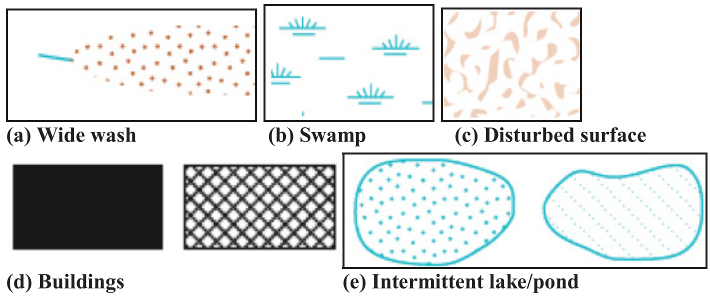


Fig. 18. Some examples of singular (bottom) and composite (top) features typically found in maps. Credit: U.S. Geological Survey (USGS topographic map symbols).

to be the combination of color space and image plane properties to produce robust segmentation results. As stated initially, segmentation techniques are sometimes difficult to differentiate from extraction and classification steps, which impedes tracking the most recent developments.

4.2. Extraction and Recognition of Singular Map Features

Symbols can be described as sets of distributed graphic signs or shapes to represent some feature or property of the world. Symbol recognition uses knowledge of the semantics attributed to these symbols and applies rules to extract them. Numerous attributes and techniques can be used to describe symbols geometrically and structurally in maps. In map processing tasks, the shape descriptors and morphological operators described in Section 3.2 are often combined and adjusted to tune the recognition process to the properties of the document of interest. Symbol recognition in maps—in particular in manually drawn maps—is challenging due to their graphical properties in general and the intrinsic meaning and characteristics of symbols in comparison to other documents, such as technical drawings [Cordella and Vento 2000; Lladós et al. 2002]. Traditionally, symbol recognition—also in map processing tasks—can be partitioned into statistical and structural approaches [Lladós et al. 2002], and the distinction from other recognition tasks such as character recognition (see below) is blurred. Note that this section focuses on the main efforts of recognizing singular symbols in maps; the case of composite objects or symbols is reviewed separately in the next subsection (Figure 18).

A few so-called knowledge-based systems have been developed for map processing tasks where knowledge is derived from the map of interest or its legend in order to improve the recognition process. For example, Samet and Soffer [1998] generate shape descriptors from the inverted images of individual symbols in a map; by creating a library of training sets of geographic features, they developed a legend-driven geographic symbol recognition system with high recognition rates. This innovative, statistical, pattern-recognition approach is limited by the time-consuming training and learning processes and a high degree of map specificity. Myers et al. [1996] focused on the problem of overlapping information in complex color maps and developed a verification-based approach using various knowledge bases of geometric (or shape) descriptors for improved recognition in USGS topographic maps. This was tested on complex wetland symbols.

Template matching, as explained earlier (Section 3.2), represents another knowledge-based recognition method. For example, Yin and Huang [2001] describe a supervised probabilistic template matching technique to extract road lines, characters, and symbols in Chinese road maps. Frischknecht and Kanani [1998] develop a hierarchical template matching approach for text and symbol extraction from modern topographic maps. In this approach, they weight critical areas within templates differently (emphasize or de-emphasize), thus improving the geometric description of the symbols of interest. Maderlechner and Mayer [1995] apply correlation-based template matching to detect symbols in Bavarian land register maps using explicit knowledge manually derived from the map legend, which makes this method highly map specific. Finally, Reiher et al. [1996] describe a symbol recognition system similar to the others described here but based on Hausdorff distance between symbol templates and the image; furthermore, they employ neural networks for their recognition process. The system is reported to be robust in detecting symbols of different scales and orientations.

Several studies on map symbol recognition have used morphological operators, and the results are often input to compute shape descriptors. For example, the feature extraction method described by Dhar and Chanda [2006] for various types of symbols (trees, grass, crop fields) in topographic maps applies initial noise removal, thinning, and skeletonization steps. They then compute attributes such as the Euler number (to define holes) and the Rutovitz connectivity number based on the skeletonized data. Erosion/dilation techniques have been used in combination with region growing for gap filling by Bessaid et al. [2003] in order to extract building symbols. For the extraction of final building contours, they use chain coding (Freeman coding) of contour points. This procedure browses contour pixels until there are no more neighbors or it reaches the starting pixel. The number of identical directions followed by the value of direction (Freeman direction) is recorded, and the result of this vectorization is written into a text file. This produces a high degree of compression compared to the original image files.

In a number of studies, shape descriptors are deployed without prior use of morphological operators. For example, in a multiangled parallelism (MAP) algorithm for multidirectional symbol recognition from topographic maps, Yamada et al. [1993] use a reformalized and parallel version of the generalized Hough transform without changing morphological properties of the objects of interest. Miyoshi et al. [2004] utilize a combination of simple and complex geometric descriptors for building extraction from scanned topographic maps without prior morphological operations. They identify limitations when symbols are merged with other map features such as roads. In an attempt to extract forest symbols from a historical topographic map, Leyk and Boesch [2009] describe existing shapes in the local neighborhood of a pixel also without prior morphological processing. These shapes are then evaluated by their similarity to shapes that are graphically defined as typical for the symbol of interest. As a final example, Trier et al. [1997] perform digit recognition in hydrographic maps for extracting symbols such as rock positions, digits, or signs without the use of morphological operators. For symbol recognition, they apply size- and orientation-independent elliptic Fourier descriptors of the outer symbol contour and use them as attributes in their statistical quadratic classifier.

Two graph-based approaches for symbol recognition in maps can be found in the recent literature. In an attempt to extract building information, Liu [2002] describes a method for house contour detection in the graph domain using breaking points of house structure; this method is demonstrated for large-scale pictorial maps. After separation from roads and characters, this approach shows satisfactory results. A connected component analysis for the extraction of quarters in historical French cadastral maps is based on a neighborhood graph [Raveaux et al. 2008]. Postprocessing of the results (pruning of the graph) is necessary to improve the recognition results.

Symbol recognition in maps has made significant progress using knowledge-based or legend-driven techniques, graph-based methods, or approaches that use advanced morphological operators combined with shape descriptors. However, most solutions are still map specific, and there is a need for more generic solutions in order to reach higher degrees of automation and robustness. The variety of different symbol recognition techniques described earlier illustrates how documents of different characteristics have been historically processed using different approaches. It is difficult to identify any clear relationships between techniques and types of maps, which may be the result of the rather experimental stage of map processing research to date.

4.3. Extraction and Recognition of Composite Features

Composite features consist of different graphic elements whose spatial distribution (or "composition") and pattern can define higher-level spatial objects of specific meaning. Examples of composite features are dotted or dashed paths [Gamba and Mecocci 1999], complex line work [Yamada et al. 1993; Zhong 2002], or areas whose spatial extent is defined by distributed single symbols such as circular symbols to describe forest areas [Leyk et al. 2006] or horizontal strokes to describe wetland areas (see examples in Figure 18). Processing such composite features in addition to single objects in maps represents a difficult task [Lladós et al. 2002], especially in maps of lower graphical quality.

The recognition of such compositions of single elements or symbols in different document types has traditionally been subject to principles of structural pattern recognition [Pavlidis 1977; Delalandre et al. 2004]. Structural recognition is based on the definition of spatial relations between graphical primitives (context), such as distances and directions. A set of rules is usually derived and applied to identify these complex symbols of interest in an image [Myers et al. 1996; Lladós et al. 2002].

In some map processing tasks, *contextual reasoning* is used to find certain types of objects in the vicinity of an already recognized object [Den Hartog et al. 1996; Ogier et al. 2001]. In a knowledge-directed image analysis effort, Ebi et al. [1994] demonstrate how recognition can be improved by identifying compositions of structure primitives on symbols of coniferous forest or symbols of dotted boundaries. Similar principles have been used for the extraction of compositions of character strings [Raveaux et al. 2008]. Another approach to structural recognition is based on the use of texture patterns in order to classify cartographic area features from maps [DeKruger and Hunt 1994].

In pursuing similar goals, Gamba and Mecocci [1999] describe a perceptual grouping approach to track discontinuous chains of symbols, which can be conceptualized as composite linear objects in topographic maps. Once individual symbols are extracted based on shape descriptors, Gestalt rules and optimal paths (i.e., shortest distance) are used to determine those symbols that are likely members of the line object and to close identified gaps. Their approach works well and has been used in numerous recognition studies that face similar problems. Finally, Leyk et al. [2006] use iterative sampling within a local search radius around a detected forest symbol to identify other circular objects in close vicinity in historical topographic maps, as shown in Figure 19. The spatial distribution of all identified forest symbols (their composition) defines the approximate extent of forest area.

Recent research that implemented the concept of composite features has demonstrated that there is great potential for improving feature extraction. Compositions of single graphical elements in maps often define meaningful higher-level objects characterized by their spatial distributions, topological relations, or their spatial context. Nevertheless, this concept is still underused in map processing possibly due to the complexity and computational costs that come with structural recognition approaches. However, solving these problems will be fundamental to efforts of map processing at higher levels of automation for complex map contents. As the previously mentioned studies demonstrate, the concept of composite features and the few techniques that have been described were reported to be successful in maps of different types and qualities.

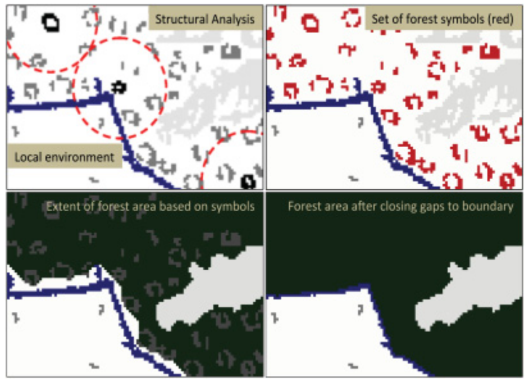


Fig. 19. Example of recognition of a composite feature using the method described in Leyk et al. [2006].

4.4. Extraction and Recognition of Linear Features

Much research has been conducted on extracting linear features from scanned maps, including the separation of lines from text [Li et al. 2000; Cao and Tan 2002; Tombre et al. 2002; Pezeshk and Tutwiler 2010b], the detection of road intersections [Habib et al. 1999; Henderson et al. 2009], the vectorization of road lines [Bin and Cheong 1998; Itonaga et al. 2003; Chiang and Knoblock 2013], the recognition of generic linear objects [Ablameyko et al. 2001, 2002a, 2002b; Bucha et al. 2007], and the recognition of contour lines [Arrighi and Soille 1999; Khotanzad and Zink 2003; Chen et al. 2006; Hancer and Samet 2011; Oka et al. 2012; Miao et al. 2013]. Linear feature extraction and recognition techniques usually consist of two steps: the separation of raster layers, which is often done through the image segmentation process described earlier, and the recognition of linear features from the individual raster layers.

Separating the raster layers is often based on histogram thresholding (Figure 20) to extract “raw” road layers. Figure 20 shows examples of well-conditioned maps on the left and the resulting road layers on the right. The top right panel shows an extracted road layer that also has few text labels and both solid road lines and road boundaries. In maps of lower quality, recognition of the linear features is typically based on the generation of centerline representations, which can require the application of different techniques for solid areas of linear features as compared to boundaries of linear features [Chiang and Knoblock 2013]. Most linear feature extraction techniques expect the extracted layer of linear features to be either areas [Itonaga et al. 2003] or boundaries of linear features [Bin and Cheong 1998; Habib et al. 1999] but not a combination of both types, which can significantly lower the accuracy of the final vectorization results. In addition, removing parts of or whole text labels from the raster layer of linear features prior to generating the centerlines has been done using text/graphics separation techniques [Cao and Tan 2002; Tombre et al. 2002; Velázquez and Levachkine 2004]. However, such approaches still require additional repair of disconnected centerlines where the overlapping text is removed [Chiang et al. 2009]. Furthermore, subsequent steps of text/graphics separation steps can be impaired when the result of line feature separation contains only pieces of the text labels, as can be seen at the bottom of Figure 20. These remaining text pixels constitute noisy objects that do not follow the criteria of character geometry typically used in text/graphics separation algorithms, such as a character is expected to be a group of small, straight line segments (strokes) [Cao and Tan 2002].

There are some techniques that exploit more sophisticated processes for extracting the raster layers potentially containing the linear features of interest. Such techniques

Fig. 20. Sample maps and their road layer extracted with manual grayscale histogram thresholding.

make it possible to process maps of inferior image quality. As mentioned in Sections 3.1 and 4.1, instead of thresholding the grayscale histogram, these techniques take advantage of the three-dimensional color space (e.g., RGB) and often employ a user-training step. For example, Salvatore and Guitton [2004] apply manual thresholds in the color space to separate the raster layer of contour lines from topographic maps. Khotanzad and Zink [2003] perform this CIS step based on user annotations to extract contour lines from the USGS topographic maps; this method was further exploited by Chen et al. [2006] in order to develop local segmentation techniques, which make it possible to process different topographic maps. In general, techniques that rely on user-trained thresholding in color space perform well on complex maps of inferior image quality. However, user-training processes such as manually determining color thresholds [Salvatore and Guitton 2004] or labeling colors of all line-background combinations [Khotanzad and Zink 2003] are complicated, laborious, and time-consuming.

Other techniques for line detection avoid relying heavily on user input. For example, Itonaga et al. [2003] exploit the geometric properties of individual map areas to determine road areas (e.g., elongated polygons); however, this approach does not work on scanned maps, because road areas are not represented as homogenous color areas in scanned maps. Henderson et al. [2009] present another interesting approach that does not rely on user input but rather exploits the metadata of the scanned maps to separate the road layers. In USGS DRG used to build the topographic maps, a set of 13 colors is used to draw the map features, and Henderson et al. [2009] exploit this predefined color scheme to isolate the road layer. Table I shows the USGS DRG color scheme.[2]

Table I. USGS DRG Color Scheme

Digital Number	Color	Red	Green	Blue
0	Black	0	0	0
1	White	255	255	255
2	Blue	0	151	164
3	Red	203	0	23
4	Brown	131	66	37
5	Green	201	234	157
6	Purple	137	51	128
7	Yellow	255	234	0
8	Light Blue	167	226	226
9	Light Red	255	184	184
10	Light Purple	218	179	214
11	Light Grey	209	209	209
12	Light Brown	207	164	142

Fig. 21. An example of USGS DRG based on a color scheme of 13 known colors to represent map layers. Credit: U.S. Geological Survey.

Figure 21 shows an example of the USGS DRG. Although their approach works well for separating the USGS DRG road layer due to the existence of the color key, it does not work if the color scheme does not exist or is unknown, which is common for most scanned maps. Henderson et al. [2009] indicate that a manual step for separating the road layer is required if the scanned map does not have a predefined color scheme. However, this manual step would involve labor-intensive user intervention before this approach could be applied to a different (scanned) map.

Once the raster layer of roads is separated from the map, raster-to-vector conversion techniques are used for the recognition of linear features from the individual raster layers. This raster-to-vector conversion step searches for elongated binary large objects (blobs) and then generates the centerline representation (skeletons) of the blobs.

[2]National mapping program technical instructions (http://gis.ess.washington.edu/data/raster/drg/docs/drg_std.txt).

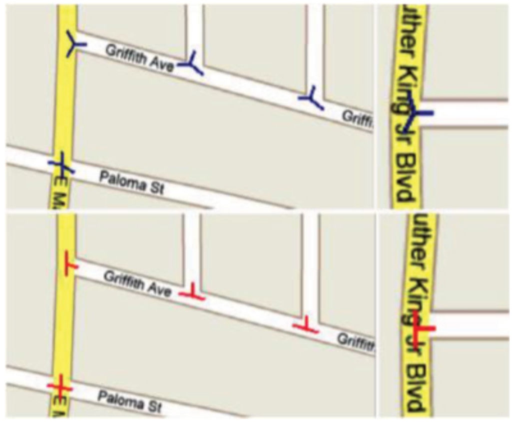


Fig. 22. Automatic repairing of the road geometry [Chiang and Knoblock 2008].

One approach to raster-to-vector conversion exploits the detected road boundaries to convert the road areas to vector format. Habib et al. [1999] use edge detection methods to identify road edges. Next, they identify corner points of these edges and determine the centroids of each group of corner points to represent road intersections. False-positive corner points or T-shaped road intersections can significantly shift the position of centroid points away from the correct locations. Bin and Cheong [1998] identify the medial axes of parallel road boundaries and link these medial axes to extract and vectorize road data from scanned maps. Linking the medial axes requires manual parameter specifications to ensure the generation of accurate representations of line geometry of road intersections.

Other techniques use morphological operators to remove noise (nonblobs), fill small gaps to reconnect broken lines, or determine the centerlines of linear features such as contour lines [Arrighi and Soille 1999] and road lines [Chiang and Knoblock 2013]. Since morphological operators usually distort the geometry of the original linear features when generating the centerlines (e.g., the road intersections in the top part of Figure 22 are displaced), alternative techniques that employ user-specified parameters to repair the road geometry have been developed. For example, Itonaga et al. [2003] correct the distorted lines around the road intersections based on user-specified constraints, such as the general road width in the considered map. In an attempt to automatically correct the line geometry, Arrighi and Soille [1999] use the centerline endpoints as anchor points to determine the geometry of centerlines. The extremities of the lines are then evaluated for connecting or disconnecting the line segments. Chiang and Knoblock [2013] detect potential areas with geometry distortions and trace line features that are outside these distortion areas to generate accurate centerlines. The top part of Figure 22 shows the distorted road geometry around road intersections when only the thinning operator is used to extract the road centerlines. The bottom part of Figure 22 shows the correct road geometry after detecting distortions automatically to trace accurate centerlines.

One technique that is reported to be not limited to a specific type of map is described in Bucha et al. [2007]. They apply the so-called pixel-force field algorithm [Bucha et al. 2006] to extract the centerlines of linear objects from scanned maps. The pixel-force field algorithm can be applied directly to color maps without the need of a preprocessing (i.e., binarization) step; however, the user needs to manually specify the start and end points of each linear feature, and this requirement significantly limits the automation scale of the pixel-force field algorithm.

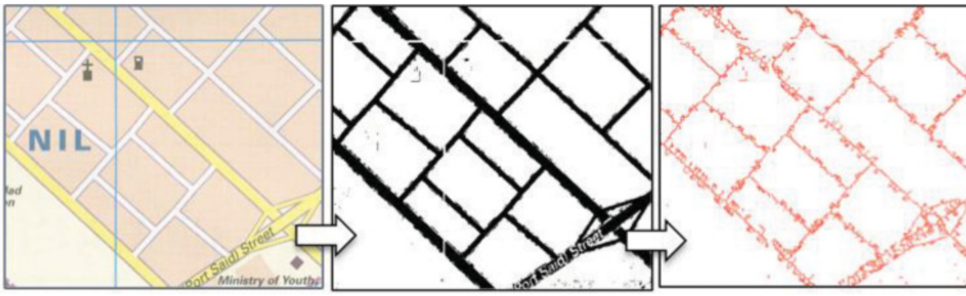


Fig. 23. A sample map processed by the automatic vectorization function in R2V.

In a series of publications, Ablameyko et al. [2001, 2002a, 2002b] present an interactive map interpretation system that supports the separation of feature layers in a scanned map based on user-selected colors. Selecting colors can be time-consuming, as scanned maps with inferior image quality can contain thousands of color values. Once colors are determined, this map interpretation system generates centerlines of linear features and contours of area features. The centerlines are then linked to generate the final results. The line-linking step also relies on user intervention, such as to confirm a merge between two line segments or to enforce the continuation of a line.

In order to complete the list of recent efforts, it is important to mention that there a few available commercial products for map processing, such as R2V from Able Software, which is an automated raster-to-vector conversion software package specialized in processing scanned maps. To vectorize roads in scanned maps using R2V, the user needs to first provide labels of road colors or select a set of color thresholds to identify the road pixels. As mentioned earlier, the manual procedure of providing labels of road pixels can be laborious, especially for scanned maps with numerous color values. In addition, color thresholding has limitations if, for example, one set of thresholds cannot completely separate the road pixels from pixels of other map features. In addition, the automatic vectorization function in R2V is sensitive to the road width without manual preprocessing, and Able Software does not provide details about this automatic vectorization technique. Figure 23 shows a sample map, the road layer, and the road centerlines extracted by R2V. The R2V automatic vectorization function produces small branches on a straight line when the road area is wide.

In summary, recent methods for the extraction and recognition of linear features from maps have largely improved in terms of reducing the required manual effort. However, the resulting vector datasets are still not ready to use in a GIS without manual postprocessing (e.g., manually identify the missing features). Additionally, although incorporating the well-developed vectorization techniques (e.g., Hori and Okazaki [1992], Röösli and Monagan [1995], Janssen and Vossepoel [1997]) in the extraction and recognition process can improve the quality of the resulting vector datasets (rather than simply using the morphological operators), a user still needs to manually validate the geometry and topology of the results. To further advance the overall user experience for the extraction and recognition of linear features, we believe that the best strategy is to enforce a tight integration between the user training steps (preprocessing), raster-to-vector conversion steps, and the postprocessing steps. Tight integration here means that the knowledge learned from a previous step can be passed into the next step and help improve the scale of automation in the next step. This allows a map processing technique to produce quality measures for each of the vectorization steps and then use the quality measures to automatically detect the areas that require manual editing. In this way, the cost of an end-to-end map processing task can be minimized.

Fig. 24. A French cadastral map from the ALPAGE (AnaLyse diachronique de l'espace urbain PArisien: approche GEomatique) project. Red rectangles show faded labels.

4.5. Extraction and Recognition of Text

Text recognition from maps is a difficult task because the text labels often overlap with other features in the maps, such as road lines, and the text labels do not follow a fixed orientation within a map [Nagy et al. 1997]. Therefore, the document structure analysis components in traditional OCR techniques often do not work well on maps [Mao et al. 2003]. In addition, depending on the quality of the original map, not every scanned map is a good candidate for text extraction and recognition. Raveaux et al. [2007, 2008] extract the text layer and identify individual text strings on a set of historical French cadastral maps. They first perform an antifading algorithm for restoring the faded colors in the historical maps. Then they use the expectation maximization (EM) algorithm to select the best color space from a set of color components (e.g., red, green, and blue for RGB or hue, saturation, and lightness for HSL) and then apply edge detection based on the chosen color space to identify individual connected components in the image. Finally, they classify the connected components into text features by detecting nearby connected components that are similar in size. Figure 24 shows a sample of the French cadastral map where the text labels are broken and faded.[3] Although Raveaux et al. overcame the difficulties to extract individual text strings from historical maps with poor graphical quality, they were not able to automatically process the identified text strings and further recognize the string characters.

Some text recognition techniques are highly specific to the type of map or the graphical representation of text elements. For example, Fletcher and Kasturi [1988] and Bixler [2000] assume that lines and text are not overlapping, which does not apply to most scanned maps. Chen and Wang [1997] utilize the Hough transform to group nearby numeric letters to identify straight numeric strings in scanned maps. They use a set of font and size independent features to recognize the numeric strings, which makes their approach specific to these numeric characters. Velázquez and Levachkine [2004] compute four imaginary lines of orientation (V-lines) that follow the main

[3]The ALPAGE project (http://websig.univ-lr.fr/alpage_public/flash/).

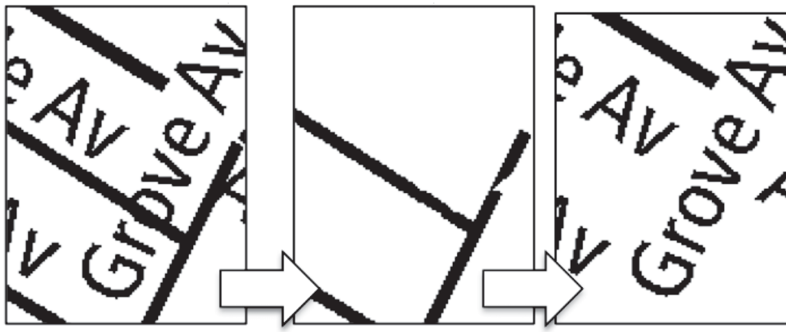


Fig. 25. Extrapolation of road lines to recover overlapping characters.

orientations of map text to separate straight text labels from graphics. They then recognize the text labels with artificial neural networks. Their method can handle slightly curved text labels by dividing a curved label into blocks of straight labels and then apply the V-line approach. This approach does not depend on the geometry of a single connected component, but it is not clear how curved labels are identified automatically.

For processing multioriented text, more general techniques exploit shape descriptors. Usually, sets of such descriptors are computed from character training samples and then utilized for pattern matching (or template matching) to recognize the characters in scanned maps. Deseilligny et al. [1995] compute rotation-invariant descriptors to identify multioriented text strings. Adam et al. [2000] compute descriptors based on the Fourier-Mellin transformation to compare the target characters with the trained character samples for recognizing text labels in scanned maps. Pezeshk and Tutwiler [2010a] produce artificial training data (i.e., a set of generic character images) from a set of initial training data using a truncated degradation model. They then recognize the map characters using two hidden Markov models with 10 states trained for each character class. The truncated degradation model helps to reduce the manual effort of generating training data. However, the user still has to generate an initial training set of character images for each font type. These methods [Deseilligny et al. 1995; Adam et al. 2000; Pezeshk and Tutwiler 2010a] require a considerable amount of training and user intervention for each input map and thus have limited applicability to other types of maps.

Another line of research on text recognition relies on text/graphics separation and subsequent recognition of the separated text strings. Li et al. [2000] use a novel integrated approach to separate text from lines and recover overlapping characters. They identify the feature layers and text labels using connected component analysis and extrapolate the feature layer (i.e., extend the linear objects in the feature layers) to identify and remove the lines that overlap with characters within each identified text label. Figure 25 illustrates this technique. The identified lines are extrapolated based on linear continuity to identify the pixels that are shared between the lines and text. Instead of assigning the overlapping pixels to either lines or text strings, they assign the pixels to both line and text, which prevents disconnected lines and broken characters and results in a high recognition rate. However, this method does not work for curved lines or text labels. Other text/graphics separation techniques are based on the geometric properties of the connected components in the text layer. Cao and Tan [2002] decompose the connected components into line segments and use a size filter to recover the character strokes that touch the lines. These text/graphics separation methods described here rely on manual work to prepare the separated text strings for

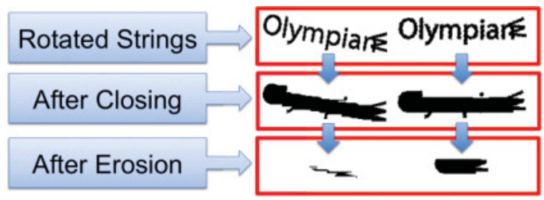


Fig. 26. Automatic string orientation detection [Chiang and Knoblock 2011].

OCR, such as manually rotating individual strings to the horizontal direction [Cao and Tan 2002].

The process of automatically rotating individual text strings to the horizontal direction as input for OCR is critical, because most OCR packages achieve the best results with horizontal strings. Pouderoux et al. [2007] apply connected component analysis followed by string analysis using parameters dynamically generated from the geometry of the connected components to identify string objects in the scanned maps. They then render the identified strings horizontally for character recognition using the average angle connecting the centroid points of the components in a string, which can be inaccurate when the characters have different heights or widths.

Chiang and Knoblock [2011] present a more general training-by-example approach that only requires a small user effort for text recognition in heterogeneous raster maps. Their approach exploits a few examples of text areas to first extract text pixels. Then they automatically expand text areas to locate individual text labels using cartographic labeling principles (e.g., the font size of a single text label will be consistent across the entire label). Next they use a robust skew detection method based on morphological operators [Najman 2004] to determine the orientation of each string and automatically rotate the strings to a horizontal orientation. Then they use commercial OCR software to recognize the rotated, horizontal strings. Figure 26 shows an example of the automatic orientation detection technique. For a given string, they generate 60 rotated strings by rotating the string clockwise from 0 to 180 degrees with a 3-degree increment. They then apply the closing and erosion operators on each of the rotated strings. Since a horizontal string has the largest number of character pixels near each other, horizontally, the orientation of the string with the largest number of remaining pixels after the erosion operator is searched; an example can be seen on the right side of Figure 26.

Other text recognition techniques include additional (or ancillary) information for identifying areas of text labels in the map images. Gelbukh et al. [2004] extend the algorithm described by Velázquez and Levachkine [2004] by exploiting additional information from a toponym database, linguistic dictionaries, and the spatial distributions of labels of various font sizes in a map to detect and correct errors introduced during the processes of text recognition. Myers et al. [1996] generate hypotheses about the potential characters in each text label and the locations of these text labels using a gazetteer. They identify text zones in the scanned maps by detecting and grouping short length patterns in the horizontal and vertical directions. The identified text zones are then verified based on the hypotheses generated from the gazetteer using a keyword recognition technique that detects the presence of possible text labels based on whole text shape within the text zone. The auxiliary information (e.g., gazetteers) used in these techniques is usually not easily accessible for map images.

These recent text recognition systems incorporate the underlying cartographic principles for map labeling into their text recognition procedures (i.e., build special document-structure-analysis techniques for maps) to achieve the best results. However, these text

Fig. 27. The user provides examples of road colors by selecting points in the center of segments.

recognition systems are map or language specific and are independent from the final character recognition process (consider OCR as a final, separate process). A key challenge here is how to build document analysis rules based on generic cartographic principles and tightly integrate these rules into the document analysis components in the state-of-the-art OCR systems.

4.6. Interactive Map Processing Systems

Maps can be quite noisy, and there is a lot of variation between maps; therefore, it is difficult, if not impossible, to fully automate the processing of maps. Thus, a number of systems exploit efficient interactive map processing techniques where the user provides some input to help in the recognition process. Two basic approaches have been used in various studies. The first approach allows a user to provide sample inputs to guide the extraction of the layers. The second approach allows the user to provide input to repair or improve the layers after they have been extracted. We will review each approach in turn.

A significant challenge in map processing is the variations in how layers are represented from one map to the next. One way to address this challenge is to allow the user to provide examples of the layer or layers that a user wants to extract from a map and use the examples to learn how to extract the layer(s). This has been done using several different methods. In the work by Wise [2002] on extracting soil maps, the user provides a seed pixel for each soil layer and then the system uses that seed pixel in a flood-fill algorithm that adds additional pixels if they are adjacent and within some threshold of the RGB value, hue, and brightness. The approach was successfully used to extract layers represented by dithered patterns of different colors, but they recognize that the results could depend on which seed pixel was selected for defining the threshold value. In the work by Bessaid et al. [2003], they extract the layers from four-color maps and the user provides examples of each of the four colors (brown, black, green, and blue) since the actual colors vary from one map to the next. In the work by Chiang and Knoblock [2013], they incorporate user input to identify multiple colors that have comprised the road lines in scanned maps. The user selects a single point in the center of a road (as shown in Figure 27) and the system then detects the road lines based on a Hough transform [Duda and Hart 1972] for each color. When the majority of the detected Hough lines are near the point that the user has selected, the system assigns that pixel color as a road color. This allows the system to determine all of the colors that comprise a road line using only a single point selected by the user. Chiang and Knoblock [2011] exploit a similar technique for determining the color of a text layer. The user provides an example of a text string, and the system uses the dominant pixel color to extract the text layer.

Since maps can be noisy and complex, a related challenge is extracting the layers when there is missing information, overlapping features, or disconnected lines. A number of systems have provided the ability for a user to interact with the system to correct the extracted layers [Suzuki and Yamada 1990; MapScan 1998; Ablameyko et al. 2001; Esri ArcScan[4]]. The capabilities of these systems are all quite similar in that they allow a user to interactively correct, add, or reject vector lines. Ablameyko et al. [2001] present a set of techniques that focus on connecting broken vectors by allowing the user to specify either two vectors to connect or the length of a sequence of vectors to connect. This addresses a common problem in map extraction where the vector layers are broken due to the overlap with other layers. The other work that is particularly notable is MapScan for Windows [MapScan 1998], which provides an extensive set of editing operations for dealing with extracted map vectors. In particular, the tool set provided by Mapscan includes the usual operations for adding, deleting, moving, and splitting nodes, as well as operations for adding, deleting, and connecting segments. However, Mapscan is a commercial system, and the details of the techniques used are not available. Although these interactive approaches are useful for processing a single map, the user interaction steps need to be repeated for each input map, and hence they do not scale well (in production) when there are a large number of maps to process.

Instead of building map processing systems that are specifically tailored to a particular map type, using interactive map processing techniques provides a general approach that makes it possible to process a wide variety of map types. The existing systems described earlier already demonstrate the generality of using interactive approaches. The remaining challenges are twofold. First, we need to develop interactive techniques that minimize the user input and then utilize that input to build end-to-end systems for extracting and recognizing features. Second, we need to develop systems that learn from the user input so that more of the processing can be automated over time.

4.7. Georegistering Maps

There are numerous maps—especially historical maps—that lack the metadata describing the details of the map location, datum, and projection used to create the map. In order to address this limitation and make use of the data extracted from these maps in a GIS environment, researchers have worked on various techniques for automatically georegistering a map. Commonly, there are three steps in the process of georegistering a map. The first step is to extract a set of reference features from the map that can be used for georegistering the document. The second step is to find a set of control point pairs that relate the extracted features with another georeferenced dataset. This is the most difficult step to automate, and in many commercial products (e.g., Esri ArcGIS), this step is performed manually. The third step is to then use this mapping to transform the map through rubbersheeting or conflation [Saalfeld 1993] so that the map is aligned with other geospatial data. Figure 28 shows the overall approach to georegistering a map.

In the first step, the features that are typically extracted from a map for georegistration are a set of road features such as a set of road intersections [Chen et al. 2008] or the entire road network [Chiang et al. 2009], as described in Section 4.4. Road features are often used because they tend to be graphically stable features that change infrequently (current roads in Europe often follow the same paths as the original Roman roads).

[4]Esri ArcScan (http://www.esri.com/software/arcgis/extensions/arcscan).

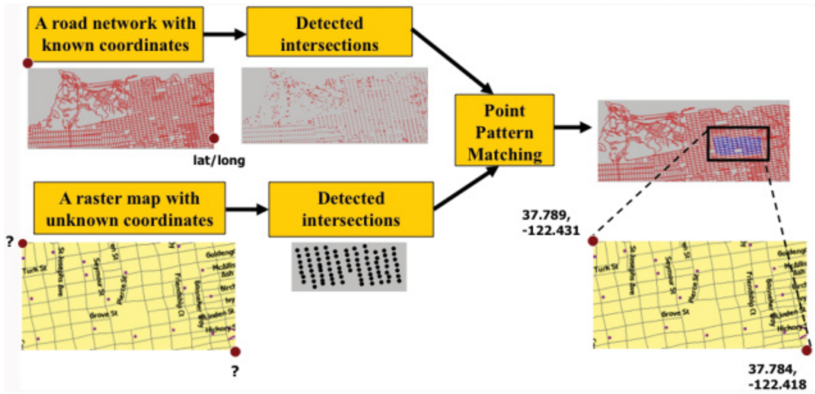


Fig. 28. The process of georegistering a map with unknown coordinates.

In the second step, the task is to determine the precise geolocation of the extracted features by comparing them against an already georeferenced dataset ("reference"). The georeferenced dataset is typically an accurate road network in vector format often from a current, digitally produced data layer. If the map to be processed already has some basic geolocation information (e.g., projection and geodetic datum used), then this is a relatively straightforward matching problem between the features extracted from the map and the georeferenced dataset. However, the more general case is a map with limited or no geolocation information where the problem is to find a small set of features extracted from a map and match them with a large set of georeferenced features in a large road network dataset (called an *asymmetric matching problem* [Li 2009]). This problem has been solved in several ways, depending on the assumptions made about the map. Chen et al. [2008] developed an approach that assumes the rotation of the map is known and then finds the translation and scale of the map using an efficient point pattern-matching algorithm (called *GeoPPM*). The Chen et al. approach is robust to noise and can deal with large numbers of missing or extraneous intersection points, which means that the technique works well on historical maps, which typically have fewer intersection points because there were typically fewer roads. More recently, Li [2009] developed the SIPPM matching algorithm, which maps the points and their relationships into polar coordinates to provide a technique that can support the mapping with rotation in addition to translation and scale. Their results show that they can map the features on a map to a georeferenced dataset with tens of thousands of points in less than 6 seconds, although the approach may be less tolerant to noise. At the conclusion of this step, these algorithms will produce a set of control points between the locations of the extracted features from a map and the corresponding features in the georeferenced dataset. These control points can then be analyzed to provide the precise geocoordinates, scale, and rotation (in the case of SIPPM) of a map. In addition, the complete set of control points provides a detailed mapping between the two datasets that can be used in the next step.

In the third step, the problem is to actually modify the map so that it aligns with the corresponding georeferenced dataset. This is particularly useful for either overlaying a map on top of an image or improving the alignment of an extracted vector layer from a map with other spatial layers. This step is based on a technique called *conflation* developed by Saalfeld [1993], which addresses the problem of how to merge two vector maps and their features. In this approach, a map is partitioned into a set of triangular regions using Delaunay triangulation. Delaunay triangulation is used because it maximizes the angles of the triangulations, which reduces the possible distortion

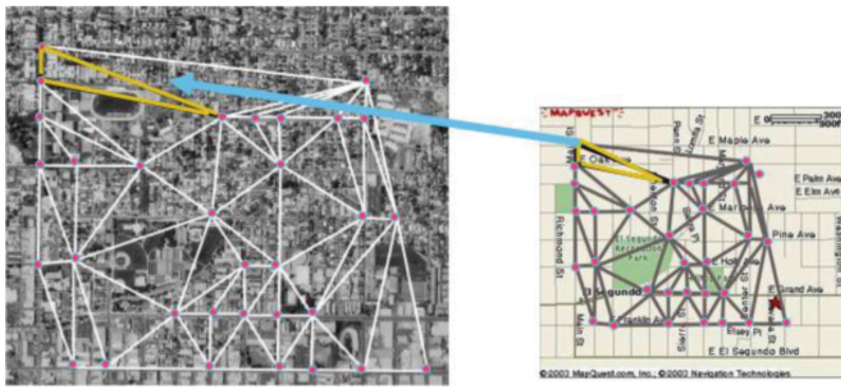


Fig. 29. Rubbersheeting a map to an image.

introduced in this process. Using these triangular regions, the map or map layer can then be rubbersheeted to the georeferenced dataset so that the two layers are aligned (as shown in Figure 29). In the case of a raster layer, the map would be stretched to align with the other layer; in the case of a vector layer, the vector geometry would simply be updated based on the alignment of the triangles.

In general, the work described previously has shown that it is possible to both find and exploit control points to automatically register a map even when the metadata describing a map is missing. However, these techniques still assume that the general location of the map is known and that there exists a reference dataset for that region. An interesting challenge is to build a reference dataset for the whole world and develop techniques that can automatically register a map to a location anywhere in the world and determine the correct metadata for that map.

5. CHALLENGES AND OPPORTUNITIES

As seen in this article, there has been a significant amount of interest in and research on map processing over the years. Existing research efforts have been scattered over many organizations and countries, have addressed a wide variety of issues, and have been published in many different venues. The authors hope that this article will provide a more integrated view of the large body of work that has been conducted. Past research has typically focused on specific map types, because the research is often driven by the need to extract the data from a specific set or series of maps. Although much progress has been made on solving numerous problems in map processing, there are many, many problems left to solve. Furthermore, we believe that the ability to extract and georeference the data in maps will unlock a wealth of information that has many applications, including constructing new maps, building better and more detailed gazetteers, performing historical research about areas that have changed over time, and conducting medical research on links between man-made features and diseases.

Given the history of scattered funding and the narrow focus on specific map types, the question is how to accelerate the progress in the area of map processing. We believe that this can be best achieved by encouraging researchers to make their software and datasets available. This will allow other researchers to build on the current state of the art instead of having to reimplement each component individually. The availability of datasets also means that researchers can then directly compare their techniques against other algorithms and evaluate their techniques on maps that have been processed by others. If researchers make their software available under open source licenses, it also means that we can begin to put together integrated systems that combine the best algorithms, which will then provide a set of tools that can be used by others for extracting data from their own maps. As has been described in this survey, it is difficult to draw direct linkages between certain techniques and types or qualities of maps. Map processing still depends on the expertise of the analyst to choose an appropriate technique for a particular extraction task depending on the type and quality of the map document at hand. Thus, currently there is no unified or generic map processing framework that would make it possible to select the best-performing techniques and methods for map processing work for various kinds of maps. However, if the community can be better integrated, a systematic understanding of existing solutions can be developed and evaluated to provide the basis for an initial framework in the near future.

Beyond the issue of developing a more integrated community, there are also several areas of specific research that we want to highlight as deserving more attention. First, processing maps is hard due to the complex overlapping data provided in many maps. Thus, we believe that it is especially important to develop interactive techniques that reduce the user effort in map processing and allow users to apply their own expertise to extract, verify, and even correct the extracted data. By developing efficient interactive techniques, we can better exploit the knowledge and human abilities for understanding map representations and possibly integrate learning processes to improve the systems. Second, there is a vast amount of historical data captured in maps that could have a wide variety of uses, but often the historical maps are the hardest documents to process. Therefore, we believe that there should be more focus and more research funding directed to the problems of extracting data from noisy maps. Third, little work has addressed the problem of the provenance, accuracy, and reliability of the extracted results. As map processing research comes into its prime, there is the opportunity to extract a great deal of information from maps. As a community, we need to record the provenance of these extracted data, develop techniques to estimate the accuracy, and provide methods to quantify the reliability of the generated spatial data.

ACKNOWLEDGMENTS

We thank Dr. Reid Priedhorsky for his work on preparing the scanned images in Figures 4 and 5. We also thank the anonymous reviewers for their helpful suggestions.

REFERENCES

S. Ablameyko, V. Beveisbik, M. Homenko, N. Paramonova, and O. Patsko. 2001. Interpretation of colour maps. A combination of automatic and interactive techniques. *Computing Control Engineering Journal* 12, 4, 188–196.

S. Ablameyko, V. Bereishik, M. Homenko, D. Lagunovsky, N. Paramonova, and O. Patsko. 2002a. A complete system for interpretation of color maps. *International Journal of Image and Graphics* 2, 3, 453–479.

S. Ablameyko, V. Bereishik, M. Homenko, N. Paramonova, and O. Patsko. 2002b. Automatic/interactive interpretation of color map images. In *Proceedings of the 16th International Conference on Pattern Recognition*. 69–72.

S. Adam, J.-M. Ogier, C. Cariou, R. Mullot, J. Labiche, and J. Gardes. 2000. Symbol and character recognition: Application to engineering drawings. *International Journal on Document Analysis and Recognition* 3, 2, 89–101.

G. Agam and I. Dinstein. 1996. Generalized morphological operators applied to map-analysis. In *Proceedings of the 6th International Workshop on Advances in Structural and Syntactical Pattern Recognition*. 60–69.

J. Angulo and J. Serra. 2003. Mathematical morphology in color spaces applied to the analysis of cartographic images. *Semantic Processing of Spatial Data, Research on Computing Science* 4, 59–66.

P. Arrighi and P. Soille. 1999. From scanned topographic maps to digital elevation models. In *Proceedings of the International Symposium on Imaging Applications in Geology*. 1–4.

P. Bajcsy. 2003. Automatic extraction of isocontours from historical maps. In *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics*. 99–104.

G. H. Ball and D. J. Hall. 1965. *ISODATA, a Novel Method of Data Analysis and Classification*. Technical Report, Stanford University, Stanford, CA.

A. Bessaid, H. Bechar, and J.-M. Ogier. 2003. Automatic topographic color map analysis system. In *Proceedings of the 3rd International Workshop on Pattern Recognition in Information Systems*. 191–196.

D. Bin and W. K. Cheong. 1998. A system for automatic extraction of road network from maps. In *Proceedings of the IEEE International Joint Symposia on Intelligence and Systems*. 359–366.

J. P. Bixler. 2000. Tracking text in mixed-mode documents. In *Proceedings of the ACM Conference on Document Processing Systems*. 177–185.

L. Boatto, V. Consorti, M. Del Buono, S. Di Zenzo, V. Eramo, A. Esposito, F. Melcarne, M. Meucci, A. Morelli, M. Mosciatti, S. Scarci, and M. Tucci. 1992. An interpretation system for land register maps. *Computer* 25, 7, 25–33.

S.-T. Bow. 1992. *Pattern Recognition and Image Preprocessing*. Marcel Dekker, New York, NY.

E. O. Brigham. 1988. *The Fast Fourier Transform and Its Applications*. Prentice Hall, Englewood Cliffs, NJ.

K. Broelemann and X. Jiang. 2011. A region-based method for sketch map segmentation. In *Graphics Recognition. New Trends and Challenges*, Y.-B. Kwon and J.-M. Ogier, Eds., Lecture Notes in Computer Science, Vol. 7423. Springer, Berlin, 1–14.

R. Brunelli. 2009. *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley, Hoboken, NJ.

V. Bucha, S. Uchida, and S. Ablameyko. 2006. Interactive road extraction with pixel force fields. In *Proceedings of the 18th International Conference on Pattern Recognition*. 829–832.

V. Bucha, S. Uchida, and S. Ablameyko. 2007. Image pixel force fields and their application for color map vectorisation. In *Proceedings of the 9th International Conference on Document Analysis and Recognition*. 1228–1242.

P. Campadelli, D. Medici, and R. Schettini. 1997. Color image segmentation using Hopfield networks. *Image and Vision Computing Image*, 15, 3, 161–166.

R. Cao and C. L. Tan. 2002. Text/graphics separation in maps. In *Proceedings of the 4th IAPR International Workshop on Graphics Recognition Algorithms and Applications*. 167–177.

J. Centeno. 1998. Segmentation of thematic maps using colour and spatial attributes. In *Proceedings of the 2nd IAPR Workshop on Graphics Recognition*, K. Tombre and A. K. Chhabra, Eds., Lecture Notes in Computer Science, Vol. 1389. Springer, Berlin, 221–230.

C.-C. Chen, C. A. Knoblock, and C. Shahabi. 2008. Automatically and accurately conflating raster maps with orthoimagery. *GeoInformatica* 12, 3, 377–410.

C.-C. Chen, C. Shahabi, and C. A. Knoblock. 2004. Utilizing road network data for automatic identification of road intersections from high resolution color orthoimagery. In *Proceedings of the 2nd Workshop on Spatio-Temporal Database Management*. 1477–1480.

L.-H. Chen, H.-Y. Liao, J.-Y. Wang, and K.-C. Fan. 1999. Automatic data capture for geographic information systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 29, 2, 205–215.

L.-H. Chen and J.-Y. Wang. 1997. A system for extracting and recognizing numeral strings on maps. In *Proceedings of the 4th International Conference on Document Analysis and Recognition*. 337–341.

Y. Chen, R. Wang, and J. Qian. 2006. Extracting contour lines from common-conditioned topographic maps. *IEEE Transactions on Geoscience and Remote Sensing* 44, 4, 1048–1057.

H. D. Cheng, C. H. Chen, H. H. Chiu, and H. Xu. 1998. Fuzzy homogeneity approach to multilevel thresholding. *IEEE Transactions on Image Processing* 7, 7, 1084–1088.

H. D. Cheng, X. H. Jiang, Y. Sun, and J. L. Wang. 2001. Color image segmentation: Advances and prospects. *Pattern Recognition* 34, 2259–2281.

H. D. Cheng, X. H. Jiang, and J. Wang. 2002. Color image segmentation based on homogram thresholding and region merging. *Pattern Recognition* 35, 373–393.

H. D. Cheng and Y. Sun. 2000. A hierarchical approach to color image segmentation using homogeneity. *IEEE Transactions on Image Processing* 9, 12, 2071–2082.

Y.-Y. Chiang and C. A. Knoblock. 2006. Classification of line and character pixels on raster maps using discrete cosine transformation coefficients and support vector machine. In *Proceedings of the 18th International Conference on Pattern Recognition*. 1034–1037.

Y.-Y. Chiang and C. A. Knoblock. 2008. Automatic extraction of road intersection position, connectivity, and orientations from raster maps. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 1–10.

Y.-Y. Chiang and C. A. Knoblock. 2009. Classification of raster maps for automatic feature extraction. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 138–147.

Y.-Y. Chiang and C. A. Knoblock. 2011. Recognition of multi-oriented, multi-sized, and curved text. In *Proceedings of the 11th International Conference on Document Analysis and Recognition*. 1399–1403.

Y.-Y. Chiang and C. A. Knoblock. 2013. A general approach for extracting road vector data from raster maps. *International Journal on Document Analysis and Recognition* 16, 1, 55–81.

Y.-Y. Chiang, C. A. Knoblock, C. Shahabi, and C. Chen. 2009. Automatic and accurate extraction of road intersections from raster maps. *GeoInformatica* 13, 2, 121–157.

Y.-Y. Chiang, S. Leyk, and C. A. Knoblock. 2012. Efficient and robust graphics recognition from historical maps. In *Graphics Recognition. New Trends and Challenges*, Y.-B. Kwon and J.-M. Ogier, Eds., Lecture Notes in Computer Science, Vol. 7423. Springer, Berlin, 25–35.

R. H. Cofer and J. T. Tou. 1972. Automated map reading and analysis by computer. In *Proceedings of the AFIPS Joint Computer Conferences, Part I*. 135–145.

D. Comaniciu and P. Meer. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 5, 603–619.

A. Cordeiro and P. Pina. 2006. Colour map object separation. In *Proceedings of the ISPRS Mid-Term Symposium 2006, Remote Sensing: From Pixels to Processes*. 243–247.

L. P. Cordella and M. Vento. 2000. Symbol and shape recognition. In *Proceedings of the 3rd International Workshop on Graphics Recognition, Recent Advances (GREC'99)*. Springer-Verlag, London, UK, 167–182.

D. DeKruger and B. Hunt. 1994. Image processing and neural networks for recognition of cartographic area features. *Pattern Recognition* 27, 4, 461–483.

M. Delalandre, E. Trupin, and J.-M. Ogier. 2004. Local structural analysis: A primer. In *Proceedings of the 5th IAPR Workshop on Graphics Recognition*, Y.-B. Kwon and J. Lladós, Eds., Lecture Notes in Computer Science, Vol. 3088. Springer, Berlin, 223–234.

J. Den Hartog, T. ten Kate, and J. Gerbrands. 1996. Knowledge-based segmentation for automatic map interpretation. In *Graphics Recognition Methods and Applications*, R. Kasturi and K. Tombre, Eds., Lecture Notes in Computer Science, Vol. 1072. Springer, Berlin, 159–178.

M. P. Deseilligny, H. L. Mena, and G. Stamonb. 1995. Character string recognition on maps, a rotation-invariant recognition method. *Pattern Recognition Letters* 16, 12, 1297–1310.

D. B. Dhar and B. Chanda. 2006. Extraction and recognition of geographical features from paper maps. *International Journal of Document Analysis and Recognition* 8, 4, 232–245.

S. Di Zenzo, L. Cinque, and S. Levialdi. 1996. Run-based algorithms for binary image analysis and processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18, 1, 83–89.

C. Dietzel, M. Herold, J. J. Hemphill, and K. C. Clarke. 2005. Spatiotemporal dynamics in California's Central Valley: Empirical links to urban theory. *International Journal of Geographical Information Science* 19, 2, 175–195.

D. Doermann. 1998. An introduction to vectorization and segmentation. In *Graphics Recognition Algorithms and Systems*, K. Tombre and A. Chhabra, Eds., Lecture Notes in Computer Science, Vol. 1389. Springer, Berlin, 1–8.

R. O. Duda and P. E. Hart. 1972. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM* 15, 11–15.

F. Dupont, M. Deseilligny, and M. Gondran. 1998. Automatic interpretation of scanned maps: Reconstruction of contour lines. In *Graphics Recognition Algorithms and Systems*, K. Tombre and A. Chhabra, Eds., Lecture Notes in Computer Science, Vol. 1389. Springer, Berlin, 194–206.

N. Ebi, B. Lauterbach, and W. Anheier. 1994. An image analysis system for automatic data acquisition from colored scanned maps. *Machine Vision and Applications* 7, 148–164.

M. Ejiri, S. Kakumoto, T. Miyatake, S. Shimada, and H. Matsushima. 1984. Automatic recognition of design drawings and maps. In *Proceedings of the International Conference on Pattern Recognition*. 1296–1305.

R. Fayek and A. Wong. 1996. Extracting buildings from aerial topographic maps. In *Proceedings of the International Conference on Image Processing*. 401–404.

L. A. Fletcher and R. Kasturi. 1988. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10, 6, 910–918.

S. Frischknecht and E. Kanani. 1998. Automatic interpretation of scanned topographic maps: A raster-based approach. In *Graphics Recognition Algorithms and Systems*, K. Tombre and A. Chhabra, Eds., Lecture Notes in Computer Science, Vol. 1389. Springer, Berlin, 207–220.

S. Frischknecht, E. Kanani, and A. Carosio. 1998. A raster-based approach for the automatic interpretation of topographic maps. *IAPRS* 32, 523–530.

H. Freeman and G. G. Pieroni, Eds. 1982. *Map Data Processing*. Academic Press, London.

K. Fu and J. Mui. 1981. A survey on image segmentation. *Pattern Recognition* 13, 1, 3–16.

H. T. Fung and K. J. Parker. 1996. Segmentation of scanned document images for efficient compression. In *Proceedings of SPIE 2727*, *Visual Communications and Image Processing*. 701.

P. Gamba and A. Mecocci. 1999. Perceptual grouping for symbol chain tracking in digitized topographic maps. *Pattern Recognition Letters* 20, 4, 355–365.

A. Gelbukh, S. Levachkine, and S.-Y. Han. 2004. Resolving ambiguities in toponym recognition in cartographic maps. In *Proceedings of the 5th IAPR International Workshop on Graphics Recognition*. 104–112.

J. Gllavata, R. Ewerth, and B. Freisleben. 2004. Text detection in images based on unsupervised classification of high-frequency wavelet coefficients. In *Proceedings of the 17th International Conference on Pattern Recognition*, 425–428.

A. Goel, M. Michelson, and C. A. Knoblock. 2011. Harvesting maps on the Web. *International Journal on Document Analysis and Recognition* 14, 4, 349–372.

A. Habib, R. Uebbing, and A. Asmamaw. 1999. *Automatic Extraction of Road Intersections from Raster Maps*. Project report submitted to the Center for Mapping, The Ohio State University.

E. Hancer and R. Samet. 2011. Advanced contour reconnection in scanned topographic maps. In *Proceedings of the 5th International Conference on Application of Information and Communication Technologies*. 1–5.

T. Henderson and T. Linton. 2009. Raster map image analysis. In *Proceedings of the 10th International Conference on Document Analysis and Recognition*. 376–380.

T. C. Henderson, T. Linton, S. Potupchik, and A. Ostanin. 2009. Automatic segmentation of semantic classes in raster map images. In *Proceedings of the 8th IAPR International Workshop on Graphics Recognition*. 253–262.

B. L. Hickman, M. P. Bishop, and M. V. Rescigno. 1995. Advanced computational methods for spatial information extraction. *Computers and Geosciences* 21, 1, 153–173.

M. E. Hodgson, X. Li, and Y. Cheng. 2004. A parameterization model for transportation feature extraction. *Photogrammetric Engineering and Remote Sensing* 70, 12, 1399–1404.

O. Hori and A. Okazaki. 1992. High quality vectorization based on a generic object model. *Structured Document Image Analysis*, 325–349.

W. Itonaga, I. Matsuda, N. Yoneyama, and S. Ito. 2003. Automatic extraction of road networks from map images. *Electronics and Communications in Japan, Part II: Electronics* 86, 4, 62–72.

R. Jain, R. Kasturi, and B. G. Schunck. 1995. *Machine Vision*. McGraw-Hill, New York, NY.

R. D. T. Janssen and A. M. Vossepoel. 1997. Adaptive vectorization of line drawing images. *Computer Vision and Image Understanding*, 38–56.

R. Kasturi and J. Alemany. 1988. Information extraction from images of paper-based maps. *IEEE Transactions on Software Engineering* 14, 5, 671–675.

E. Katona and G. Hudra. 1999. An interpretation system for cadastral maps. In *Proceedings of the International Conference on Image Analysis and Processing*. 792–797.

A. Khotanzad and E. Zink. 1996. Color paper map segmentation using eigenvector line-fitting. In *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*. 190–194.

A. Khotanzad and E. Zink. 2003. Contour line and geographic feature extraction from USGS color topographical paper maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 1, 18–31.

F. Kienast. 1993. Analysis of historic landscape patterns with a geographical information system—a methodological outline. *Landscape Ecology* 8, 2, 103–118.

J. Kozak, C. Estreguil, and M. Troll. 2007. Forest cover changes in the northern Carpathians in the 20th century: A slow transition. *Journal of Land Use Science* 2, 2, 127–146.

F. W. Leberl and D. Olson. 1982. Raster scanning for operational digitizing of graphical data. *Photogrammetric Engineering and Remote Sensing* 48, 4, 615–672.

S. Leyk. 2010. Segmentation of colour layers in historical maps based on hierarchical colour sampling. In *Graphics Recognition. Achievements, Challenges, and Evolution*, J.-M. Ogier, W. Liu, and J. Lladós, Eds., Lecture Notes in Computer Science, Vol. 6020. Springer, Berlin, 231–241.

S. Leyk and R. Boesch. 2009. Extracting composite cartographic area features in low-quality maps. *Cartography and Geographical Information Science* 36, 1, 71–79.

S. Leyk and R. Boesch. 2010. Colors of the past: Color image segmentation in historical topographic maps based on homogeneity. *GeoInformatica* 14, 1, 1–21.

S. Leyk, R. Boesch, and R. Weibel. 2006. Saliency and semantic processing: Extracting forest cover from historical topographic maps. *Pattern Recognition* 39, 5, 953–968.

L. Li, G. Nagy, A. Samal, S. C. Seth, and Y. Xu. 2000. Integrated text and line-art extraction from a topographic map. *International Journal of Document Analysis and Recognition* 2, 4, 177–185.

Y. Li. 2009. An automated system for image-to-vector georeferencing. Ph.D. Dissertation. University of Texas, Dallas, TX.

Y. Liu. 2002. An automation system: Generation of digital map data from pictorial map resources. *Pattern Recognition* 35, 9, 1973–1987.

J. Lladós, E. Valveny, G. Sanchez, and E. Marti. 2002. Symbol recognition: Current advances and perspectives. In *Graphics Recognition Algorithms and Applications*, D. Blostein and Y.-B. Kwon, Eds., Lecture Notes in Computer Science, Vol. 2390. Springer, Berlin, 104–128.

L. Lucchese and S. K. Mitra. 2001. Color image segmentation: A state-of-the-art survey. *Proceedings of the Indian National Science Academy (INSA-A)*, 67, 2, 207–221.

H. Luo and R. Kasturi. 1998. Improved directional morphological operations for separation of characters from maps/graphics. In *Graphics Recognition Algo-rithms and Systems*, K. Tombre and A. Chhabra, Eds., Lecture Notes in Computer Science, Vol. 1389. Springer, Berlin, 35–47.

G. Maderlechner and H. Mayer. 1995. Conversion of high level information from scanned maps into geographic information systems. In *Proceedings of the International Conference on Document Analysis and Recognition*. 253–256.

S. Mao, A. Rosenfeld, and T. Kanungo. 2003. Document structure analysis algorithms: A literature survey. *Document Recognition and Retrieval X* 5010, 197–207.

MapScan. 1998. *MapScan for Windows Software Package for Automatic Map Data Entry, User's Guide and Reference Manual*. Computer Software and Support for Population Activities, INT/96/P74, United Nations Statistics Division, New York, NY.

J. McQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*. 281–296.

C. A. B. Mello, D. C. Costa, and T. J. dos Santos. 2012. Automatic image segmentation of old topographic maps and floor plans. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 132–137.

Q. Miao, P. Xu, T. Liu, Y. Yang, J. Zhang, and W. Li. 2013. Linear feature separation from topographic maps using energy density and the shear transform. *IEEE Transactions on Image Processing* 22, 4, 1548–1558.

T. Miyoshi, W. Li, K. Kaneda, H. Yamashita, and E. Nakamae. 2004. Automatic extraction of buildings utilizing geometric features of a scanned topographic map. In *Proceedings of the 17th International Conference on Pattern Recognition*. 626–629.

S. P. Morse. 1969. Concepts of use in contour map processing. *Communications of the ACM* 12, 3, 147–152.

G. K. Myers, P. G. Mulgaonkar, C.-H. Chen, J. L. DeCurtins, and E. Chen 1996. Verification-based approach for automated text and feature extraction from raster-scanned maps. In *Graphics Recognition Methods and Applications*, Lecture Notes in Computer Science, Vol. 1072. Springer, 190–203.

G. Nagy, A. Samal, S. Seth, T. Fisher, E. Guthmann, K. Kalafala, L. Li, S. Sivasubramaniam, and Y. Xu. 1997. Reading street names from maps—technical challenges. In *Proceedings of the GIS/LIS Conference*. 89–97.

L. Najman. 2004. Using mathematical morphology for document skew estimation. In *Proceedings of the SPIE Document Recognition and Retrieval IX*. 182–191.

J.-M. Ogier, S. Adam, A. Bessaid, and H. Bechar. 2001. Automatic topographic color map analysis system. In *Proceedings of the 4th IAPR Workshop on Graphics Recognition*. 229–244.

S. Oka, A. Garg, and K. Varghese. 2012. Vectorization of contour lines from scanned topographic maps. *Automation in Construction* 22, 192–202.

N. R. Pal and S. K. Pal. 1993. A review on image segmentation techniques. *Pattern Recognition* 26, 9, 1277–1294.

T. Pavlidis. 1977. *Structural Pattern Recognition*. Springer, New York, NY.

C. C. Petit and E. F. Lambin. 2002. Impact of data integration technique on historical land-use/land-cover change: Comparing historical maps with remote sensing data in the Belgian Ardennes. *Landscape Ecology* 17, 2, 117–132.

A. Pezeshk and R. L. Tutwiler. 2008. Contour line recognition and extraction from scanned colour maps using dual quantization of the intensity image. In *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*. 173–176.

A. Pezeshk and R. L. Tutwiler. 2010a. Extended character defect model for recognition of text from maps. In *Proceedings of the IEEE Southwest Symposium on Image Analysis Interpretation*. 85–88.

A. Pezeshk and R. L. Tutwiler. 2010b. Improved multi angled parallelism for separation of text from intersecting linear features in scanned topographic maps. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. 1078–1081.

A. Pezeshk and R. L. Tutwiler. 2011. Automatic feature extraction and text recognition from scanned topographic maps. *IEEE Transactions on Geoscience and Remote Sensing* 49, 12, 5047–5063.

J. Pouderoux, J. C. Gonzato, A. Pereira, and P. Guitton. 2007. Toponym recognition in scanned color topographic maps. In *Proceedings of the 9th International Conference on Document Analysis and Recognition*. 531–535.

W. K. Pratt. 2001. *Digital Image Processing: PIKS Scientific Inside*, 3rd ed. Wiley-Interscience.

T. Randen and J. Husoy. 1999. Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 21, 4, 291–310.

R. Raveaux, E. Barbu, H. Locteau, S. Adam, P. Héroux, and E. Trupin. 2007. A graph classification approach using a multi-objective genetic algorithm application to symbol recognition. In *Graph-Based Representations in Pattern Recognition*, F. Escolano and M. Vento, Eds., Lecture Notes in Computer Science, Vol. 4538. Springer, 361–370.

R. Raveaux, J.-C. Burie, and J.-M. Ogier. 2008. Object extraction from colour cadastral maps. In *Proceedings of the IAPR International Workshop on Document Analysis Systems*. 506–514.

E. Reiher, L. Ying, V. Delle Donne, M. Lalonde, C. Hayne, Z. Chona. 1996. A system for efficient and robust map symbol recognition. In *Proceedings of the 13th International Conference on Pattern Recognition*. 783–787.

A. Rosenfeld and A. C. Kak. 1982. *Digital Picture Processing*, 2nd ed. Academic Press, New York, NY.

M. Röösli and G. Monagan. 1995. A high quality vectorization combining local quality measures and global constraints. In *Proceedings of the 3rd International Conference on Document Analysis and Recognition*. 243–248.

A. Rosenfeld and G. J. VanderBrug. 1977. Coarse-fine template matching. *IEEE Transactions on Systems Man and Cybernetics* 7, 2, 104–107.

A. J. Saalfeld. 1993. Conflation: Automated map compilation. Ph.D. Dissertation. University of Maryland at College Park, College Park, MD.

P. K. Sahoo, S. Soltani, A. K. C. Wong, and Y. C. Chen. 1988. A survey of thresholding techniques. *Computer Vision, Graphics, and Image Processing* 41, 233–260.

J. P. Salmon, L. Wendling, and S. Tabbone. 2007. Improving the recognition by integrating the combination of descriptors. *International Journal on Document Analysis and Recognition* 9, 1, 3–12.

S. Salvatore and P. Guitton. 2004. Contour line recognition from scanned topographic maps. In *Proceedings of the Winter School of Computer Graphics*. 1–3.

H. Samet and A. Soffer. 1998. MAGELLAN: Map acquisition of geographical labels by legend analysis. *International Journal on Document Analysis and Recognition* 1, 2, 89–101.

R. Samet and E. Hancer. 2012. A new approach to the reconstruction of contour lines extracted from topographic maps. *Journal of Visual Communication and Image Representation* 23, 4, 642–647.

L. M. San, S. M. Yatim, N. A. M. Sheriff, and N. Isrozaidi. 2004. Extracting contour lines from scanned topographic maps. In *Proceedings of the International Conference on Computer Graphics, Imaging and Visualization*. IEEE Computer Society, Washington, DC, 187–192.

R. Santos, T. Ohashi, T. Yoshida, and T. Ejima. 1998. Filtering and segmentation of digitized land use map images. *International Journal on Document Analysis and Recognition* 1, 167–174.

G. Sharma and J. Trussel. 1997. Digital color imaging. *IEEE Transactions on Image Processing* 6, 7, 901–932.

C. Steger, H. Mayer, and B. Radig. 1997. The role of grouping for road extraction. In *Automatic Extraction of Man-Made Objects from Aerial and Space Images (II)*, 245–256.

J. Stoffel and J. Moreland. 1981. A survey of electronic techniques for pictorial image reproduction. *IEEE Transactions on Communications* 29, 12, 1898–1925.

S. Suzuki and T. Yamada. 1990. Maris: Map recognition input system. *Pattern Recognition* 23, 8, 919–933.

K. Tombre, C. Ah-Soon, P. Dosch, G. Masini, and S. Tabbone. 2000. Stable and robust vectorization: How to make the right choices. In *Graphics Recognition Recent Advances*, A. Chhabra and D. Dori, Eds., Lecture Notes in Computer Science, Vol. 1941. Springer, Berlin, 3–18.

K. Tombre, S. Tabbone, L. Pélissier, B. Lamiroy, and P. Dosch. 2002. Text/graphics separation revisited. In *Proceedings of the 5th International Workshop on Document Analysis Systems V (DAS'02)*, D. P. Lopresti, J. Hu, and R. S. Kashi, Eds., Springer-Verlag, London, UK, 200–211.

O. Trier, T. Taxt, and A. Jain. 1997. Recognition of digits in hydrographic maps: Binary versus topographic analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 4, 399–404.

A. Velázquez and S. Levachkine. 2004. Text/graphics separation and recognition in raster-scanned color cartographic maps. In *Graphics Recognition*, J. Lladós and Y.-B. Kwon, Eds., Lecture Notes in Computer Science, Vol. 3088. Springer, 63–74.

T. Watanabe and R. Zhang. 1997. Recognition of character strings from color urban map images on the basis of validation mechanism. In *Proceedings of the 4th International Conference on Document Analysis and Recognition*. 805–808.

W. Wigand. 1988. Kartographische mustererkennung. *Kartographische Nachrichten* 38, 113–120.

S. Wise. 2002. Capturing raster data from scanned thematic maps using desktop graphics software. *Transactions in GIS* 6, 3, 327–337.

J. Wu, H. Yan, and A. N. Chalmers. 1994. Color image segmentation using fuzzy clustering and supervised learning, *Journal of Electronic Imaging* 3, 4, 397–403.

D. Xin, X. Zhou, and H. Zheng. 2006. Contour line extraction from paper-based topographic maps. *Journal of Information and Computing Science* 1, 5, 275–283.

H. Yamada, K. Yamamoto, and K. Hosokawa. 1993. Directional mathematical morphology and reformalized Hough transformation for the analysis of topographic maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15, 4, 380–387.

K. Yamamoto, H. Yamada, and S. Muraki. 1993. Symbol recognition and surface reconstruction from topographic map by parallel method. In *Proceedings of the 2nd International Conference on Document Analysis and Recognition*. 914–917.

P.-Y. Yin and Y.-B. Huang. 2001. Automating data extraction and identification on Chinese road maps. *Optical Engineering* 40, 5, 663–673.

H.-L. Zheng, X.-Z. Zhou, and J.-Y. Wang. 2003. Research and implementation of automatic color segmentation algorithm for scanned color maps. *Journal of Computer-Aided Design and Computer Graphics* 15, 1, 29–33.

D. X. Zhong. 2002. Extraction of embedded and/or line-touching character-like objects. *Pattern Recognition* 35, 11, 2453–2466.

Received February 2013; revised October 2013; accepted November 2013