From Paper Map to Geospatial Vector Layer: Demystifying the Process

Peter Peller[1]

Abstract

With paper map use in decline, one of the strategies that libraries and archives can adopt to make the information contained within them more accessible and usable is to extract features of interest from their scanned raster maps and convert those to geospatial vector data. This process adds valuable unique data to library geospatial collections and enables those previously map-bound features to be used separately in geographic information systems (GIS) software for custom mapping and analysis. Advances in partially automating most of the process have made this a much more viable option for libraries and archives. Although there is no one-size-fits-all automated solution for all maps and map features, this paper provides a complete description of the entire process incorporating examples of the various techniques and software used in selected studies that would be applicable in the library and archive environment.

Keywords

Map features, vectorization, raster-to-vector conversion, geospatial vector layers, digitization, feature extraction

Introduction

With the paradigm shift towards digital mapping sources, the use of paper maps has significantly declined over the past fifteen years to the point that much of the valuable information in them is at risk of being forgotten or ignored. Many libraries and archives have responded to this challenge by digitizing[2] parts of their map collections – making the resulting raster[3] images more universally accessible through the Internet. Some have even gone a step further and have transformed those raster images – through a process called georeferencing – into geospatial raster images that are compatible with geographic information systems (GIS) software. The georeferenced maps in the David Rumsey Map Collection (Cartography Associates, 2017) are an excellent example of how historic geospatial raster images can be used in Google Earth or Google Maps as overlays. More recently, a few libraries and archives have taken the next step and have started to experiment with the extraction of specific features of interest from the geospatial raster imagery to create entirely new sources of geospatial vector[4] data that can be manipulated in innovative ways that were not previously possible.

In order to work with GIS software, the extracted features must be point, line or polygon vectors (geospatial vector data) with geographic coordinate information – representing real-world geographic features. These vectors are organized into single-type layers (ex/ all road segments) which are the fundamental units used by GIS software.These are normally created from air photos, satellite data, or from ground surveys. Historic maps allow us to travel backwards in time to reverse engineer the original data.

The conversion of raster images to vector data – otherwise known as vectorization – is analogous to OCR technology extracting words from scanned print documents. Without OCR it would be impossible to search for specific text in a scanned document or do any kind of textual analysis without manually going through the entire document. The same is true for scanned maps. Vectorization extracts pixels delineating physical and cultural features (e.g. contour lines, roads, soil zones) from a geospatial raster image and converts them into vector point, line and polygon layers. The resulting individual geospatial vector layers are searchable and useable for custom mapping and spatial analysis purposes in GIS.

Although it is possible to manually vectorize features from geospatial raster images through heads-up digitizing[5], the process is very tedious and labor-intensive, making it a suitable option for only simpler one-off maps. A great deal of research effort has gone into developing automated solutions, but a fully automated solution that works on all features in all maps still does not exist due to the huge variety and complexity of maps (see Figure 1): different features, symbols, colors (including textures), labels, and textual information – all of which can also overlap and intersect each other. Nevertheless, progress has been made, and it is now possible to semi-automate the map vectorization process. This development makes it a more viable option for libraries and archives considering map vectorization projects.

The goal of this paper is to provide a detailed review of all the steps that comprise the process of converting a map feature on a print map into a geospatial vector layer. This review incorporates examples of methods and procedures that would be most relevant in a library and archives setting. The examples are taken from nine selected studies which include all the published library-based ones (Arteaga, 2013; Godfrey and Eveleth, 2015; Marciano et al., 2013; Pearson et al., 2013; Bracke et al., 2008) and a few others that utilized commercial or open source solutions and were of a more applied nature (Brown, 2002; Jung, 2009; Southall, 2003; Whitfield, 2005). For the purposes of this review, two of the selected studies (Jung, 2009; Southall, 2003) have been split into two separate cases each: one study used two different methods and the other study extracted two different types of features. The eleven cases are summarized in the Appendix.

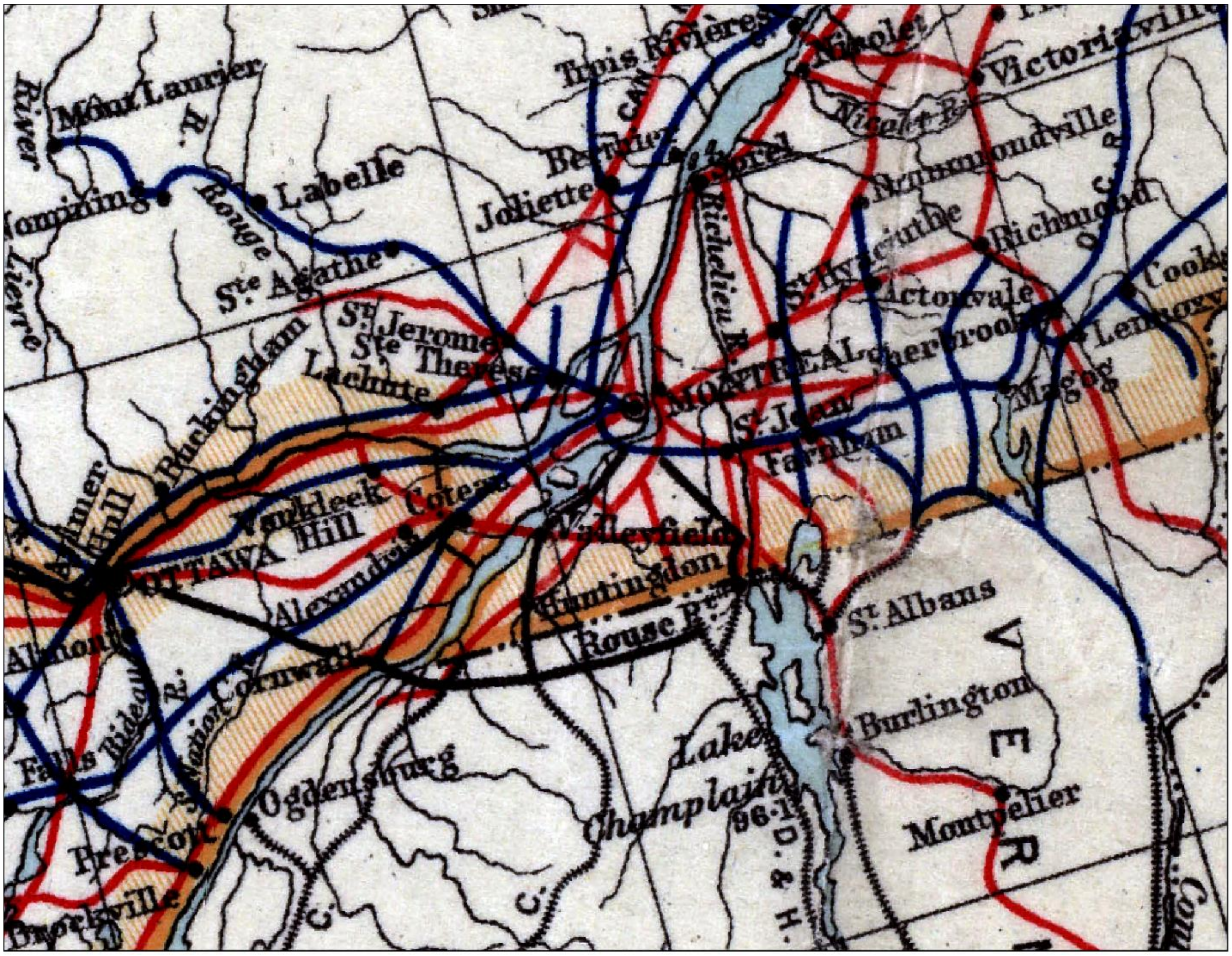


Figure 1. An example of map complexity: overlapping text, intersecting features, different features with the same color, grid line, symbols, textured areas and map damage. (Source: Map of the Dominion of Canada, 1929)

Print Map to Geospatial Vector Layer Process

The success of automated vectorization is dependent not only on the raster-to-vector conversion, but also on everything from the quality of the original print map to all the processes that enhance, isolate and edit the pixels of interest prior to this step, and the subsequent fine-tuning work on the extracted vector layer. The entire process can be broken down into the following steps:

1) Scanning

2) Georeferencing

3) Image Enhancement

4) Image Segmentation

5) Raster Editing

6) Raster to Vector Conversion (Vectorization)

7) Vector Editing.

1. Scanning

Usage, storage conditions and time can take their toll on the original paper map: tearing, staining, creases, fading, color loss and color bleeding. Although some of these defects can be mitigated in later steps, a poor quality map will nevertheless negatively impact the vectorization process. One option may be to borrow maps that are in better condition from other libraries (Bracke et al., 2008).

The paper map is scanned using either a digital camera system or a flatbed or roll scanner. Scanning resolution is an important factor. On a scanner the terms dots per inch (DPI) or pixels per inch (PPI) are often used interchangeably and basically refer to samples per inch. The studies that reported their scanning resolutions used between 200-600 DPI (Bracke et al., 2008; Brown, 2002; Pearson et al., 2013; Southall et al., 2003; Whitfield, 2005). The major finding from these studies is that – for vectorization purposes – higher resolutions are not always better (Pearson et al., 2013; Southall et al., 2003). It really depends on the size of the features of interest and – in the case of lines – their closeness to each other. Slightly higher resolutions prevent very close lines from fusing together. On the other hand, higher resolutions tend to capture the paper texture and ink spread which introduces noise and errors into the image. Pearson et al's (2013) research found that a 400 PPI resolution produced 1.37 times more gaps and bridges than the 330 PPI resolution of the same map. A bridge is a link between two lines that shouldn't be linked and a gap is a break in a line that should be continuous. The other consideration with higher resolutions is the larger file size and its implications for later processing. For most purposes, it seems that a 300 DPI scanner resolution is sufficient and preferable for vectorization purposes.

Other factors are bit depth, thresholding, and output file type. A 24 bit color is most commonly used with colored maps for vectorization purposes (Allord et al., 2014; Chiang et al., 2016). Bit depth also affects image file size: one study found that 8 bit color was sufficient and generated TIFF images that were 26Mb in size at a resolution of 200dpi (Southall et al., 2003); while another used 32 bit color at a resolution of 600dpi and generated a TIFF image that was 730Mb in size (Bracke et al., 2008). Thresholding on a scanner was employed by one United States Geological Survey (USGS) study to remove greenlines from their mylar maps (Whitfield, 2005). As recommended by the USGS (Allord et al., 2014), saving output in a lossless compression format, such as TIFF, ensures the retention of as much of the original data from the map as possible; this was the output file format used by the majority of the studies.

2. Georeferencing

In the studies examined, there was a bit of variation as to when georeferencing occurred. It doesn't really matter, but the one advantage of completing it before the segmentation and raster editing steps is that the georeferenced raster map image can then be overlaid on a base map. This enables one to see how well the segmentation worked and also helps to visualize whether the raster editing is being done correctly.

Georeferencing is the process of assigning the raster map its geographic coordinates and coordinate system; in other words, it associates a map image with its actual location in geographic space. GIS software like ArcGIS is normally used to do the georeferencing (Bracke et al., 2008; Godfrey and Eveleth, 2015; Pearson et al. 2013; Whitfield, 2005); however, the R2V vectorization software has the georeferencing capability built into it (Brown, 2002).

Basically the process involves using GIS software to fit the raster map to a geospatial layer – that does have a coordinate system – using control points. A control point is simply a point on the raster map image that matches a point on the geospatial layer such as a street intersection, boundary, or grid point. When applying control points it is advisable to alternate adding them at opposite sides of the map and distributing them throughout the map. How many control points are created will depend on the map; there will be a diminishing return after a certain number. The main thing is to keep an eye on each point's residual error and the total error which is calculated by taking the Root Mean Square (RMS) of all the residuals. The "residual error is the difference between where the point ended up as opposed to the actual location that was specified" (ESRI, 2016a). It is advisable to delete and redo any control points that have unacceptable errors. "When georeferencing, the goal is to attain a Root Mean Square (RMS) error that is less than or equal to the cell size of the raster file; this cell size represents the accuracy of the data" (ESRI, n.d.). The Bracke et al. study (2008), which used 330 control points, reported a RMS of approximately zero; however, it acknowledged that this may have been overkill when dealing with older maps and themes such as soil zones which don't actually have finite edges. For most purposes, it would probably be sufficient to use a similar number (16) to the Whitfield case (2005).

In order to save the georeferenced raster image with its new coordinate information it needs to be georectified. This georectification involves transforming the image (scale, skew, rotate, translate, stretch and warp) with a transformation equation. A first order polynomial transformation is sufficient for the vast majority of scanned maps (ESRI, n.d.). Once a raster map in TIFF format has been georeferenced and georectified it is usually saved as a GeoTIFF file.

Georeferencing can be a time-consuming process and a few major projects have actually used crowdsourcing for this. The New York Public Library (NYPL) has adapted a program called MapWarper for this purpose and the British Library has used GeoReferencer for their crowdsourced georeferencing. Both of these georeferencing tools are open source. (Fleet et al., 2012)

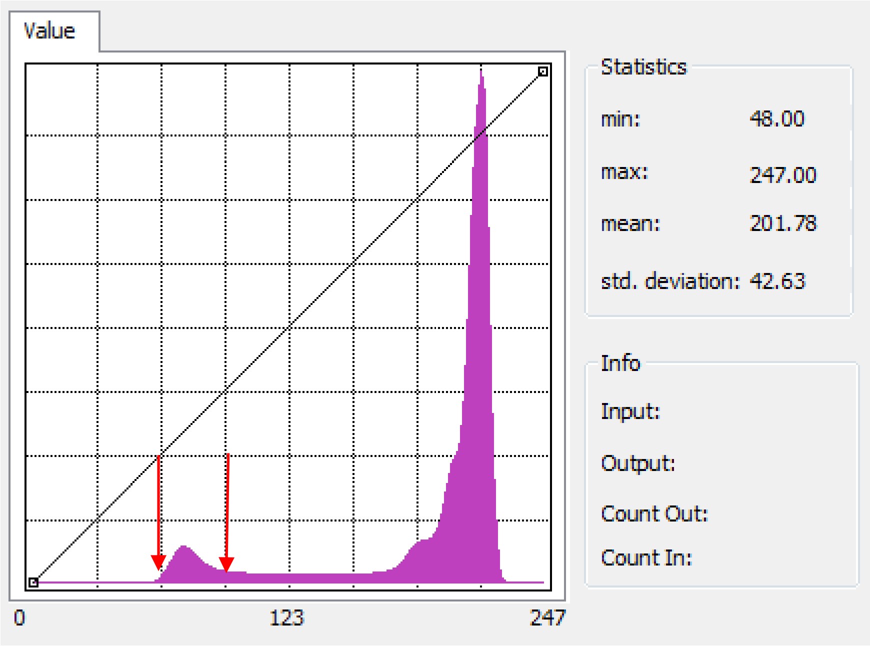
3. Image Enhancement

Most studies actually incorporated the image enhancement into the raster editing step; however, a couple of studies did it prior to segmentation which is why it is listed separately here. Various techniques can be applied to enhance the raster image created by scanning. If the image does end up skewed, it can often be deskewed with the scanning software. In one case, Adobe Photoshop and ArcGIS Desktop were used to apply blurring, stretching and cubic convolution resampling to “reduce variation and noise by smoothing the inconsistencies with the colors” (Godfrey and Eveleth, 2015, p 27). Photoshop was used to resample the image in one project where the researchers wanted to reduce the resolution of the original scanned image (Southall et al., 2003). This latter process is a way to turn an archival quality scanned map into a lower resolution raster image without having to scan it again at a lower resolution. A mask can be used to exclude extraneous parts of the map such as the title, legend, scale, neat lines and annotations outside the actual mapped area; this simplifies the segmentation and reduces some of the raster editing that might otherwise need to be done later (Godfrey and Eveleth, 2015).

4. Image Segmentation

Segmentation is the process used to isolate the feature of interest. It is predominantly based on the color characteristics of the feature. “In particular, focus has been made on line extraction on binary images, and in maps on feature extraction on each colored layer” (Lacroix, 2009, p 318). With a pure binary image where black pixels make up the foreground feature of interest and white pixels make up the background or vice versa, there is no need to do any further segmentation. However, with greyscale and color images, there are several ways to accomplish this segmentation.

Simple thresholding can be done on greyscale images just using the pixel color intensity histogram to identify which pixel values best identify the feature of interest – see Figure 2. The situation gets more complicated with color maps. One option is to convert the color maps to 8 bit grayscale and then use the simple thresholding to segment the foreground pixels of interest and background pixels. Color image segmentation can also be accomplished with image processing software such as Photoshop (Southall et al., 2003), GIMP (Arteaga, 2013) or ImageMagick (Pearson et al., 2013). The Southall project tested a class reduction technique where all the colors were first divided into 100 classes and then manually each one of those was assigned to 1 of the 7 land use categories in the map (Southall et al., 2003). R2V, a commercial vectorization tool, has the thresholding function built into it and this was used to isolate geologic contacts and faults in one case (Brown, 2002). Most of the projects – where color features were involved – used remote sensing techniques in software such as Erdas Imagine (Southall et al., 2003), Definiens eCognition/Professional (Bracke et al., 2008; Jung, 2009), and ArcGIS (Godfrey and Eveleth, 2015; Marciano et al., 2013; Southall et al., 2003) to perform the segmentation.



Statistics

min: 48.00

max: 247.00

mean: 201.78

std. deviation: 42.63

Info

Input:

Output:

Count Out:

Count In:

Figure 2. Grayscale thresholding histogram with x-axis showing intensity and y-axis showing number of pixels. Foreground pixels of interest are approximately between the 60 and 90 intensity levels.

Remote sensing classification techniques, normally applied to classifying satellite image pixels by their spectral reflectance values, can be adapted for classifying map images. There are two major types of classification: unsupervised and supervised. In unsupervised classification, the number of different classes are specified by the user and the algorithms will cluster similar pixels – based on shared spectral patterns – into the specified number of classes; this method utilizes clustering statistical methods. In supervised classification, the user selects “training samples” of similar pixels from the image and assigns them to a class; based on the training sample, the algorithm groups together pixels which are neighbours and have similar pixel values and separates them from groups of pixels which are dissimilar in value.

These training samples can be reused with other maps in the same series provided that the colors in the maps are fairly similar (Southall et al., 2003); this benefit can potentially save a lot of time in the classification process. When picking training samples it is important to pick a number of them for each color zone; a single sample is not sufficient to define a good average for a color zone. Also, it is recommended that training samples for a color zone should be selected from cluttered areas – which also contain text and unwanted map elements in addition to the homogenously colored area – rather than very clean areas; it is believed that this technique will train the algorithm to ignore some of this noise (Southall et al., 2003). Ultimately the goal is to minimize the microscopic heterogeneous noise in macroscopically homogeneous zones without actually losing any legitimate microscopic homogeneous zones (Bracke et al., 2008).

Classification can be either pixel-based or object-based; the latter goes beyond just classifying pixels spectrally but also combines that with structural analysis to use shape and spatial characteristics to group pixels together into objects. Object-based classification can offset some of the problems with variations in the same color as well as the effects of embedded noise (Bracke et al., 2008; Jung, 2009); however, it does require more expertise and experimentation in order to determine the optimal parameters.

Depending on the feature colors and segmentation technique used, the output from the image segmentation will be an image with foreground pixels delineating one of the following categories (Figure 3):

a. lines representing one feature type such as geologic contacts, roads or contours (Brown, 2002; Jung, 2009; Pearson et al., 2013)
b. outlines of areas representing one feature type such as geologic formations (Whitfield, 2005)
c. areas representing one feature type (with one class) such as water bodies or urban areas (Jung, 2009)
d. areas representing one feature type (with multiple classes) such as building types, soil zones, snow loads or land use (Arteaga, 2013; Bracke et al., 2008; Godfrey and Eveleth, 2015; Southall et al., 2003)

The usual reason for using outlines of areas in category b) above is due to fact that the area pixels are indistinguishable from the background pixels.

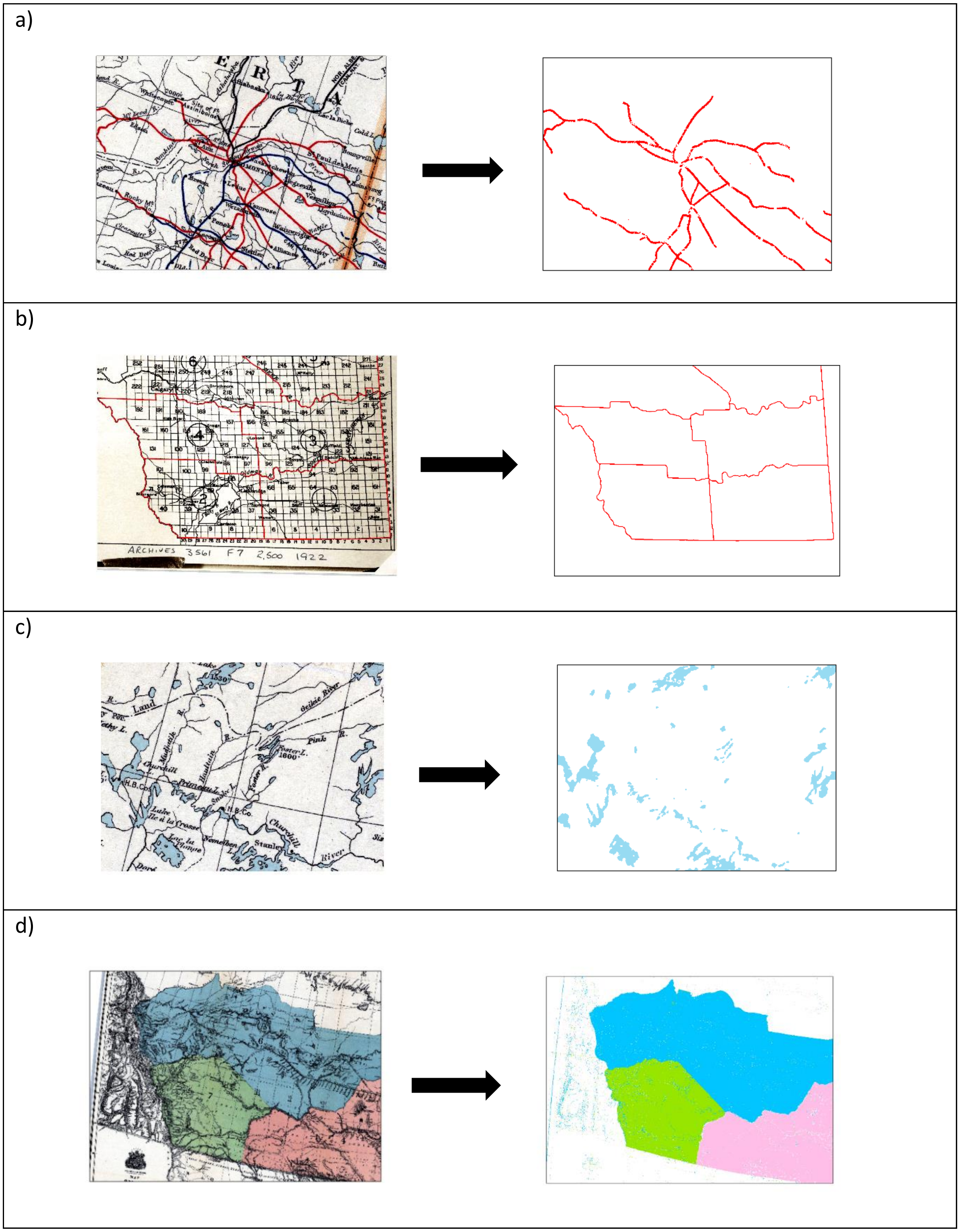


Figure 3. The four categories of output from the segmentation step with original map on left and segmented features on right: a) single class line feature (rail lines), b) area feature outlines (census divisions), c) single class area feature (lakes), and d) multi-class area feature (First Nations treaty areas).

5. Raster Editing

Due to the problems with consistent colors, overlapping text, other intersecting features, and varying line widths, the segmentation process rarely isolates the feature of interest perfectly; therefore, the raster image usually requires some editing before it can be vectorized in an automated way. Three of the examined cases, however, did bypass the raster editing step: for Arteaga (2013) it was due to the automated process followed and choice of software; for Bracke et al. (2008), it was because the segmentation result was exported as a vector geospatial file from the Definiens software resulting in automatic vectorization; and, for Pearson et al. (2013), it was for the reason that the purpose of their study was to record errors. Although, raster editing can be quite laborious, it is usually worth the effort; however, it will depend on the software and processes used.

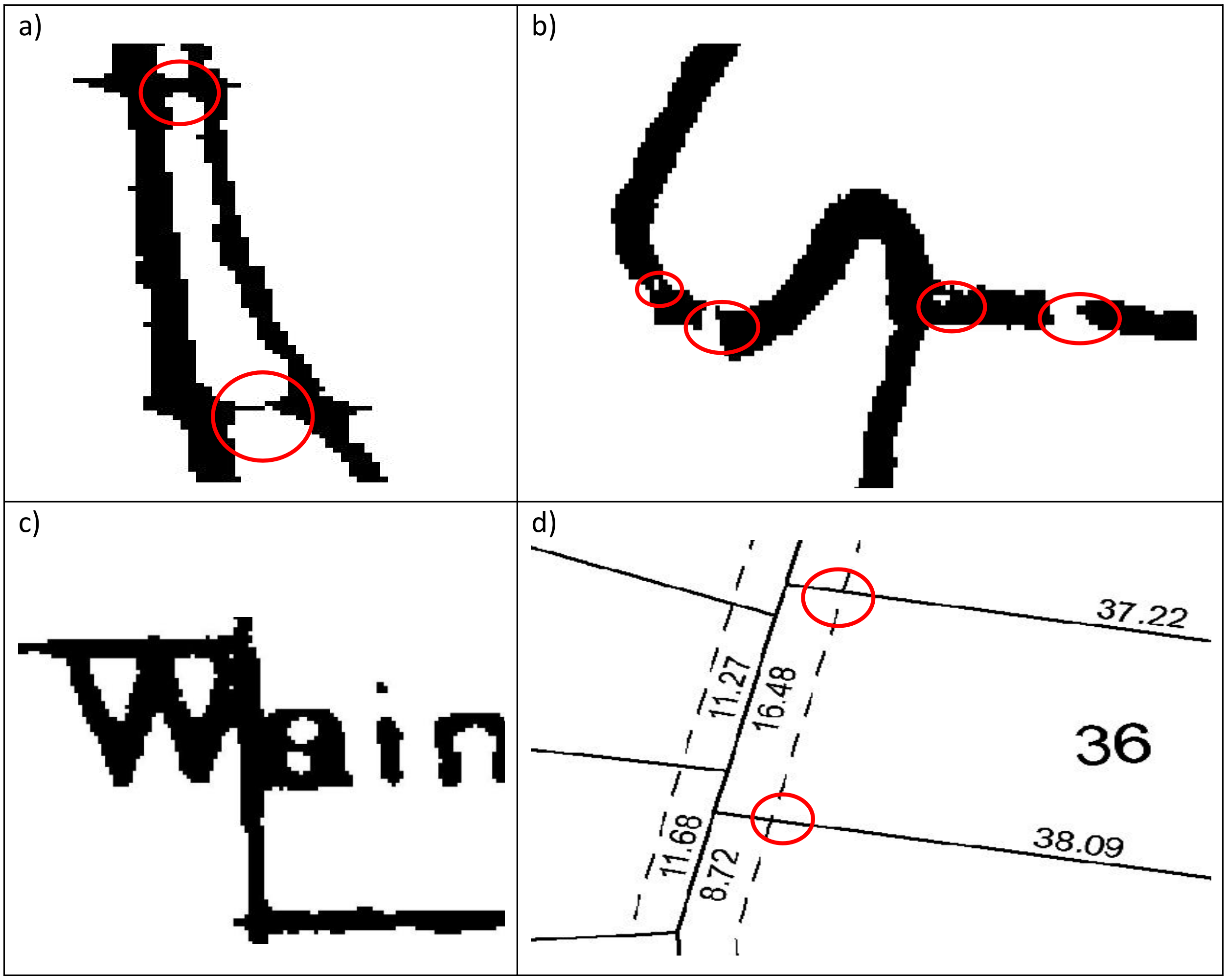


Figure 4. Errors in raster image: a) bridges between two lines that should not be connected; b) gaps and holes in a continuous line; c) text overlapping feature; and, d) intersecting features (dashed power line right of way intersecting solid property lines).

The usual errors with foreground pixels representing lines and outlines are holes within the pixel group making up a line feature, breaks (gaps) in a continuous line feature, connected lines that should be separate (bridges), and intersecting pixels belonging to other map elements such as text or other features – see Figure 4. For foreground pixels representing areas, the errors are leftover pixels from unwanted map elements within the feature areas, holes, and misclassified pixels. The main tools available to repair these issues include morphological operators, GIS functions, bulk erase, manual erase, and manual draw/paint.

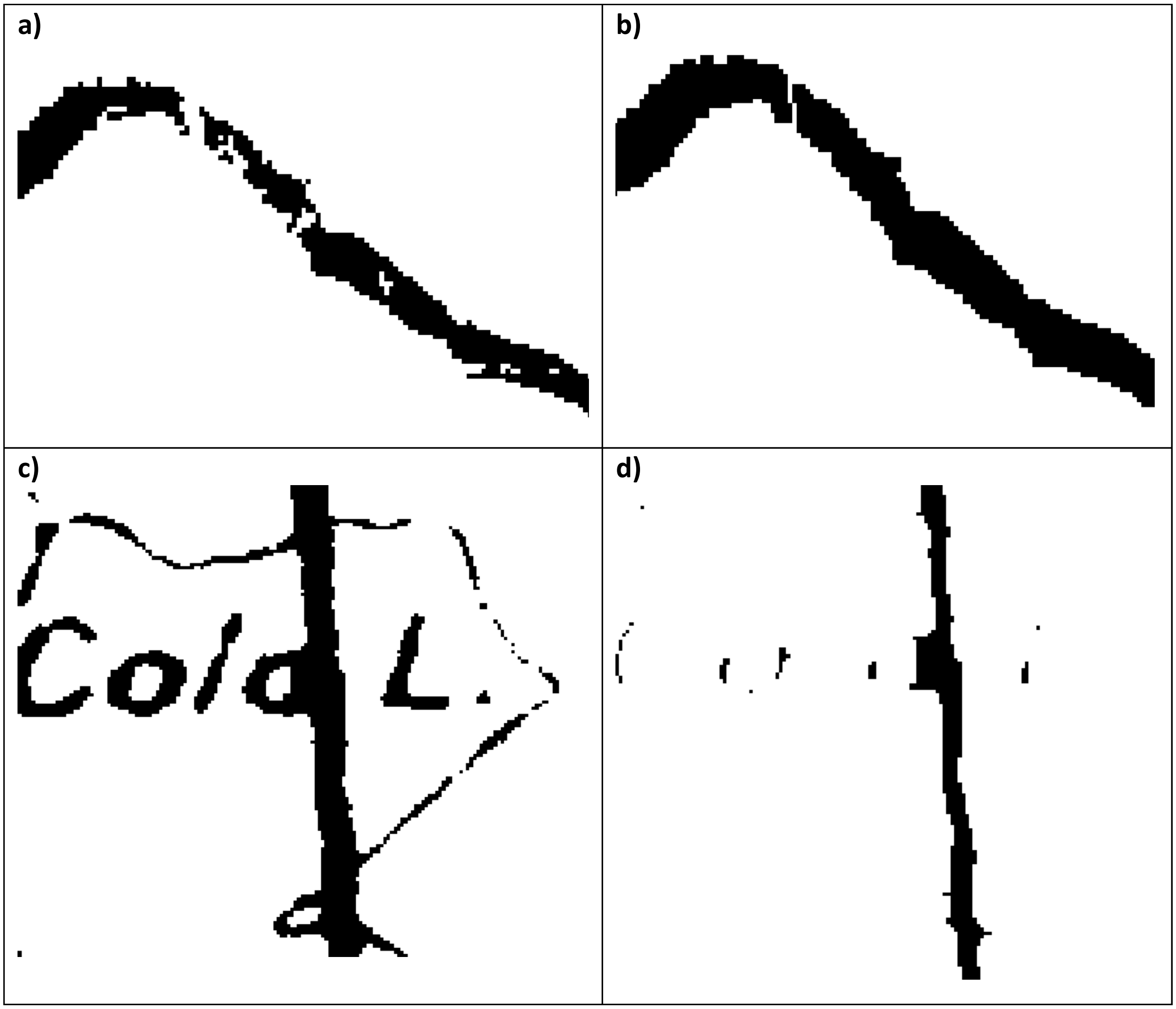


Figure 5. Morphological Operators: a) pixels representing line with gaps and holes; b) Dilate operator applied to line in (a) fixing holes and gaps; c) pixels representing a boundary line that is intersected by a lake feature outline and text with small noise pixel at bottom left; and, d) Erode operator applied to line in (c) removing the intersecting text and lake outline as well as noise – a few bits of leftover text remaining that can be easily erased.

The use of morphological operators automates the editing of foreground pixels delineating lines, outlines and single class feature areas (Jung, 2009). Morphological operators are filters – composed of a small array of pixels – that are applied to each pixel in the raster image. If the pixels in the filter match those in the underlying image then it is a "hit" and if not then a "miss". Depending on the operator type, the hit or miss results in a certain action. The two most common operators are erode and dilate which are correspondingly used to remove or add foreground pixels to a raster image – see Figure 5. The erode operator can be used to remove text, unwanted intersections, and bridges; the dilate operator can be used to fill holes and to close gaps (Chiang, 2010; Chiang et al., 2005, 2014). Care must be taken when applying these operators because fixing one problem can sometimes create another: for example, fixing gaps can create bridges. Morphologicial operators can also be applied iteratively to achieve the required clean up (Jung, 2009). ArcGIS Desktop includes the ArcScan extension which comes with the erode, dilate, opening (erode then dilate), closing (dilate then erode) morphological operators (ESRI, 2016b).

If there are still problematic pixels or missing pixels after applying the morphological operators, the manual erase and manual draw/paint can be used to fix these problems more precisely. With both ArcScan and R2V there is also the ability to bulk select connected foreground pixels based on parameters such as area, diagonal length and width; once selected, the foreground pixels can be changed into background pixels and vice versa (Able Software Corporation, 2008; ESRI, 2016b). ArcScan's magic eraser will bulk erase connected foreground pixels by touching them with the tool or drawing a box around them with it (Whitfield, 2005). ArcScan also has a gap setting (width & angle) and a hole setting: these will respectively direct the automatic vectorization to leap any matching gaps and to ignore smaller holes. One other very handy element of ArcScan is the ability to preview in advance what the vectorization will look like as each edit is made (ESRI, 2016b). In both ArcScan and R2V it is easy to undo changes that don't result in the desired outcome.

With foreground pixels delineating multiple class feature areas, GIS functions such as ArcGIS' Nibble, Shrink and Expand tools and Majority Filter (ESRI, 2016c) are the most automated way to repair them – see Figure 6. The Southall study (2003) applied the Majority Filter first to get rid of the bulk of the unwanted noise pixels within the feature areas; this replaces the noise pixels with the value of the majority of their contiguous neighbours. The Nibble was then applied to eat up the small leftover noise bits and remove them (Southall et al., 2003). Another investigation (Godfrey and Eveleth, 2015) took a slightly different approach since their unwanted map elements had been removed completely from the raster image leaving No Data areas. This was due to the iterative process they used of creating a separate raster for the best matching feature class by masking everything else out each time. Following each iteration, they ran the Shrink and Expand to fix inconsistencies with the edges and after merging all the separate raster images together, they applied the Expand tool again to fill any remaining No Data gaps. Although not explicitly stated, it appears that a similar approach for dealing with distortion on the edges was achieved in another case through the application of the Boundary Clean operation; this smooths boundaries with a combined Expand and Shrink in one or two passes (Marciano et al., 2013).

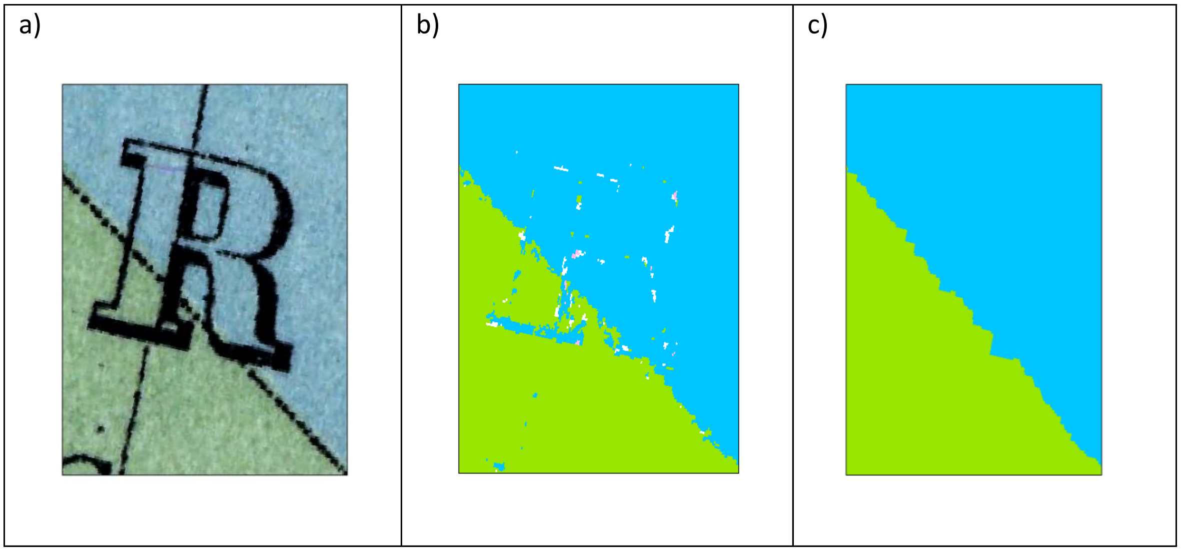


Figure 6. a) Original map with overlapping text and intersecting lines; b) Segmented map showing boundary between 2 multi-class areas; c) Using Majority filter, Expand and Shrink to remove leftover noise and unwanted map elements.

6. Raster to Vector Conversion (Vectorization)

The raster to vector conversion depends on the category of output from the segmentation (Figure 3). In the case of feature areas, vectorization is based on the colored raster classes created in the segmentation step and can be performed on a single feature type (one or multiple class). For pixels representing lines, the conversion is done on one feature type at a time.

Automated polygon vectorization is usually achieved through a standard raster to vector conversion process using GIS, such as ArcGIS' RasterToPolygon tool (Godfrey and Eveleth, 2015; Southall et al., 2003) or in NYPL's case the open source GDAL's Polygonize tool (Arteaga, 2013). The projects that used the Definiens software for segmentation/classification were able to directly export the resulting raster as a vector polygon shapefile (Bracke et al., 2008; Jung, 2009). ArcScan can be used to vectorize polygon-like pixel groups that exceed a specified pixel width; this requires a binary image and can only be done on a single feature type at a time. Another option is using ArcScan to vectorize the boundaries of feature areas as outlines and then later convert them into polygons (Whitfield, 2005) in the vector editing stage.

Lines are inherently more problematic to vectorize in an automated fashion. A line or a boundary on a raster map image consists of a certain thickness of pixels; however, due to quality issues – either on the original map or from the scanning process – this thickness will not be uniform throughout. This makes it more difficult for software to accurately vectorize lines – see Figure 7. In addition, corners, junctions and line intersections are harder to interpret – especially low angle intersections. ArcScan has three settings – geometrical (preserves angles and straight lines), median (designed for non-rectilinear angles) and none (designed for non-intersecting features) – that can be applied to mitigate some of the issues with intersections. The commercial software that was used for line vectorization in the projects examined was either ArcScan (Whitfield, 2005; Jung, 2009) or R2V (Brown, 2002). Due to the requirement of a binary image, lines of only one color can be vectorized at a time.

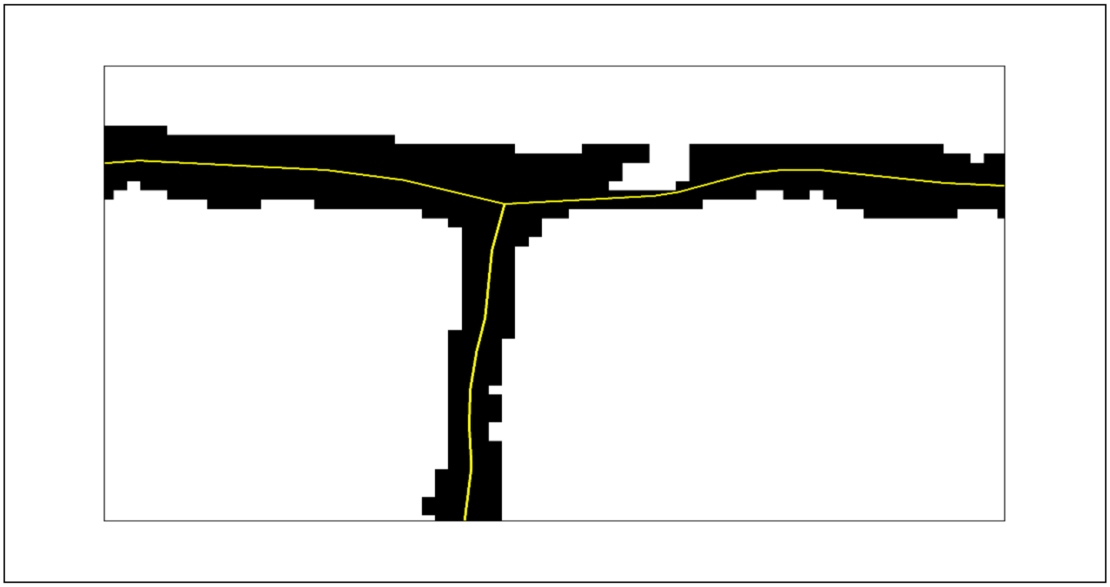


Figure 7. How differing line width can impact vectorization of a right angle intersection.

If the majority of errors in the raster map image have been removed or corrected, the automated vectorization should output a reasonable vector line layer. It will never be entirely perfect and some vector editing may be required, but it is a real time-saver (Whitfield, 2005). When the raster image simply has too many problems – with noise, missing pixels, and intersecting unwanted pixels – to be edited in a reasonable amount of time, the other option is to use interactive raster tracing to delineate the lines or boundaries. With interactive raster tracing, the user clicks on the line pixels and indicates the direction; the software then automatically traces a line until it encounters a spot – usually an intersection or gap – where it doesn't know which way to proceed. The user then points it in the right direction and off it goes again to the next ambiguous spot. This process is essentially a semi-automated form of heads-up digitizing. ArcScan and R2V include both automated vectorization and interactive raster tracing as well as the option to select areas of the raster image for either automatic or trace vectorization. This allows for a hybrid approach: automatic vectorizing of clean straightforward areas and trace vectorizing of noisier more complex areas. The different areas can then be exported and merged together into one vector line layer.

7. Vector Editing

Vector editing is the final phase of the entire process. There are a number of different operations done in the vector editing step: cleaning up and fixing errors, dealing with leftover noise, filling No Data gaps, smoothing lines and polygons, and assigning attribute data to features.

A number of strategies were applied to dealing with the leftover noise or No Data gaps on multiple-class feature type polygons. Southall et al. (2003) ran the ArcGIS eliminate function to dissolve areas of unwanted map elements (below a minimum threshold) into the polygon with the longest shared boundary. Bracke et al. (2008) closed up these gaps with empty filler polygons in ArcGIS. Then, using a Spatial Join operation twice, they assigned the class from the nearest polygon to each empty filler polygon. The dissolve tool was then applied to aggregate all the smaller filler polygons that intersected or were contained within larger same class feature polygons. As none of the above methods were foolproof in assigning the correct category, some manual recoding was required for a few misclassified polygons. Godfrey and Eveleth (2013) also dissolved their polygon features to clean things up.

Smoothing of the vector polygon boundaries was done with the ArcGIS Smooth Polygon tool in the Godfrey and Eveleth project (2013) in contrast to some of the others who had done this as part of the raster editing step. They also had to clip the vector result to the extent of the original mapped area; this was necessitated by the spillover caused by their use of the Expand tool to fill in all the No Data gaps during the raster editing phase.

The NYPL project did not do any raster editing prior to vectorization (Arteaga, 2013). Most of their work was done in the vector editing stage. They used the R software for shape simplification and for polygon exclusion; the latter was based on minimum and maximum thresholds. Further polygon exclusion was done by comparing the polygon to the color of the corresponding area on the raster; the white polygons – which corresponded to the background – were removed. Although not stated in the Arteaga description, it appears that since their initial project was begun, NYPL has developed a crowd-sourced tool, Building Inspector, to assist with the quality control work of checking, and if necessary, modifying building polygons through the adjustment of vertices (New York Public Library, n.d.). NYPL has put together all the script and templates that went into their project into an open source tool called Map Vectorizer which is available through GitHub (Arteaga, 2017).

As for the vector editing of the extracted line features (including polygon boundary lines), it is important to compare it to the original map so that any missing or erroneous lines be corrected. These can be fixed with ArcGIS' Edit tools. In the case of boundary lines these can be converted to polygons using ArcGIS' FeatureToPolygon tool (Whitfield, 2005). The vector lines created through vectorization often have too many vertices; this gives the resulting lines a stair-case appearance. ArcGIS has a Smooth Line tool that can make the line look better. The R2V software also has a built-in "smooth lines" command to do something similar (Brown, 2002). Pearson et al. (2013) used the novel approach of smoothing the extracted contour lines by converting the vector back to raster and then re-vectorizing.

Once the vector lines and polygons have been finalized there is still one more task to complete – the addition of attribute data to the features. These can be road names, river names, administrative units, geologic units, etc. In the case of single or multiple class feature polygons, it is a straightforward process to assign each class a proper name by editing the existing class name. Where more detail is required or in the case of most linear features, this is accomplished by creating fields in each feature's attribute table and populating them with their corresponding data. Although this is largely a manual process, it can be expedited by the use of lookup tables; one basic field is entered in the attribute table and then it is joined to the lookup table on the key field (Whitfield, 2005). This, in essence, automatically transfers all of the corresponding information in the other fields to the feature.

Discussion

The previous sections have highlighted what seems to be a myriad of ways to extract features from a raster map and convert them to a vector geospatial layer, but it is important to keep in mind that the eleven different cases really aren't that different in their overall scheme; they do the same thing but just differ slightly in which tools are used and when. Each study was unique though, and employed some novel techniques worth considering for any map vectorization project. The key point is that a certain amount of experimentation will be required upfront to identify the optimal processes for any vectorization project.

The goal of this review was not to judge these methods but rather to report them. In order to state that one method is better than another, a comprehensive comparison of the results from all methods would need to be done on the same map and that was beyond the scope of this paper. Even then, some methods may work better for a particular kind of map or the specific feature to be extracted. These kinds of comparative analyses would be potential areas for further research.

When evaluating the methods used, the quality of the end result is not the only factor to consider. The time required to complete the whole process is just as important, particularly when vectorizing large numbers of maps. Most of the reviewed studies did not mention the specific time involved so it was not possible to compare them by this factor; however, the Southall et al. study (2003) did compare the time requirements between its two methods as well as with a full manual approach. The upshot is that achieving greater spatial accuracy will usually require more time; this means balancing the trade-off between accuracy and time taken to best serve the potential map use.

The requisite spatial accuracy is determined by the ultimate use of the geospatial vector layer and is impacted by all the steps in the process. Although vector layers are scalable – basically to any scale level – they are usually intended for a specific narrow range of scales depending on their purpose. Levachkine identified two types of GIS: Analytical GIS and Register GIS (Levachkine, 2004). Analytical GIS does not require the same high level of accuracy as Register GIS, because the former entails working with thematic (soils, geology, vegetation, etc.) data which is usually less exact and at smaller scales. On the other hand, exactness is much more of an issue for Register GIS as it is concerned with topographic (contours), cadastral (properties and buildings), utility or transportation data at larger scales. The extraction of buildings (Arteaga, 2013), roads (Jung, 2009) and contour lines (Pearson et al., 2013) would be categorized as Register GIS; the other projects would all fall into the Analytical GIS category (Bracke et al., 2008; Brown, 2002; Godfrey and Eveleth, 2015; Marciano et al., 2013; Southall et al., 2003; Whitfield, 2005).

This review did not comprehensively test the different software; although some experimentation was done to better understand the processes, it was not applied in a systematic way. Therefore, the review can't recommend one software solution over another. The focus, as specified earlier, was on projects that used either commercially available or open source solutions and each program used was identified in its corresponding step. The selected studies were done over a period of time from 2003 to the present and one must keep in mind that the different software have probably evolved over the same period. Where doing a specific task with a certain software may not have been possible a decade ago, it may now be possible to do so.

Through the close examination of these nine studies, it has been demonstrated that there is no easy-to-use, one-size-fits-all automated solution – that would work for all maps and all features – for extracting features from a paper map and converting them into vector geospatial layers. The most progress on automation has been achieved in the segmentation and vectorization steps, but there have been some developments in the raster and vector editing steps as well. The type of map and feature (and amount of noise) will largely determine the level of automation that can be exploited, but in all cases some manual intervention and handling are unavoidable.

The significant decline in the use of paper maps has prompted many libraries to either put into storage or give away large parts of their collections. Efforts to scan and make them more easily and universally accessible have breathed some new life into maps and played an important role in their preservation. The resulting raster map images also provide libraries with a tremendous, largely unrealized opportunity: mining these raster maps for their features of interest and converting those features into geospatial vector layers unleashes all kinds of possibilities for customized maps and spatial analysis using GIS, not to mention easier discovery and the augmentation of library geospatial data collections. It is within the means of libraries to accomplish this using currently available software and following the steps and methods outlined above.

References

Able Software Corporation. (2008). *R2V User's Manual: Advanced Raster to Vector Conversion Software*. [online] Available at: http://www.ablesw.com/r2v/R2Vmanual.pdf [Accessed 12 May 2017].

Allord, G., Fishburn, K. and Walter, J. (2014). *Standard for the U.S. Geological Survey Historical Topographic Map Collection*. [online] Available at: https://dx.doi.org/10.3133/tm11B03 [Accessed 4 May 2017].

Arteaga, M. (2013). Historical Map Polygon and Feature Extractor. In: *Proceedings of the 1st ACM SIGSPATIAL International Workshop on MapInteraction*. [online] New York, NY: ACM, pp. 66–71. Available at: dx.doi.org/10.1145/2534931.2534932 [Accessed 18 Apr 2017].

Arteaga, M. (2017). [software]. *Map Vectorizer*. [online] Available at: https://github.com/nypl-spacetime/map-vectorizer [accessed 21 May 2017].

Bracke, M., Miller, C. and Kim, J. (2008). Adding value to digitizing with GIS. *Library Hi Tech*, [online] 26(2), pp. 201-212. Available at: http://10.1108/07378830810880315 [Accessed 19 Apr 2017].

Brown, K. (2002). *Raster to Vector Conversion of Geologic Maps: Using R2V from Able Software Corporation*. [online] Available at: https://pubs.usgs.gov/of/2002/of02-370/brown.htm [Accessed 5 May 2017].

Cartography Associates. (2017). *David Rumsey Map Collection*. [online] Available at: http://www.davidrumsey.com/ [Accessed 9 May 2017].

Chiang, Y-Y. (2010). *Harvesting Geographic Features From Heterogeneous Raster Maps*. PhD. University of Southern California.

Chiang, Y-Y., Knoblock, C. and Chen, C. (2005). Automatic Extraction of Road Intersections from Raster Maps. In *Proceedings of the 13th Annual ACM International Workshop on Geographic Information Systems*. [online] New York, NY: ACM, pp. 267–276. Available at: dx.doi.org/ 10.1007/s10707-008-0046-3 [Accessed 24 Mar 2017].

Chiang, Y-Y., Leyk, S., Honarvar Nazari, N., Moghaddam, S. and Tan, T. (2016). Assessing the impact of graphical quality on automatic text recognition in digital maps. *Computers & Geosciences*, [online] 93, pp. 21–35. Available at: dx.doi.org/ 10.1016/j.cageo.2016.04.013 [Accessed 12 Apr 2017].

Chiang, Y-Y., Leyk, S. and Knoblock, C. (2014). A Survey of Digital Map Processing Techniques. *ACM Computing Surveys*, [online] 47(1). Available at: https://doi.org/10.1145/2557423 [Accessed 11 May 2017].

ESRI. (2016a). *Fundamentals of Georeferencing a Raster Dataset*. [online] Available at: http://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/fundamentals-for-georeferencing-a-raster-dataset.htm#GUID-4DEEE2E1-E031-4CEA-9318-8CA707ED31CB [Accessed 5 May 2017].

ESRI. (2016b). *Getting Started with ArcScan: ArcMap 10.4*. [online] Available at: http://desktop.arcgis.com/en/arcmap/10.4/extensions/arcscan/what-is-arcscan-.htm [accessed 11 May 2017].

ESRI. (2016c). *Generalizing zones with Nibble, Shrink and Expand*. [online] Available at: http://desktop.arcgis.com/en/arcmap/10.4/tools/spatial-analyst-toolbox/generalizing-zones-with-nibble-shrink-and-expand.htm [accessed 12 May 2017].

ESRI. (2016d). *What is raster data?*. [online] Available at: http://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/what-is-raster-data.htm [Accessed 25 May 2017].

ESRI. (n.d.). *GIS for Humanitarian Mine Action: Georeferencing and Digitizing Web Course*. [online] Available at: https://www.esri.com/training/catalog/57630434851d31e02a43ef72/gis-for-humanitarian-mine-action:-georeferencing-and-digitizing/ [Accessed 4 May 2017a].

ESRI. (n.d.). Vector. In: *GIS Dictionary*, [online] Available at: http://support.esri.com/other-resources/gis-dictionary/term/vector [Accessed 25 May 2017b].

Fleet, C., Kowal, K. and Pridal, P. (2012). Georeferencer: Crowdsourced Georeferencing for Map Library Collections. *D-Lib Magazine*, [online] 18(11/12). Available at: dx.doi.org/ 10.1045/november2012-fleet [Accessed 5 May 2017].

Godfrey, B. and Eveleth, H. (2015). An Adaptable Approach for Generating Vector Features from Scanned Historical Thematic Maps Using Image Enhancement and Remote Sensing Techniques in a Geographic Information System. *Journal of Map & Geography Libraries*, [online] 11(1), pp. 18-36. Available at: dx.doi.org/10.1080/15420353.2014.1001107 [Accessed 24 Mar 2017].

Jung, W.R. (2009). *Vector Feature Extraction Using Object-Oriented Image Analysis Techniques from Scanned Maps*. M.A.Western Michigan University.

Lacroix, V. (2009). Raster-to-vector conversion: problems and tools towards a solution a map segmentation application. In *Proceedings of the 7th International Conference on Advances in Pattern Recognition, ICAPR 2009*, [online] pp. 318–321. Available at: dx.doi.org/ 10.1109/ICAPR.2009.96 [Accessed 24 Mar 2017].

Levachkine, S. (2004), Raster to Vector Conversion of Color Cartographic Maps. In: Llados, J. Kwon, Y.,eds., *Graphics Recognition. Recent Advances and Perspectives. GREC 2003. Lecture Notes in Computer Science*, [online] 3088. Berlin: Springer, pp. 50-62. Available at dx.doi.org/10.1007/978-3-540-25977-0_5 [Accessed 24 Mar 2017].

Marciano, R., Allen, R., Hou, C. and Lach, P. (2013), “Big Historical Data” Feature Extraction. *Journal of Map & Geography Libraries*, [online] 9(1/2), pp. 69-80. Available at dx.doi.org/10.1080/15420353.2012.732020 [Accessed 24 Mar 2017].

New York Public Library. (n.d.). “Building Inspector”, [online] Available at: http://buildinginspector.nypl.org/ [Accessed 17 May 2017].

Olson, J. (2009), Which Is It? Scan or Digitize. Make Up Your Mind!. *Journal of Map & Geography Libraries*, [online] 5(1), pp. 108-111. Available at: dx.doi.org/10.1080/15420350802470913 [Accessed 1 May 2017].

Pearson, M., Mohammed, G., Sanchez-Silva, R. and Carbajales, P. (2013), Stanford University Libraries Study: Topographical Map Vectorization and the Impact of Bayer Moiré Defect. *Journal of Map & Geography Libraries*, [online] 9(3), pp. 313–334. Available at: dx.doi.org/10.1080/15420353.2013.820677 [Accessed 24 Mar 2017].

Southall, H., Brown, N. and Burton, N. (2003). *Digitising the Inter-War Land Use Survey of Great Britain: A Pilot Project*. [online] Available at: https://researchportal.port.ac.uk/portal/files/174820/Digitising_LUSGB_2003_Pilot_Project_Report.pdf [Accessed 24 March 2017].

Whitfield, T. (2005). *Capturing and Vectorizing Black Lines from Greenline Mylars*. [online] Available at: https://pubs.usgs.gov/of//2005/1428/whitfield/index.html [Accessed 4 May 2017].

End-notes

[1] Peter Peller is the Director of the Spatial and Numeric Data Services unit at Libraries and Cultural Resources, University of Calgary and can be reached by email: ppeller@ucalgary.ca.

[2] The term “digitize” is a generic term that is sometimes used to individually describe both scanning and vectorization processes which can be confusing (Olson, 2009). To avoid confusion, for the rest of the paper the term “scan” will be used for the processing of creating a raster image from a paper map and the term “vectorize” will be used for the process of converting a raster image to a vector file.

[3] In its simplest form, a raster consists of a matrix of cells (or pixels) organized into columns and rows where each cell contains a value representing information.” (ESRI, 2016d) A scanned map is a raster image.

[4] Vector is “a coordinate-based data model that represents geographic features as points, lines and polygons.” (ESRI, n.d.) Information is associated with each vector feature.

[5] Heads-up digitizing is a process where lines and boundaries are manually traced with a mouse interactively on the computer screen using GIS software.

Appendix

Lead Author	**Arteaga**	**Bracke**	**Brown**	**Godfrey**	**Jung 1**	**Jung 2**
Library Project	✓	✓		✓		
Map						
Type	thematic	thematic	thematic	thematic	topographic	topographic
Number	100s	1	multiple	1	multiple	multiple
Feature(s)	buildings	soil classes	geologic contacts, faults	snow load zones	contours, roads, rail, streams	water, urban, vegetation
1. Scanning						
Resolution		600 dpi	300 dpi			
Color depth		32 bit	24 bit			
Output file type	tiff	tiff	jpeg			
2. Georeferencing						
Software	Map Warper	ArcGIS	R2V	ArcGIS	Definiens	Definiens
3. Image Enhancement						
Software				ArcGIS, Photoshop		
Operations: (b=blur, bi= convert to binary, c=convolution, cr=crop, m=mask, r=resample, s=sharpen, st=stretch)				b, c, m, r, s		
4. Segmentation						
Software	GIMP	Definiens	R2V	ArcGIS	Definiens	Definiens
Type: (RS=remote sensing, T=thresholding, O=other)	T	RS	T	RS	RS	RS
RS Type (PS=pixel supervised, PU=pixel unsupervised, OS=object supervised		OS		PU	OS	OS
RS Algorithm (H=heuristic, I=ISO Cluster, M=Maximum Likelihood, P=Principle Components Analysis)		H		I	H	H
Output Pixel Features (L=lines, M=multi-class areas, S=single-class areas)	M	M	L	M	L	S
5. Raster Editing						
Software			R2V	ArcGIS	ArcScan	ArcScan
Operations (c=convert to binary, e=erase, ex=expand, m=majority, ma=mask, mo=morphological operator, n=nibble, p=paint, s=shrink)			e	ex, ma, s	mo	mo
6. Vectorization						
Software	GDAL Polygonize	Definiens	R2V	ArcGIS	ArcScan	ArcScan
Output (L=lines, P=polygons, PO=polygon outlines)	P	P	L	P	L	P
7. Vector Editing						
Software	R, Building Inspector	ArcGIS	R2V	ArcGIS		

Appendix

Operations(a=attribute data, c=clip, d=dissolve, e=eliminate, fp=feature to polygon, m=merge, me=manual edits, p=polygon exclusion, s=smoothing, sp=spatial join, ss=shape simplification)	a, me, p, ss	d, me, s, sp	s	c, d, s

Lead Author	**Marciano**	**Pearson**	**Southall 1**	**Southall 2**	**Whitfield**
Library Project		✓			
Map					
Type	thematic	topographic	thematic	thematic	thematic
Number	multiple	multiple	multiple	multiple	multiple
Feature(s)	neighborhoods	contours	land use classes	land use classes	geologic contacts
1. Scanning					
Resolution		330-440 ppi	200 dpi	200 dpi	300-400 dpi
Color depth			8 bit	8 bit	
Output file type	tiff	tiff			tiff
2. Georeferencing					
Software	ArcGIS	ArcGIS	ArcGIS	ArcGIS	ArcGIS
3. Image Enhancement					
Software		ImageMagick	ArcGIS, Photoshop, Paintshop Pro	ArcGIS, Photoshop, Paintshop Pro	ArcGIS
Operations: (b=blur, bi= convert to binary, c=convolution, cr=crop, m=mask, r=resample, s=sharpen, st=stretch)		c	cr ,r, s	cr, r, s	bi
4. Segmentation					
Software	ArcGIS	ImageMagick	Paintshop Pro, ArcGIS	Erdas Imagine	scanner
Type: (RS=remote sensing, T=thresholding, O=other)	RS	T	O	RS	T
RS Type (PS=pixel supervised, PU=pixel unsupervised, OS=object supervised	PS			PS	
RS Algorithm (H=heuristic, I=ISO Cluster, M=Maximum Likelihood, P=Principle Components Analysis)	M			P	
Output Pixel Features (L=lines, M=multi-class areas, S=single-class areas)	M	L	M	M	L
5. Raster Editing					
Software	ArcGIS / ArcScan	ImageMagick	ArcGIS	ArcGIS	ArcScan
Operations (c=convert to binary, e=erase, ex=expand, m=majority, ma=mask, mo=morphological operator, n=nibble, p=paint, s=shrink)	c, e, ex, s	c		ex, m, n, s	e, p
6. Vectorization					
Software	ArcScan	Potrace	ArcGIS	ArcGIS	ArcScan
Output (L=lines, P=polygons, PO=polygon outlines)	P	L	P	P	PO
7. Vector Editing					

Appendix

Software	ArcGIS		ArcGIS	ArcGIS	ArcGIS
Operations(a=attribute data, c=clip, d=dissolve, e=eliminate, fp=feature to polygon, m=merge, me=manual edits, p=polygon exclusion, s=smoothing, sp=spatial join, ss=shape simplification)	m		e, me	e, me	a, me, fp