J. Vis. Commun. Image R. 25 (2014) 1262–1274

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Accurate segmentation of land regions in historical cadastral maps

Nam Wook Kim [a], Jeongjin Lee [b,*], Hyungmin Lee [c], Jinwook Seo [c]

[a] Mobile Communication Research Center, LG Electronics, 219-24 Gasan-dong, Geumcheon-gu, Seoul 153-801, Republic of Korea
[b] School of Computer Science & Engineering, Soongsil University, 369 Sangdo-Ro, Dongjak-Gu, Seoul 156-743, Republic of Korea
[c] School of Computer Science and Engineering, Seoul National University, 599 Kwanak-ro, Kwanak-gu, Seoul 151-742, Republic of Korea

ARTICLE INFO

Article history:
Received 17 June 2013
Accepted 31 December 2013
Available online 11 January 2014

Keywords:
Cadastral map
Historical geographical information system
Line reconstruction
Line extraction
Polygonal approximation
Land segmentation
Character recognition
Grid removal

ABSTRACT

Historical cadastral maps are valuable sources for historians to study social and economic background of changes in land uses or ownerships. In order to conduct large-scale historical research, it is essential to digitize the cadastral maps. As being established in antiquity, however, they suffer from significant noise artifacts attributed to hand-drawn cartography. In this paper, we propose a novel method of extracting land regions automatically in historical cadastral maps. First, we remove grid reference lines based on the density of the black pixel with the help of the jittering. Then, we remove land owner labels by considering morphological and geometrical characteristics of thinned image. We subsequently reconstruct land boundaries. Finally, the land regions of a user's interest are modeled by their polygonal approximations. Our segmentation results were compared with manually segmented results and showed that the proposed method extracted the land regions accurately for assisting cadastral mapping in historical research.

1. Introduction

From the past to the present, land tenure and parcels have been crucial information for land administration such as land valuation and taxation [1]. To maintain the property records of a country, many nations adopted a cadastre survey, which involves the documentation of land registration such as location, area and ownership [2,3]. Two famous historical examples are the Domesday book from early England and the Napoleonian cadastre from 19th century France. Both of them laid the historical foundation for modern cadastral systems [4]. Historians use those records to study the evolving histories of land parcellation in conjunction with social and economic aspects of changes in land uses or ownerships [5].

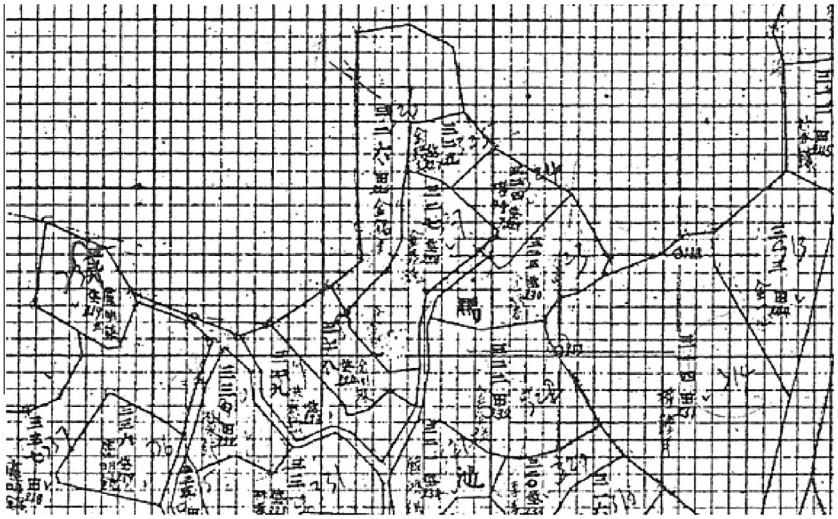
In Korea, the Kyujanggak Institute for Korean Studies (KIKS) preserves a significant number of cadastres from the 17th to 19th century, which cover major cities and suburban areas in the Joseon dynasty of Korean history [6]. A cadastral map has geographical boundaries of land ownership, while a textual cadastre is a tabular data including survey direction, neighborhood, area and owner (Fig. 1). Unfortunately, since they were recorded independently, the integrated cadastral research has been difficult. The researchers in the KIKS are currently working on constructing a mapping between the textual cadastre and cadastral map (i.e., cadastral mapping), which were recorded for same areas but at different times. Through this mapping, they expect to understand the temporal and spatial changes of the land ownership, development status, and residential areas in an integrative manner.

A typical task involved in the cadastral mapping is to use sticky notes and highlighters to mark the mapping from a land owner's name in the textual cadastre to the corresponding land region in the cadastral map [7]. Since this task is done on a physical copy of each cadastral map, it is difficult to undo if the mapping result turns out to be wrong. Also, such manual work is cumbersome for editing and searching the existing mappings. And the large quantity of cadastres makes maintenance tasks laborious.

To facilitate the cadastral mapping task, it is desirable to digitize the historical cadastral maps by allowing land regions to be searchable and manageable. Accordingly, there have been many works in building historical geographical information system (GIS) to assist cadastral mapping. A notable example is the Great Britain historical GIS [8]. It holds the changing boundaries of administrative units with historical statistics recorded from 1840 to 1970. Its GIS database is structured in a way that the land boundaries, if exist, are provided for specified dates. It was constructed by combining geographical maps with textual sources that provided specific dates for boundary changes. Another example is the China historical GIS which covers 2000 years of dynastic

* Corresponding author. Fax: +82 2 886 7589.
E-mail addresses: namw.kim@samsung.com (N.W. Kim), leejeongjin@ssu.ac.kr (J. Lee), hmlee@hcil.snu.ac.kr (H. Lee), jwseo@hcil.snu.ac.kr (J. Seo).

http://dx.doi.org/10.1016/j.jvcir.2014.01.001
1047-3203/



Global Index	Survey Direction	4-Neighbor Information				Area	Owner
		East	West	North	South		
212	West	JRW	LBK	Road	Mountain	783	KMB
213	West	KMB	LSW	Road	LKS	1445	LBK
214	West	LBK	Creek	Road	LKS	4080	LSW
215	North	Road	KSS	LSW	LYS	1924	LKS
216	West	LKS	Creek	LSW	LYS	2940	KSS

Fig. 1. A pre-modern cadastre in KIKS – cadastral map (top) and textual cadastre (bottom).

history in China [9]. Because of the poor accuracy in the records of many administrative regions, it did not attempt to reconstruct the exact boundaries. Instead, it used the locations of administrative units and their relative positions to approximate their boundaries. By integrating statistical data and geospatial data collected at different times (e.g., textual cadastre and cadastral map) into a single computer system, this system expedites historical research such as geo-referenced demographic study [10]. However, building those systems have never been easy particularly with historical cadastral data. It involves manual vectorization of spatial data into points, lines and polygons, which is a highly time-consuming and costly process [5]. To alleviate this problem, we need a more efficient way to perform the vectorization of geographical information.

This paper proposes a segmentation method that extracts the land regions accurately in historical cadastral maps. We use the cadastral maps from the KIKS (Fig. 1), which are currently being used by the Korean historians. As being established in pre-modern era, the maps suffer from significant noise artifacts attributed to hand-drawn cartography. They have not only compact grid lines and label characters, which are considered noise, but also eroded region boundaries. We designed a staged segmentation pipeline by devising a series of image processing techniques. We first eliminate noises in a scanned map image by removing grid reference lines based on the density of the black pixel with the help of the jittering. Then, we remove land owner labels by considering morphological and geometrical characteristics of thinned image. Then, we subsequently reconstruct broken land boundaries which are originated from both the eroded map and noise reduction phase. Finally, we extract the land regions of user's interest by generating their polygonal approximations.

The remainder of this paper is organized as follows. The next section describes the proposed method of automatically extracting the land region. Section 3 presents the experimental results. Section 4 describes one of applications for assisting historical research using the proposed method, followed by conclusion and future work in Section 5.

2. Related work

In the traditional practice of cartography, maps such as topographic or cadastral maps were hand-drawn on papers [11]. Nowadays, due to the scalability and efficiency, geographical information for urban planning or resource management is processed through computer systems [12]. With the recent proliferation of GIS applications, there has been an increasing need for converting existing analog maps to vector forms [13]. The vector data has many advantages over the raster data by encoding the topological structure of a map only in the form of points and lines. In addition, it takes less storage. It is not limited by spatial resolution, and is easier to manage and update [14].

The digitization is often performed manually either through a digitizing tablet with a paper map on its surface or an on-screen digitization using a scanned map [15]. Since the manual digitization is time consuming with intrinsic human errors, more advanced semi-automated or fully-automated procedures of the extraction of cartographic information from maps have been proposed [16]. Unfortunately, a fully-automatic vectorization is still a challenging task as types of maps vary considerably and human verification is almost always necessary [17]. In particular, it is considerably difficult to accurately vectorize historical maps because of its poor graphical quality caused by scanning or image compression processes, as well as the aging of the archived paper material, which often causes false coloring, blurring or bleaching problems [16,17].

A typical automatic vectorization procedure involves (a) digitization of paper documents using a scanner, (b) filtering, (c) thres-

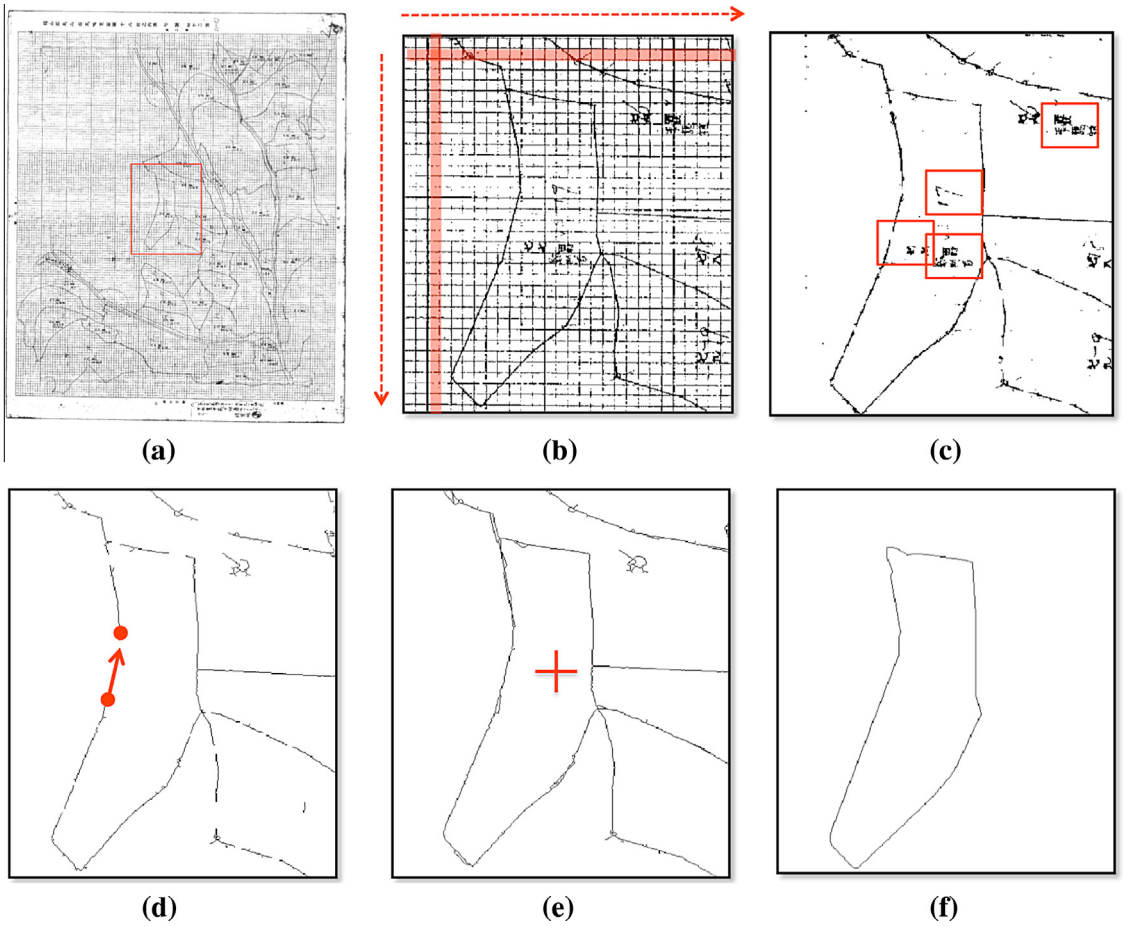


Fig. 2. Pipeline of cadastral map segmentation: (a) original map image, (b) removal of grid lines, (c) removal of labels and pixel fragments, (d) reconstruction of boundaries, (e) selection of a region of interest, (f) generation of a polygon approximation.

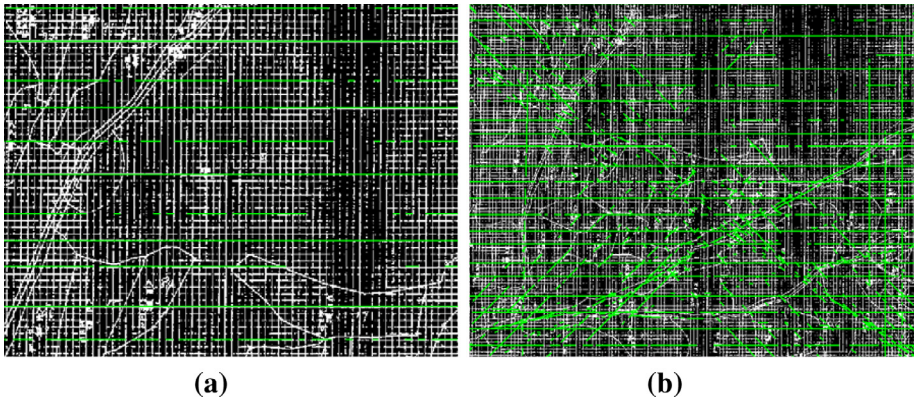


Fig. 3. Hough transform being applied to a cadastral map image: (a) HT with high peak threshold, (b) HT with low peak threshold.

holding, (d) thinning and pruning the binary image, and (e) raster to vector conversion [18]. Basic vectorization methods for raster line images can be roughly classified into three main categories: Hough transform (HT)-based, thinning-based, and contour-based methods [19]. While HT-based method is the fastest, the thinning-based method generates the best quality for the preservation of the line geometry and is comparatively faster in comparison to other methods [14]. Although the thinning-based method does not preserve the line-width, its advantage of maintaining the topological structure, including connectivity, adjacency, and relative position, is more valuable when vectorizing the spatial data of cartographic maps [12].

Lam et al. [20] gave a comprehensive survey of thinning algorithms, while non-thinning algorithms were reviewed in [21]. After the skeleton of an image is produced from thinning, a polygonization is performed on the skeleton points to approximate the detected lines and eliminate redundant points [22]. Postprocessing is often necessary to handle problems including broken graphics to convert low-level vectors into fine graphic objects [23]. While a crude vectorization such as skeletonization methods focuses on

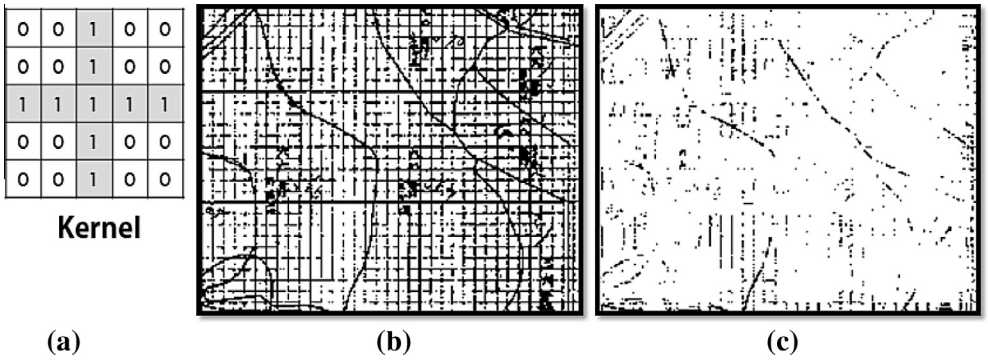


Fig. 4. Pattern matching approach by filtering with a kernel: (a) a filtering kernel, (b) original image, (c) filtered image.

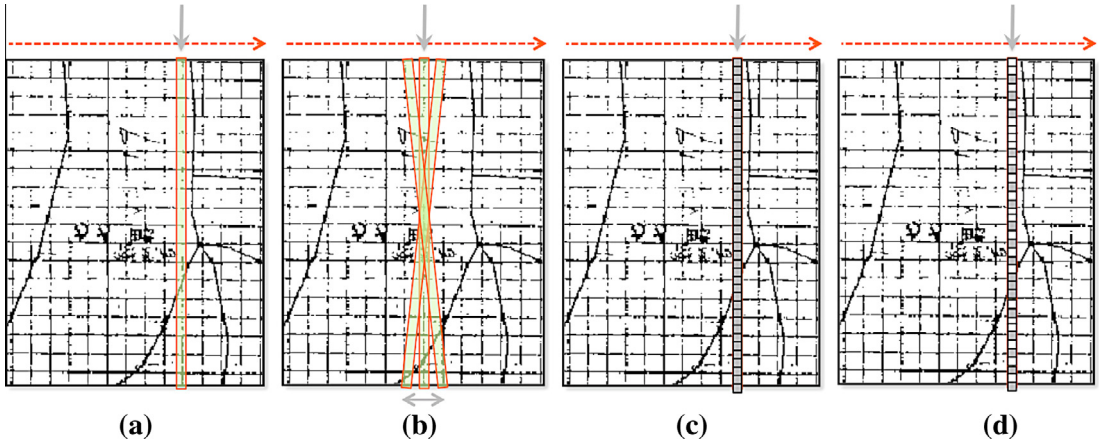


Fig. 5. Removal of a vertical grid line: (a) construction of a scan line, (b) jittering of the scan line, (c) recognition of a grid line based on the density of black pixels, (d) filtering non-grid pixels by examining neighboring pixels.

the shape preservation, more advanced researches have been devoted to improve and refine the quality of vectors [24–28].

While these previous approaches laid the groundwork for the recognition of maps, these methods were far from providing optimal solutions. Most of these algorithms employed vectorization models that were suited for more generic drawings including mechanical, electronic, and construction drawings and did not take the complex characteristics of map into account [13]. To overcome these limitations, segmentation and vectorization of maps have been studied. However, no analytic solution has been proposed yet and most existing methods employed ad hoc rules based on heuristics [29]. It is widely accepted that fully automatic solution is not achievable and customized methods using contextual knowledge can lead to the significant improvement in performance [13,30–32].

Many automatic vectorization methods designed for processing maps have been proposed. Wu et al. extracted contour lines from topographic map-based cartography and graphics knowledge such as coordinates and symbol system [32]. Janssen et al. exploited the knowledge of cartographic rules to improve and correct the result of vectorization [33]. Similarly, Lee et al. constructed a knowledge base, where cartographic features are contained based on different types of maps, to identify the characteristics of the input map image and apply various image-processing operations to vectorize it [12]. Frischknecht et al. used a knowledge-based template matching to extract areal objects from the scanned official Swiss topographic map [34]. There were other approaches that attempted to separate the map into constituent layers and recognize the features in different layers on the basis of symbol-specific geometrical and morphological attributes [35–37].

In most case, automatic solutions usually work well for maps with high-quality images. For maps with low-quality images, most prevalent and practical solutions are interactive and semi-automatic methods. Most common way to interactively vectorize linear features is the tracking of a line from a user-specified point and the use of the additional manual interventions in case where the automatic tracking fails [38,15]. Bucha et al. supported snapping seed points and tracking area objects [39]. Other techniques involve

Table 1
Pseudocode for the removal of grid lines.

	Step 1. Removing grids (vertical grid)
1:	**procedure** RemovGrid (bw, grid_width, thresh, jitter_range)
2:	FOR each scan line
3:	FOR dt = - jitter_range to jitter_range
4:	jittered = create_line(*grid_width*, dt)
5:	C(dt) = collect_intensity (jittered)
6:	ENDFOR
7:	[max, index] = find_max(C)
8:	IF max > *thresh*
9:	jittered = find_line(index)
10:	FOR each pixel(p) in jittered
	IF neighbor(p) = white
	p = white
	ENDIF
	ENDFOR
11:	ENDIF
12:	ENDFOR
13:	RETURN bw

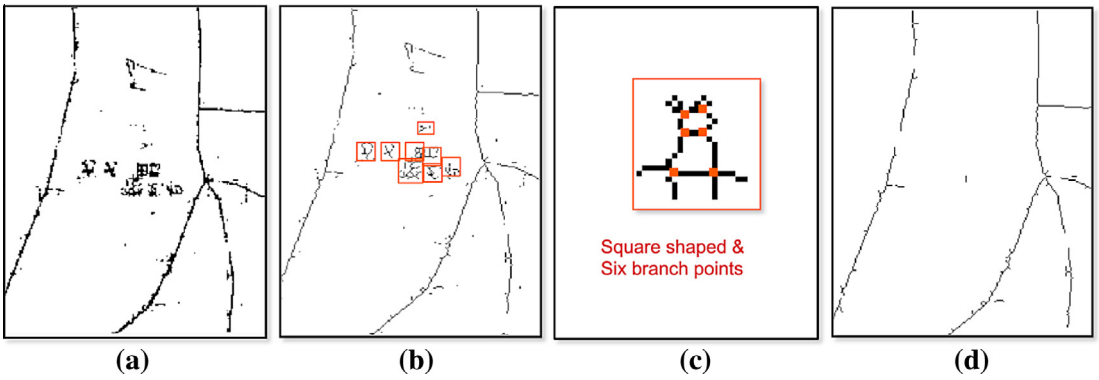


Fig. 6. Removal of characters: (a) image after grid removal, (b) thinned image with connected components, (c) examination of size, aspect ratio and branch point, (d) removal of detected characters.

Table 2
Pseudocode for removal of characters.

Step 2. Removing letters	
1:	**procedure** RemoveLetters *(bw, brch_pts, box_size, ratio)*
2:	ccs = find_connected_components (*bw*)
3:	FOR each connected component (cc)
4:	box = find_bounding_box(cc)
5:	[left, right] = get_side_length(box)
6:	IF (left > *ratio*·right) or (right > *ratio*·left)
7:	continue
8:	ENDIF
9:	IF (left × right) > *box_size*
10:	continue
11:	ENDIF
12:	num_brch_pts = find_branch_points (cc)
13:	IF (num_brch_pts ⩾ *brch_pts*) then
14:	erase_connected_component(bw, cc)
15:	ENDIF
16:	ENDFOR
17:	RETURN *bw*

Table 3
Pseudocode for the reconstruction of land boundaries.

Step 3. Reconstructing land boundaries	
1:	**procedure** CloseHoles *(bw, ray_len, lookup)*
2:	end_pts = find_end_points(*bw*)
3:	FOR each end point (ep) in end_pts
4:	FOR each prev pixel (pp) from ep to *lookup*
5:	dir += calc_dir_vec(pp, ep)
6:	ENDFOR
7:	search = construct_search_space(dir, *ray_len*)
8:	FOR each end point (p) in search
9:	dist = distance(ep, p)
10:	ENDFOR
11:	cp = find_closest_point(dist)
12:	connect(*bw*, ep, cp)
13:	ENDFOR
14:	RETURN *bw*

users in editing and cleaning unwanted details after segmenting the map and before the vectorization [16,17,30,31].

There are also many commercial software solutions such as MapGIS, R2V, VPStudio, and RxAutoImage. They still suffer from jagged and discontinuous dithering. And they also require that noises have to be removed before the vectorization. Lacroix presented the analysis of raster-to-vector softwares and proposed an improvement strategy in consideration to a map segmentation problem [40]. Dharmaraj also provided a comprehensive review on commercial vectorization softwares [14].

In this paper, we focused on extracting polygonal land regions enclosed by boundary lines. Unfortunately, previous methods cannot be directly applied to our problem. First, they rarely consider preprocessing by assuming high-quality input images. In addition, our datasets are completely different from those used in the previous methods. For example, for contour reconstruction, most of them focused on topographic maps where contour lines never intersect each other and form closed loops. Inspired by previous works and following the guideline that recommends incorporating contextual knowledge in a document [22], we developed a customized vectorization pipeline for the cadastral maps of the Joseon dynasty in Korea.

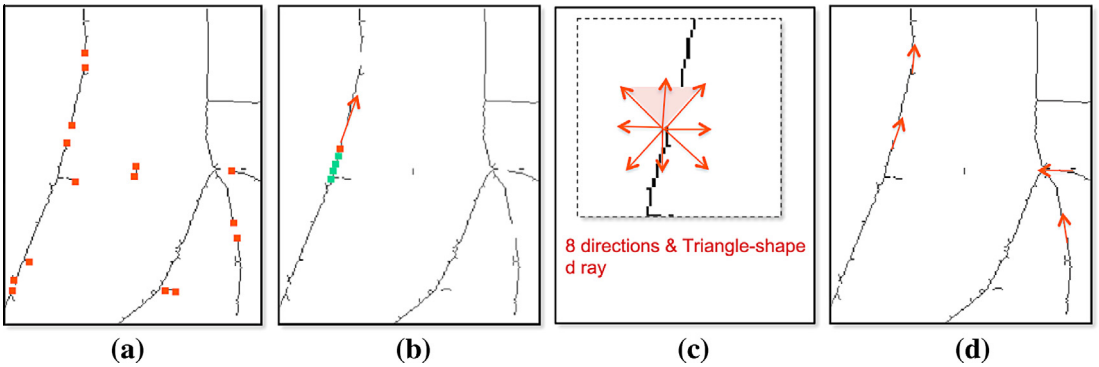


Fig. 7. Reconstruction of land boundaries: (a) detection of end points, (b) determination of the ray direction by considering previous pixels, (c) defining a search area among eight possible directions, (d) connection to the nearest end point; it is possible that end points can be connected to undesirable counterparts.

Table 4
Pseudocode for the generation of polygons.

Step 4. Generating polygons	
1:	**procedure** GeneratePolygon *(bw)*
2:	[x, y, num_pts] = mouse_input (*bw*)
3:	polygons = []
4:	regions = []
5:	for i = 1:num_pts do
6:	region = seeded_region_growing(bw, x(i), y(i))
7:	region = bwmorph(region, 'close')
8:	polygon = min_peri_poly(region)
9:	polygons = polygons + polygon
10:	regions = regions + region
11:	end for
12:	RETURN (regions, polygons)

3. Method

The proposed segmentation method is conducted in four stages as shown in Fig. 2. Beforehand, we preprocess the cadastral maps by scanning and resampling (Fig. 2a). In the first stage, we remove grid lines by constructing scan lines and classifying them based on the density of black pixels (Fig. 2b). Second, the label characters, which describe the owner names of land regions, are removed based on their morphological and geometrical characteristics (Fig. 2c). Third, we reconstruct land boundaries by connecting end points of broken line segments (Fig. 2d). Finally, land regions are extracted into polygons using seeded region growing and minimum-perimeter polygon algorithm [41] (Fig. 2e and f).

3.1. Preprocessing

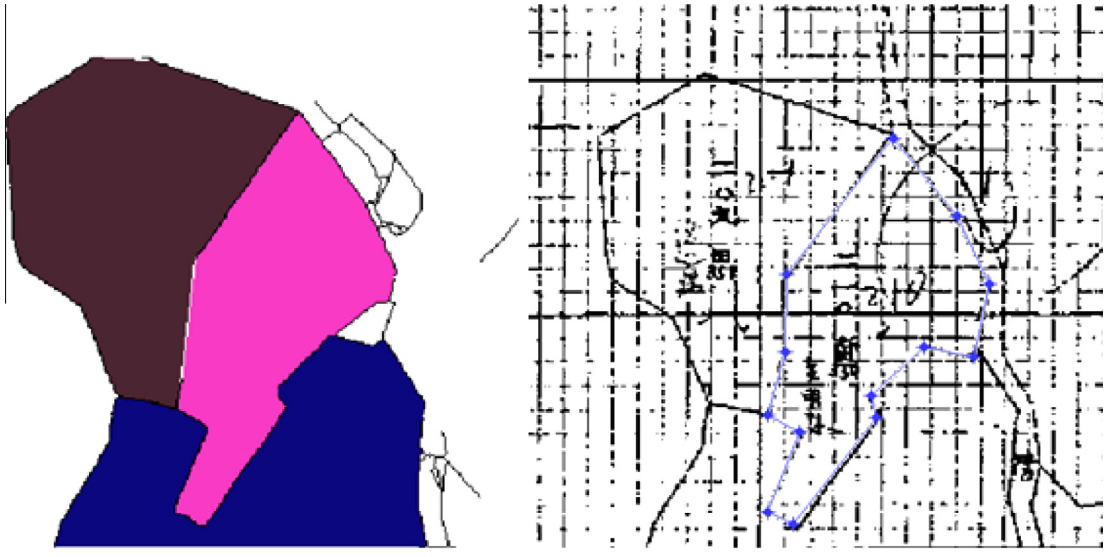
The cadastral maps are hand-drawn maps. To be fed into our segmentation pipeline, we first scan them into digital images. The original map is drawn on a rectangular box, but it is often not axis-aligned. If a scanned map is severely tilted, we manually align the image by rotating it such that vertical and horizontal grid lines on the map are close to be orthogonal. Such well-aligned image is preferred when a scan line is matched into the grid line. The image is initially scanned in a high-resolution grayscale format (e.g., 5204×6513), which requires substantial time to process. We lower the resolution to 2000 pixel width and its proportional height in order to shorten the processing time while maintaining accuracy. Finally, we convert the image into a binary image to take advantage of morphological operations, which include thinning, end point detection, and connected component labeling, in the subsequent character removal and boundary reconstruction steps. In the input image, black pixels correspond to the foreground and white pixels correspond to the background. Thus, it can be easily segmented using a global thresholding technique. We used Otsu's method to compute a global threshold for the binarization.

3.2. Removal of grid lines

In this step, we remove dense grid lines in cadastral map images. The grid is composed of a series of horizontal and vertical lines, and is independent of land regions. The grid lines are regularly placed showing the periodicity of occurrence. Although this grid might have been a good layout reference for the cartographer, it is regarded as a noise that makes it more difficult to extract the land regions automatically.

Quackenbush [42] provides a comprehensive survey of straight line detection techniques. Since the grid is not correlated with land regions, such content-independent algorithms could be employed for detecting and removing lines in an image. Unfortunately, they typically work best on clean images and are very sensitive to various types of noise. For example, Fig. 3 shows our initial trial to detect grid lines using Hough transform (HT). With the high peak threshold, HT missed a lot of grid lines (Fig. 3a). Also, by lowering the peak threshold, HT failed to detect correct grid lines by finding diagonal lines instead of connecting their intersections (Fig. 3b). Fig. 4 shows the grid detection result by a kernel based method [42]. The kernel was designed to find a grid pattern (Fig. 4a) but was also unable to capture the exact grid lines (Fig. 4c). In lieu, it severely degraded region boundaries making the map almost illegible.

We identified two main challenges for the grid removal: (1) grid lines are severely broken due to the low quality of the map images, (2) the grid lines are overlapped with text and land regions, making



(a) **(b)**

Fig. 8. Evaluation of the segmentation method: (a) automatically segmented land regions, (b) manual segmentation of a land region in the original map image.

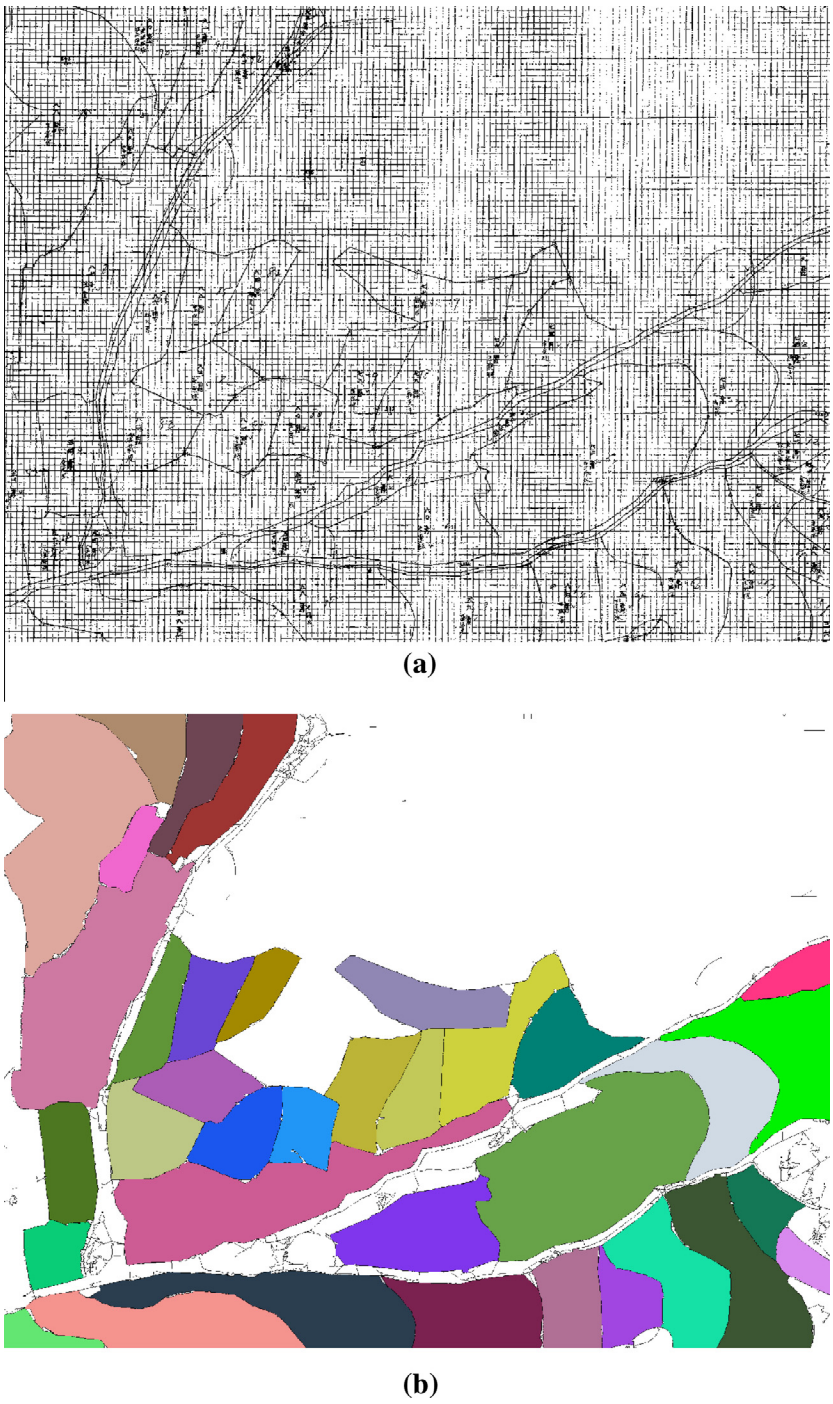


Fig. 9. The result of the segmentation method applied to the fourth dataset: (a) The cadastral map is scanned into a binary image, (b) land regions are extracted into polygons.

the separation difficult, and (3) the line width, gap, and slope are irregular. Therefore, it is very difficult to detect the grid lines without contextual information.

There have been previous researches that leverage the contextual information to remove background lines from binary document images [43,44]. Their algorithms are based on the observation that the lines are parallel and the gaps between any two neighboring lines are roughly equal. However, they have limitations as well in that the estimation error of the modeling parameters (e.g., line slope, line gap, and the position of the first line) is accumulated throughout the process. In addition, their input images are rather cleaner than ours and the background lines are printed whereas our lines were hand-drawn.

For the grid removal, we construct scan lines of varying widths horizontally as well as vertically to detect grid lines (Fig. 5a). For each scan line, we calculate the density of black pixels by summing up the number of black pixels over total number of pixels in the scan line. If this density is above a threshold, we regard the scan line as a grid line and clear the black pixels along the scan line (Fig. 5c). This density threshold is negatively correlated with the width of the scan line. The wider the line width is, the lower the density threshold is. Considering the image resolution, the optimal

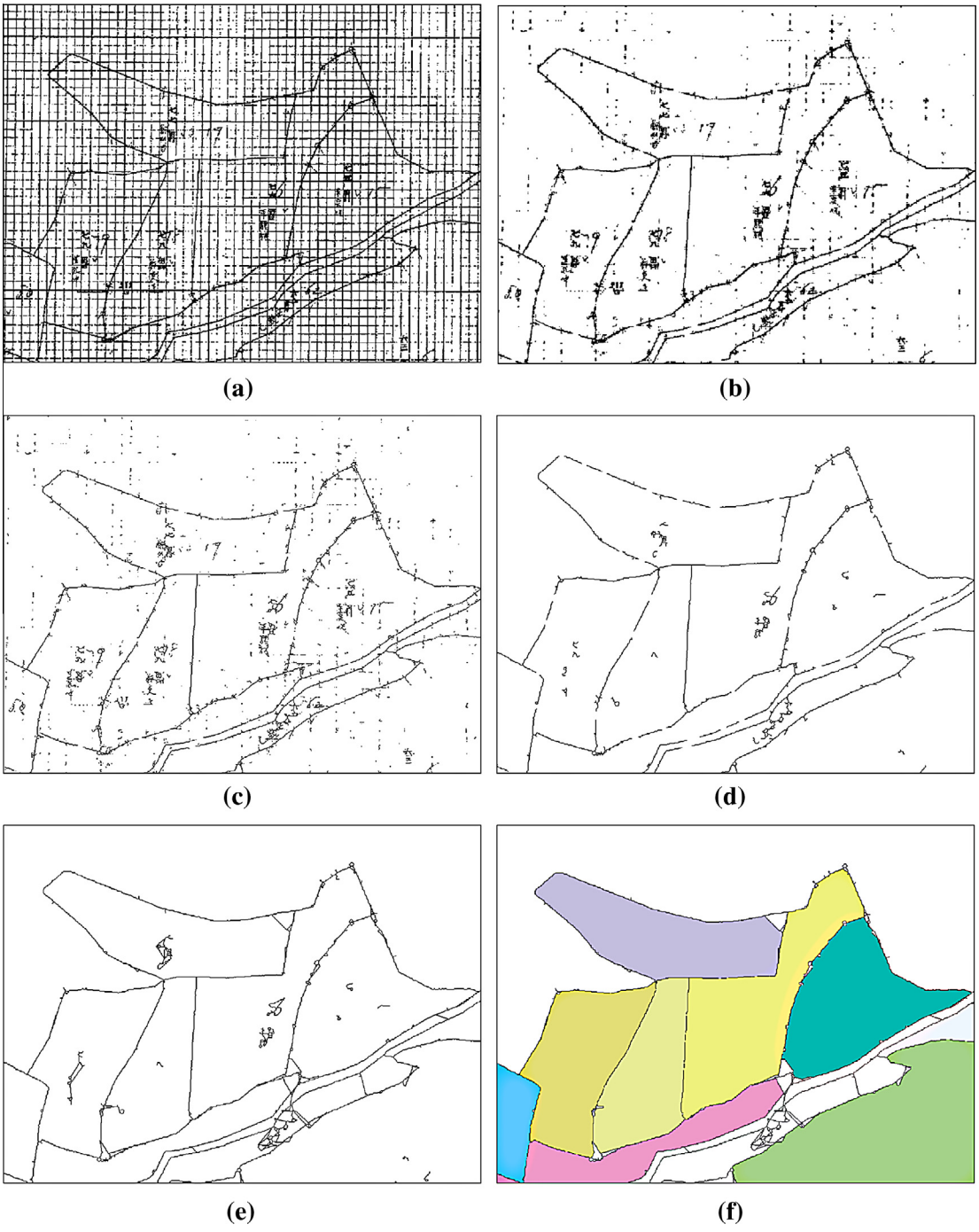


Fig. 10. The enlarged image of the sub-region result of the segmentation method applied to the fourth dataset (Fig. 9). It shows the resulting images at intermediate steps of the segmentation pipeline: (a) the cadastral map is scanned into a binary image, (b) grid lines are removed, (c) thinning is applied, (d) characters and pixel fragments are removed, (e) land boundaries are reconstructed, (f) land regions are extracted into polygons.

values of the density threshold and scan line width were empirically determined as 30% and 2 pixels, respectively. Since lines are severely broken, it is rare that the black pixels constitute more than about one third of the total pixels in a scan line.

However, due to the low quality of the original map and slanted scanned images, the grid lines are not always orthogonal to the sides of the image. Therefore, we perform the jittering of scan lines to compensate for the skewed grid lines (Fig. 5b). Although an increased jittering range could take care of more slated lines, it adversely affects the computational performance. Based on this observation, we chose to use 40 pixels for the jittering range by tilting the scan line pixel by pixel.

Since no prior knowledge on land regions is available for now, the grid removal algorithm removes the boundaries of land regions that overlap with the grid lines. To alleviate such unintentional removal, we examine the neighboring pixels before removing a pixel to determine if the pixel is a part of land boundaries (Fig. 5d). If black pixels exist around the pixel under consideration, we keep it intact. Pseudo code for the removal of grid lines is shown in Table 1.

Table 5
Accuracy assessment results of land region extraction.

Dataset	Measure	E_{fp}	E_{fn}	E_{area}	E_{sim}
1	AVG	1.159	1.560	−0.402	1.362
	STD	0.712	0.694	1.038	0.476
2	AVG	1.151	1.826	−0.675	1.494
	STD	0.813	0.896	1.198	0.613
3	AVG	2.417	2.334	0.083	2.372
	STD	1.325	0.881	1.636	0.764
4	AVG	1.825	2.631	−0.806	2.238
	STD	0.899	0.915	1.092	0.726
5	AVG	1.528	2.730	−1.092	2.127
	STD	0.362	1.234	1.168	0.816
6	AVG	1.245	2.676	−1.227	1.977
	STD	1.169	1.316	1.190	1.175
7	AVG	2.775	2.738	0.037	2.755
	STD	1.521	1.307	1.754	1.107
8	AVG	2.743	2.919	−0.022	2.799
	STD	1.948	1.463	1.981	1.462
9	AVG	1.900	2.811	−0.746	2.340
	STD	0.934	1.060	1.723	0.693
10	AVG	1.839	2.952	−1.113	2.407
	STD	1.516	1.112	2.208	0.737

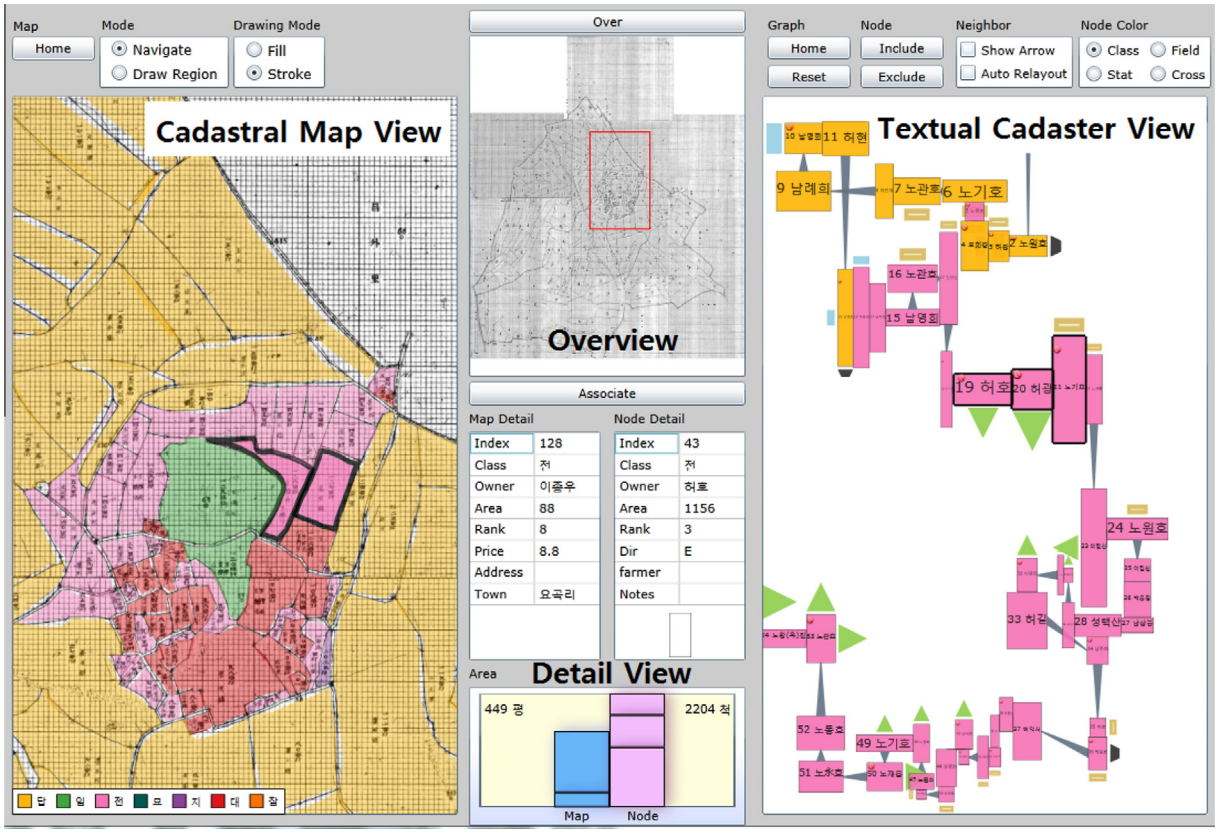


Fig. 11. Jigsawmap system being used by historians to construct a mapping between the textual cadastre and the cadastral map.

3.3. Removal of characters

Next, we remove label characters and pixel fragments. The fragments are salt-n-pepper noises caused by the grid removal process as well as the poor quality of the original map. The labels are Chinese characters written inside each land region. Although the labels were hand-written, the size of characters tends to be regular. They often touch land boundaries and even themselves, particularly for small land regions. The labels give important clues about the identity of owners of land regions, but they are obstacles in extracting land regions as is the same case with grid lines.

Handwritten character recognition, which could be used for removing the labels, has long been researched. However, the simulation of human reading is still a challenging problem [45–47].

The recognition rates of most techniques are sensitive to the age and quality of input documents. Thus, they typically require preprocessing steps to eliminate noises. Although we managed to remove grid lines in the cadastral maps, the remaining labels are even difficult for human to read. They were degraded since the inks were often spread. Sometimes, they became broken by the errors of the grid removal process. All these adverse conditions make them unfavorable for the conventional recognition techniques. Fortunately, we do not actually have to recognize the characters, but merely need to detect them for removal.

Many methods have been proposed to address the problem of extracting text from graphical documents, and can be divided into three main categories: morphological analysis, connected component analysis, and multi-resolution analysis [48]. Our work is based on the connected component analysis and largely inspired by the previous work by Fletcher and Kasturi [46] and a follow-up work by Tombre et al. [48], which used decision rules on area and dimensional ratio of connected components to separate text from graphics. To remove the characters, we use geometrical and morphological characteristics, which include the size, aspect-ratio, and branch points, without using any semantic information. In order to extract such features, we first apply 2D thinning [49] to the result image after the grid removal for the detection of branch points. We then find connected components [50] of the characters and calculate their bounding boxes (i.e., top-left and bottom-right points). For each connected component, we examine the size and aspect ratio of its bounding box and ignore it if they are not within the defined threshold. To be recognized as a character on the map, the aspect-ratio has to be close to a square (Fig. 6b). To deal with the connected characters, however, we increase the aspect-ratio threshold to accept those whose width is at most two times larger than the height. Finally, we eliminate remaining components whose number of branch points is above a chosen threshold (Fig. 6c). Since the Chinese character has many curves crossing each other, it could have more branch points compared to other elements in the image. In addition to characters, pixel fragments are removed based on the number of pixels in each connected component; if it is less than 15 pixels, the fragment is considered as a kind of salt-n-pepper noise hence removed from the image.

We determine the threshold values based on the manual inspection of representative sample images. The optimal parameter values may change depending on the selection of image resolution. For the removal of characters, we used the characteristics of a letter as having at least three branch points, above 40×40 size and at most 1:2 aspect ratio for 2000 pixel wide image. Pseudo code for the removal of characters is shown in Table 2.

3.4. Reconstruction of land boundaries

Next, we reconstruct land boundaries by connecting broken line segments. Such fragmentation was generated due to not only the noise in the original map but also the removal of overlapping pixels with grid lines and characters. It is not unusual that cartographers make annotations such as land names, symbols or grids on historical maps. They write this information on top of the map or often intentionally erase a part of it to make such notes. Such supplemental information generates broken lines when being removed in the vectorization process. Also, the noise in the original or scanned map limits the efficiency of the vectorization by producing gaps. It is necessary to restore the broken boundaries into closed loops in order to apply the final segmentation with a flood-fill operation in the next step.

Methods for reconstructing disconnected lines can be categorized into image-based approach and geometric-based approach [11]. In geometric-based approach, the gap filling is modeled as the more general problem of curve reconstruction [18]. Salvatore and Guitton use a Delaunay triangulation, where the Delaunay edges satisfying the topology of the contour lines are filtered, to vectorize the thinned binary image [18]. Pouderoux and Spinello proposed a parameterless reconstruction scheme based on the gradient orientation field of the available contour lines [51]. Other researchers used A* algorithm to find a shortest path that was in turn used to close the gaps [52,53].

Image-based approaches are mostly based on perceptual principles to connect two disconnected segments with two primary conditions: proximity and continuity [18]. Arrighi and Soille found the extremities of contour lines and used a combination of a distance and direction criteria for reconnecting broken lines [54]. This approach was used in [11]. Eikvil et al. used a line-tracing algorithm to reconstruct contour lines. When gaps occur, they assume there is only one possible continuation and cross the gaps by searching from the point at the end of the line within a sector around the current direction [55].

Excessive grids in our datasets produce small branches along the land boundaries, making the geometric-based approach not suitable for our problem. Thus, we take an image-based approach similar to [55]. First, we find end points of all connected components in the result image of previous steps (Fig. 7a) using the hit-and-miss transform [56]. For each end point, we shoot a ray whose direction is most likely to find the neighboring boundary fragment (Fig. 7b). To determine the ray direction, we look up previous pixels of the selected end point. We calculate the direction vector from each look–up pixel to the end point under consideration. The average direction is then used as a ray direction. We restrict the direction to 8-connectivity (Fig. 7c). Since it is rarely possible for this single ray to hit the counterpart end point of the neighboring boundary fragment, we parameterize the thickness of the ray to widen the search area. The shape of the search area is a right triangle where the source vertex of the ray is located at the end point. Within the search space, we find candidate end-points and simply connect the nearest one to the end point (Fig. 7d). Pseudo code for the reconstruction of land boundaries is shown in Table 3.

3.5. Generation of polygons

The final step is to extract land regions from the reconstructed image. Our method engages a user to select a region of interest and then uses a seeded region growing (i.e., flood-fill) to derive a pixel set of the selected region. It is possible that fragments survived previous steps could result in holes in this step. To remove such holes, we perform a morphological closing operation. We then use a minimum-perimeter polygon algorithm [41] to retrieve an approximate polygonal boundary of the region. At the end, the vertices of the polygon are saved in a csv file. We repeat this process until all the interested regions are extracted or the user is no longer able to provide a promising seed that leads to a segmented region. Pseudo code for the generation of polygons is shown in Table 4.

4. Experimental results

We tested our segmentation method on an Intel i7 laptop system with 1.73 GHz and 4 GB of memory. We prepared ten pre-modern cadastral maps of Mamyeong region around 18th century, all of which were from KIKS. The maps were scanned using Epson Expression 1680 scanner with 600 dpi resolution and manually axis-aligned as accurately as possible. Although our method is not limited to a certain image size, we adjusted the image width to 2000 pixel and let the height change proportionally (the pixel size was all 0.307×0.307 mm) to achieve the optimal performance without sacrificing overall accuracy. All datasets suffer from

poor quality and noise artifacts attributed to their age and hand-drawing.

The parameters for each intermediate algorithm were the same for all datasets. For the removal of grid lines, we used 2 pixel wide grids, 30% threshold, and 40 pixel jittering range. For the removal of letters and fragments, we used 3 minimum branch points, 40×40 box size, and 15 pixels for the fragment size. For the reconstruction of boundaries, we used 25 pixel-length ray and 5 pixels for look-up. These values were empirically determined based on visual inspection for ten training datasets, which are different from the ten test datasets. The parameters are more sensitive to image resolution which we fixed in the preprocessing stage than to the image content.

To validate the accuracy of our segmentation method, we developed an interface that enables us to manually segment land regions in the datasets (Fig. 8). For each dataset, we specified as many vertices as necessary along the boundary of each region. The vertices were then used to produce a polygon which closely approximates the land region. The manual segmentation was done on the original scanned map images. In this way, we obtained the manually segmented land region, which serves as the ground truth for the accuracy assessment of our segmentation method.

Fig. 9 shows the result of the segmentation method applied to the fourth dataset with the scanned cadastral map. And Fig. 10 shows the enlarged image of the sub-region outcome of each stage in the segmentation pipeline for the fourth dataset. It appears that our method significantly remove irrelevant noises and accurately identified land regions from the background. We first preprocessed the cadastral map into the resampled and binarized image as shown in Figs. 9a and 10a. It is apparent that the image is almost illegible because of the compact grid lines. Fig. 10b show that most grid lines are successfully removed, while leaving the characters and pixel fragments. They were subsequently removed in the next stages as shown in Fig. 10c and d. Fig. 10e shows the reconstructed boundaries. Finally, Figs. 9b and 10f show the segmented regions. Although it sometimes failed to identify small regions, most of the regions were successfully extracted. The labels touching boundaries are more prevalent in the small regions and are not removed by our method, causing errors in the reconstruction phase.

We evaluated the segmentation accuracy of our method by measuring the discrepancy between the manually segmented regions and automatically segmented ones. To assess the accuracy of our segmentation method, we employed four different evaluation metrics as follows:

$$E_{fp} = \frac{num\{A_{auto}\} - num\{A_{auto} \cap A_{manual}\}}{num\{A_{manual}\}} \times 100\%, \quad (1)$$

$$E_{fn} = \frac{num\{A_{manual}\} - num\{A_{auto} \cap A_{manual}\}}{num\{A_{manual}\}} \times 100\%, \quad (2)$$

$$E_{area} = \left(\frac{num\{A_{auto}\}}{num\{A_{manual}\}} - 1\right) \times 100\%, \quad (3)$$

$$E_{sim} = \left(1 - 2\left(\frac{num\{A_{auto} \cap A_{manual}\}}{num\{A_{auto}\} + num\{A_{manual}\}}\right)\right) \times 100\% \quad (4)$$

where A_{auto} is the set of pixels in the automatically segmented land region. The false positive error, E_{fp}, is the ratio of the set of pixels, in the automatically segmented region but not in the manually segmented region, to the set of pixels in the manually segmented region. The false negative error, E_{fn}, is the ratio of the set of pixels, in the manually segmented region but not in the automatically segmented region, to the set of pixels in the manually segmented region. E_{area} is the area measurement error and the similarity error, E_{sim}, is defined using the similarity index [57].

Table 5 summarizes the accuracy evaluation result of ten datasets. The average number of land regions for the datasets is 38 and the average and standard deviation of the errors are shown in the table. All the errors are less than 5%, showing that our method is indeed an accurate segmentation scheme for the cadastral maps. For most datasets, E_{fn} was higher than E_{fp}, and E_{area} had negative values. This means that A_{manual} is generally larger than A_{auto}. Based on the examination of the evaluation result images, we observed that the segmentation error occurs on the boundaries, which decrease the total area measurement for the automatically segmented region. During the preprocessing steps, land boundaries are often degraded. The resulting effect is the loss of accurate area measurement as the reconstruction step softens original boundaries. The granularity of the manual segmentation (i.e., the number of vertices) also affected the evaluation result.

5. Application

To demonstrate the practicality of our segmentation method, we developed an interactive visualization system, JigsawMap, which assisted the cadastral mapping task between a cadastral map and a textual cadastre [58].

5.1. User interface

The application provides two separate views: cadastral map view and textual cadastre view (Fig. 11). The user can analyze these views side by side and perform the mapping task. In the cadastral map view, land regions, which are automatically generated by our segmentation method, are visualized on top of the original map image. In addition, we provide a sketch-based tool for users to manually edit segmented land regions. On the right side, we visualize textual cadastres using a node-link graph layout. A node represents a land region and an edge indicates a survey direction. The layout is generated using survey direction, owner names and neighbor information. The land region on both view are color coded by the types of land use. The system also supports the overlay function, which lays the node-link diagram on the cadastral map. Using this function, a node will be placed on the corresponding land region.

5.2. Mapping interaction

To perform the cadastral mapping, a user selects a land region from the cadastral map view and another region (i.e., node) from the textual cadastre view. The user can exploit contextual information, such as owner name, size, shape, and neighborhood owners, to decide whether both indicate the same region. For example, if the selected land regions share the owner name, similar size and same neighborhood, it is more likely that they are the same region. Once being convinced, the user can perform 'associate' action to place a marker indicating that the matching has been done. The user will continue this process until all the regions are matched between the cadastral map and textual cadastre.

6. Conclusion

In this paper, we presented a novel segmentation method that combined a series of image processing algorithms to extract land regions automatically from historical cadastral maps. In the first stage, we remove grid lines by constructing scan lines and classifying them based on the density of black pixels. Second, the label characters, which describe the owner names of land regions, are removed based on their morphological and geometrical characteristics. Third, we reconstruct land boundaries by connecting end

points of broken line segments. Finally, land regions are extracted into polygons using seeded region growing and minimum-perimeter polygon algorithm.

Since most historical cadastres were generally hand-drawn and degraded, it has rarely been attempted to automate the vectorization of the maps that involves generating polygons from land regions. For large scale historical research, however, it is inevitable to digitize the maps, or construct a historical GIS, which typically involves the manual vectorization of geographical elements in the maps. Our method is designed to reduce significant time and effort that goes into such time-consuming, yet invaluable, vectorization process of the land regions. We contributed a series of image processing algorithms. All together tackled the segmentation problem that was otherwise difficult to handle using conventional noise reduction and reconstruction techniques.

The experimental results and interactive application of this study demonstrated that the proposed method accurately extracted the land regions and was useful for large scale historical research. We applied our method to ten sets of representative maps and compared the automatically segmented results with the manually segmented regions resulting in the average absolute area error of $0.62 \pm 0.61\%$. The method was integrated with the application and helped the historians easily identify the land regions.

Though it is encouraging, a number of limitations remain as well. First, our method cannot distinguish between grid lines and land boundary if they are completely overlapping each other. It is also unable to accurately segment labels that overlap with land boundaries; there are previous works that attempt to extract overlapping text from graphics [59,60], but their algorithms are based on printed and undistorted text. These edge cases in turn affect the result of reconstructing boundaries. To address this issue, we plan to explore additional features for characterizing the grid lines and text labels and also, instead of the current deterministic approach, employ probabilistic learning techniques such as [61] to enhance the quality of the noise removal. Another avenue for future work is to adopt a level set approach that produces an evolving contour fitting the boundary in place of connecting broken boundaries in pixel level. Finally, by focusing on segmenting land regions, we do not address other geographical elements such as creek and mountain. This remains as an interesting challenge for future research.

Acknowledgments

This work was partly supported by the IT R&D program of MSIP/KEIT [10044910, Development of Multi-modality Imaging and 3D Simulation-Based Integrative Diagnosis-Treatment Support Software System for Cardiovascular Diseases] and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2011-0030813).

References

[1] FIG, Statement on the cadastre, International Federation of Surveyors, Canberra, FIG Australian Bureau 1992–95 (1995) 22.
[2] J.L.G. Henssen, I.P. Williamson, Land registration, cadastre and its interaction a world perspective, in: Federation Internationale des Geometres, XIX International Congress, 1990.
[3] J. Henssen, Basic principles of the main cadastral systems in the world, in: Proceedings of the One Day Seminar held during the Annual Meeting of Commission 7, Cadastre and Rural Land Management of the International Federation of Surveyors (FIG), 1995.
[4] G. Larsson, Land registration and cadastral systems, Longman Scientific and Technical, New York, 1991.
[5] Z. Mou, Using cadastral maps in historical GIS research: the French Concession in Shanghai (1931–1941), Ann. GIS 18 (2) (2012) 147–156.
[6] Kyujanggak Institute for Korean Studies, May 2013, http://kyujanggak.snu.ac.kr/.
[7] H. Miyajima, in: Comparative analysis of Kwangmu yang'an and cadastral, research on the cadastral survey in Chosôn dynasty, Minumsa, 1997, pp. 199–248.
[8] I. Gregory, C. Bennett, V. Gilham, S. Humphrey, The great Britain historical GIS project: from maps to changing human geography, Cartogr. J. 39 (1) (2002) 37–49.
[9] M.L. Berman, Boundaries or networks in historical GIS: concepts of measuring space and administrative geography in Chinese history, Hist. Geogr. 33 (2005) 118–133.
[10] P. Ekamper, Using cadastral maps in historical demographic research: some examples from the Netherlands, Hist. Family 15 (1) (2010) 1–12.
[11] Y. Chen, R. Wang, J. Qian, Extracting contour lines from common-conditioned topographic maps, IEEE Trans. Geosci. Remote Sens. 44 (4) (2006) 1048–1057.
[12] K. Lee, S. Cho, Y. Choy, Automated vectorization of cartographic maps by a knowledge-based system, Eng. Appl. Artif. Intell. 13 (2) (2000) 165–178.
[13] S.A. Shereen, H.E. ElDeeb, D.M. Atiya, A new model for automatic raster-to-vector conversion, Int. J. Eng. Technol. 3 (3) (2011) 182–190.
[14] G. Dharmaraj, Algorithms for automatic vectorization of scanned maps, University of Calgary, Department of Geomatics Engineering, 2005.
[15] Y. Yang, X. An, L. Huang, Vectorization of linear features in scanned topographic maps using adaptive image segmentation and sequential line tracking, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XXXIX-B4 (2012) 103–108.
[16] Y.-Y. Chiang, L.S. Leyk, C.A. Knoblock, in: Efficient and robust graphics recognition from historical maps, Graphics Recognition. New Trends and Challenges, Springer, Berlin Heidelberg, 2013, pp. 25–35.
[17] Y.-Y. Chiang, L.S. Leyk, C.A. Knoblock, Integrating color image segmentation and user labeling for efficient and robust graphics recognition from historical maps, in: The Ninth IAPR International Workshop on Graphics Recognition, 2011.
[18] S. Salvatore, P. Guitton, Contour line recognition from scanned topographic maps, J. WSCG 12 (1–3) (2004).
[19] L. Wenyin, D. Dori, From raster to vectors: extracting visual information from line drawings, Pattern Anal. Appl. 2 (1) (1999) 10–21.
[20] L. Lam, S. Lee, C.Y. Suen, Thinning methodologies-a comprehensive survey, IEEE Trans. Pattern Anal. Mach. Intell. 14 (9) (1992) 869–885.
[21] L. Wenyin, D. Dori, in: A survey of non-thinning based vectorization methods, Advances in Pattern Recognition, Springer, Berlin Heidelberg, 1998, pp. 230–241.
[22] K. Tombre, C. Ah-Soony, P. Dosch, G. Masini, S. Tabbone, in: Stable and robust vectorization: How to make the right choices, Graphics Recognition Recent Advances, Springer, Berlin Heidelberg, 2000, pp. 3–18.
[23] J. Song, M. Cai, M.R. Lyu, S. Cai, Graphics recognition from binary images: one step or two steps, IEEE Int. Conf. Pattern Recognit. 3 (2002) 135–138.
[24] X. Hilaire, K. Tombre, Robust and accurate vectorization of line drawings, IEEE Trans. Pattern Anal. Mach. Intell. 28 (6) (2006) 890–904.
[25] D. Dori, W. Liu, Sparse pixel vectorization: an algorithm and its performance evaluation, IEEE Trans. Pattern Anal. Mach. Intell. 21 (3) (1999) 202–215.
[26] X. Hilaire, K. Tombre, in: Improving the accuracy of skeleton-based vectorization, Graphics Recognition Algorithms and Applications, Springer, Berlin Heidelberg, 2002, pp. 273–288.
[27] J. Song, F. Su, J. Chen, C. Tai, S. Cai, Line net global vectorization: an algorithm and its performance evaluation, IEEE Conf. Comput. Vision Pattern Recognit. 1 (2000) 383–388.
[28] O. Hori, S. Tanigawa, Raster-to-vector conversion by line fitting based on contours and skeletons, in: IEEE International Conference on Document Analysis and Recognition, 1993.
[29] T.C. Henderson, in: Segmentation and Vectorization, Analysis of Engineering Drawings and Raster Map Images, Springer, New York, 2014, pp. 17–31.
[30] B. Baily, M. Riley, P. Aucott, H. Southall, Extracting digital data from the first land utilisation survey of Great Britain-methods, issues and potential, Appl. Geogr. 31 (3) (2011) 959–968.
[31] B. Baily, The extraction of digital vector data from historic land use maps of Great Britain using image processing techniques, e-Perimetron 2 (4) (2007) 209–223.
[32] R.-Q. Wu, X.-R. Cheng, C.-J. Yang, Extracting contour lines from topographic maps based on cartography and graphics knowledge, J. Comput. Sci. Technol. 9 (2009).
[33] R.D. Janssen, R.P. Duin, A.M. Vossepoel, Evaluation method for an automatic map interpretation system for cadastral maps, in: IEEE International Conference on Document Analysis and Recognition, 1993.
[34] S. Frischknecht, E. Kanani, A. Carosio, A raster-based approach for the automatic interpretation of topographic maps, Photogramm. Eng. Remote Sens. 32 (1998) 523–530.
[35] S. Leyk, R. Boesch, Improving feature extraction of composite cartographic information in low-quality maps, Cartogr. Geogr. Inf. Sci. (2008).
[36] Y.-Y. Chiang, Harvesting geographic features from heterogeneous raster maps, University of Southern California, 2010.
[37] D.B. Dhar, B. Chanda, Extraction and recognition of geographical features from paper maps, Int. J. Doc. Anal. Recogn. (IJDAR). 8 (4) (2006) 232–245.
[38] L. Eikvil, K. Aas, H. Koren, Tools for interactive map conversion and vectorization, IEEE International Conference on Document Analysis and Recognition, vol. 2, 1995.
[39] V. Bucha, S. Ablameyko, T. Pridmore, Semi-automatic extraction and vectorisation of multicoloured cartographic objects, in: IEEE International Conference on Visual Information Engineering, 2005, pp.115–120.

[40] V. Lacroix, Raster-to-vector conversion: problems and tools towards a solution a map segmentation application, in: IEEE International Conference on Advances in Pattern Recognition, 2009.
[41] R.C. Gonzalez, R.E. Woods, S.L. Eddins, Digital Image Processing Using MATLAB, vol. 2, Gatesmark Publishing, Tennessee, 2009.
[42] L.J. Quackenbush, A review of techniques for extracting linear features from imagery, Photogramm. Eng. Remote Sens. 70 (12) (2004) 1383–1392.
[43] Y. Zheng, H. Li, D. Doermann, A model-based line detection algorithm in documents, in: Proceedings of Seventh International Conference on Document Analysis and Recognition, 2003, pp. 44–48.
[44] Y. Zheng, H. Li, D. Doermann, Background line detection with a stochastic model, in: IEEE Workshop on Computer Vision and Pattern Recognition, 2003.
[45] N. Arica, F.T. Yarman-Vural, An overview of character recognition focused on off-line handwriting, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 31 (2) (2001) 216–233.
[46] L.A. Fletcher, R. Kasturi, A robust algorithm for text string separation from mixed text/graphics images, IEEE Trans. Pattern Anal. Mach. Intell. 10 (6) (1988) 910–918.
[47] K. Tombre, S. Tabbone, L. Pélissier, B. Lamiroy, P. Dosch, in: Text/graphics separation revisited, Document Analysis Systems V, Springer, 2002, pp. 200–211.
[48] T.V. Hoang, S. Tabbone, Text extraction from graphical document images using sparse representation, Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, ACM, 2010.
[49] A.R. Widiarti, Comparing Hilditch Rosenfeld, Zhang-Suen, and Nagendraprasad-Wang-Gupta Thinning, World Acad. Sci. Eng. Technol. 78 (2011) 146–150.
[50] F. Chang, C.J. Chen, C.J. Lu, A linear-time component-labeling algorithm using contour tracing technique, Comput. Vision Image Understand. 93 (2) (2004) 206–220.
[51] J. Pouderoux, S. Spinello, Global contour lines reconstruction in topographic maps, in: IEEE International Conference on Document Analysis and Recognition, vol. 2, 2007, pp. 779–783.
[52] A. Khotanzad, E. Zink, Contour line and geographic feature extraction from USGS color topographical paper maps, IEEE Trans. Pattern Anal. Mach. Intell. 25 (1) (2003) 18–31.
[53] L.M. San, S.M. Yatim, N.A.M. Sheriff, N. Isrozaidi, Extracting contour lines from scanned topographic maps, in: IEEE International Conference on Computer Graphics, Imaging and Visualization, 2004.
[54] P. Arrighi, P. Soille, From scanned topographic maps to digital elevation models, Proc. Geovision 99 (1999) 1–4.
[55] E. Hancer, R. Samet, Advanced contour reconnection in scanned topographic maps, in: IEEE International Conference on Application of Information and Communication Technologies, 2011.
[56] R. Haralick, L. Shapiro, Computer and Robot Vision, vol. 1, Addison-Wesley Publishing Company, 1992, pp. 168–173.
[57] A.P. Zijdenbos, B.M. Dawant, R.A. Margolin, Morphometric analysis of white matter lesions in MR images: method and validation, IEEE Trans. Med. Imaging 13 (4) (1994) 716–724.
[58] H. Lee, S. Lee, N. Kim, J. Seo, JigsawMap: connecting the past to the future by mapping historical textual cadasters, in: Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems, 2012, pp. 463–472.
[59] R. Cao, C.L. Tan, in: Text/graphics separation in maps, Graphics Recognition Algorithms and Applications, Springer, Berlin Heidelberg, 2002, pp. 167–177.
[60] P.P. Roy, E. Vazquez, J. Lladós, R. Baldrich, U. Pal, in: A system to segment text and symbols from color maps, Graphics Recognition. Recent Advances and New Opportunities, Springer, 2008, pp. 245–256.
[61] M Agrawal, D. Doermann, Stroke-like pattern noise removal in binary document images, in: International Conference on Document Analysis and Recognition, 2011, pp. 17–21.