A Toponym-Based Approach to the Automated Georeferencing of Physically Mapped Documents: The TAAG Algorithm

Karim Bahgat[a] and Dan Runfola[b]

[a]Applied Science Department William & Mary, Williamsburg, VA, USA; [b]Integrated Science Center, 540 Landrum Drive, Room 0279, Williamsburg, VA 23185, USA

ARTICLE HISTORY
Compiled November 10, 2020

ABSTRACT
Paper maps are an important source of information in many areas of research, but modern use can be challenging due to the costs of georeferencing maps into digital format. While there have been some efforts to automate map georeferencing, these have so far been limited to specific types of maps and assumptions. This paper presents a novel Toponym-based Approach to Automated Georeferencing (TAAG), using information about the names and locations of toponyms contained in most maps to determine the coordinates of the map contents. Through large-scale testing of computer simulated maps, we demonstrate that TAAG is capable of georeferencing a wide range of maps at accuracy levels similar to that of humans. Further, we explore possible sources of error when using the TAAG algorithm, providing guidance to practitioners or researchers that might seek to implement or extend the methodology.

KEYWORDS
Map georeferencing; map digitizing; map rectification; map registration; control points; automation; toponyms; gazetteers; historical maps.

1. Introduction

Museums and libraries across the world are filled with geographical representations of knowledge in many different forms. These include sketched maps created by early explorers, topological or reference map sheets produced by government agencies, and numerous thematic maps featured in atlases and scientific publications (Bolstad, 2014; Hill, 2009). These maps contain a wealth of knowledge and are an invaluable source of information for contemporary research (Chiang, 2016; Herold, 2017; Herold, Roehm, Hecht, & Meinel, 2011; Hunziker & Cederman, 2017; D. Liu et al., 2018; Muller-Crepon, Hunziker, & Cederman, 2019; Statuto, Cillis, & Picuno, 2017; Thieler & Danforth, 2012; Wucherpfennig, Weidmann, Girardin, Cederman, & Wimmer, 2011, c.f.).

There are a number of reasons why researchers want to digitize physical maps and use them as a source of information. The first reason - common across a range of disciplines - is that maps can contain information available nowhere else. While museums and libraries seek to make large map collections digitally searchable through geographic information retrieval (Budig, 2018; Chiang, 2016; Hill, 2009), historians are

Contact Karim Bahgat. Email: karim.bahgat.norway@gmail.com

finding new ways to explore old maps using modern geospatial technologies (Balletti, 2006; Knowles & Hillier, 2008; Rumsey & Williams, 2002). Conservationists and other researchers use maps to study the long term trends of environmental land cover, land use, sea level change, and urban growth (Herold, 2017; Herold et al., 2011; D. Liu et al., 2018; Statuto et al., 2017; Thieler & Danforth, 2012). Among political scientists, digitizing maps from the recent historical past has contributed to studying the relationship between conflict, ethnic groups, resource distribution, and road infrastructure (Hunziker & Cederman, 2017; Joo & Steinert-Threlkeld, 2018; Muller-Crepon et al., 2019; Wucherpfennig et al., 2011).

A second reason for the use of physical maps is the effort and cost required to survey or gather new field-data; digitizing existing paper maps is often a cheaper and more accessible solution. Since the emergence of geographic information technologies in the 1960s, readily available physical maps have often been used as one of the main sources of information when creating geospatial databases (Bolstad, 2014, 131,133; Goodchild, 1991, 46; Jackson & Woodsford, 1991, 241). More recent examples include databases created based on the digitizing of cadestral property maps (Gielsdorf, Gruendig, & Aschoff, 2003; Mendes, 1995), soil survey maps (Panagos, Jones, Bosco, & Kumar, 2011), botanical fauna atlases (Schölzel, Hense, Hübl, Kühl, & Litt, 2002), or industry maps of oil- and gas-wells (McDonald et al., 1997).

Today, a wide range of maps containing valuable or unique information can be downloaded from the internet at little to no cost (Michelson, Goel, & Knoblock, 2008). While there are several options available to automatically digitize the contents of maps (for a recent overview, see T. Liu, Xu, & Zhang, 2019), fewer options exist to automatically identify the information needed to place those contents in geographic space - the geographic reference frame of the map itself. This paper engages with this challenge by providing a novel algorithm for the process of map georeferencing, a key missing step in the pursuit of fully automated map digitization.

The paper is structured as follows. First, we provide a brief literature review of map digitizing, including automation efforts, advances, and limitations of existing map georeferencing algorithms. Second, we introduce the **T**oponym-based **A**pproach to **A**utomated **G**eoreferencing (TAAG) algorithm, including the underlying theory and implementation strategy. Third, we examine the accuracy of the TAAG algorithm for a range of different maps to illustrate its effectiveness under different circumstances. Finally, we discuss some of the limitations and future goals for toponym-based georeferencing approaches.

1.1. The Map Digitizing Process

The process of extracting and creating a digital record of the information contained in maps is known as "map digitizing". Early approaches to map digitizing involved physically placing the paper map onto a digitizing table, and manually tracing the location and outlines of objects by hand using a digital puck; this approach is known as hardcopy digitizing (see Bolstad, 2014, 140; Jackson & Woodsford, 1991). With advances in paper scanning and computer availability in the 1980s, map digitizing shifted towards a computer environment where the map is scanned as a digital image and loaded on a computer screen; an approach known as on-screen or heads-up digitizing (Bolstad, 2014, 140-142; Jackson & Woodsford, 1991). This approach affords more convenience and functionality such as dynamic zoom, and is how most digitizing is still done today.

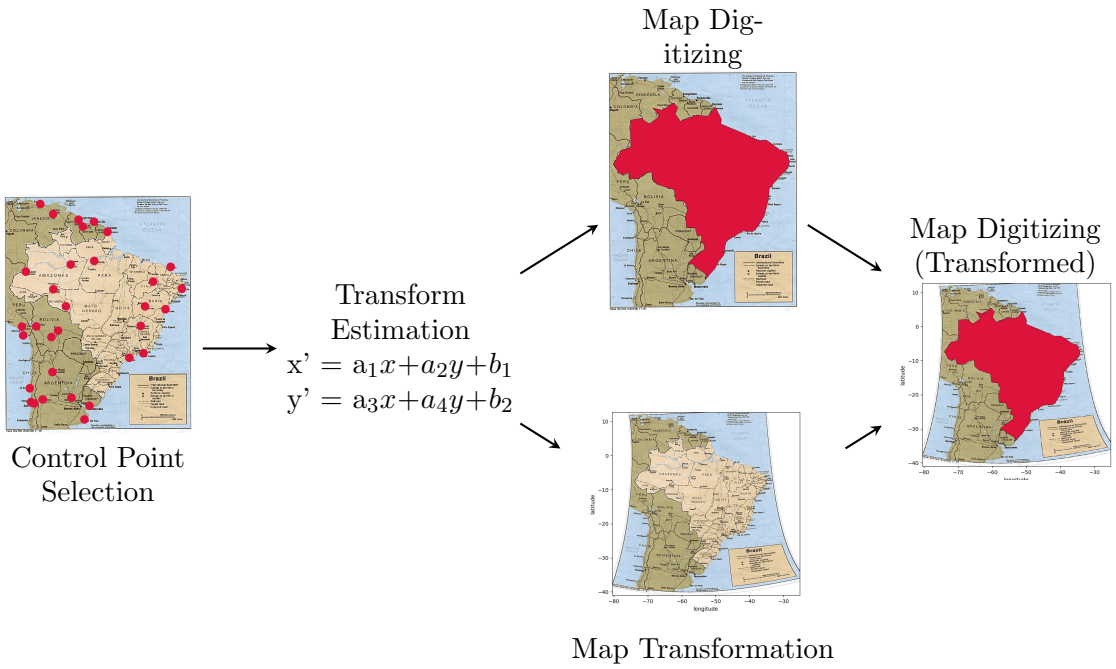

Figure 1. The Map Digitizing Pipeline

In order for the digitized map contents to be meaningful, the map has to be georeferenced so we can know the geographic coordinates of the contents. This georeferencing process is a crucial but often overlooked part of the map digitizing pipeline, and can be seen as the first two steps in Figure 1. In the first step, a selection of control points in the scanned map image, given as centimeter or pixel coordinates, are assigned geographic coordinates representative of locations on the Earth's surface. These control points are selected based on prominent and easily identifiable points on the map, such as towns or road intersections, or based on annotated map corner coordinates for local-scale maps when those are available (Bolstad, 2014; Dowman, 2005). In the second step, these identified control points (for which both the image and geographic coordinates are known) are used as the basis for estimating a transform equation that predicts the geographic location of any point on the map image.

To arrive at the transform function, various approaches are used to estimate a function that approximates the relationship between the chosen control points. Typically this involves estimating a single global polynomial equation of the 1st, 2nd, or 3rd order, which will then be applied uniformly across all image pixels (Bolstad, 2014, 155; Balletti, 2006). Several other alternatives are also possible but not as commonly used due to distortions: these include local transforms using thin-plate-splines (Bolstad, 2014, 155; Balletti, 2006), and hybrid global-local methods (Fuse, Shimizu, & Morichi, 1998; Inoue, Wako, & Shimizu, 2007; Wu, Carceroni, Fang, Zelinka, & Kirmse, 2007).[1]

The estimated transform function computed from the georeferencing process can then be used for digitizing in one of two possible ways (i.e. the two possible pathways in Figure 1). The approach most commonly available in software packages today is

[1]Research into new and better alternatives is ongoing, including improvements to global transforms (Felus & Felus, 2009; Havlíček, 2016; C. Li et al., 2014), local neural network transforms (Yilmaz & Gullu, 2012), and analytical transform methods (Bayer, 2016).

using the estimated transform function to transform the original map image to a new georeferenced map, where each pixel in the original image is resampled to a new location. The user can then use the new georeferenced map as the basis for digitizing, so that the object coordinates are recorded directly in the geographic coordinate system of the transformed map. In the other approach to digitizing – more common in the early days of GIS when digitizing was done directly on the hardcopy paper map – the goal is to digitize the objects directly in the original map image, and then using the estimated transform function to transform the digitized object coordinates in post-processing, i.e. after the data collection is complete (Bolstad, 2014, 155; Jackson & Woodsford, 1991).

1.2. Automated Map Digitizing & Georeferencing: Current Research

One of the challenges with map digitizing is that it is typically a manual process done by hand, making it a very labor-intensive and costly activity (Mendes, 1995). For this reason, automating the map digitizing process has been a high priority for many research groups (for an overview, see T. Liu et al., 2019). Most of the research on automation has focused the digitization of map contents and object extraction, and has a long history that includes various semi-automated and interactive technologies going back to the 1980s (Jackson & Woodsford, 1991; Bolstad, 2014, 149; Jackson & Woodsford, 1991, 246-8; Musavi, Shirvaikar, Ramanathan, & Nekovei, 1988; White, Corson-Rikert, & Maizel, 1987; UNSD, 1996; Samet & Soffer, 1998; Ebi, Lauterbach, & Anheier, 1994; Lawrence, Means, & Ripple, 1996). Continued research and technological innovations in image recognition and machine learning in recent years has resulted in a rich literature and set of available tools for fully automated map content and object extraction (Arteaga, 2013; Auffret et al., 2017; Budig, 2018; Chiang, Leyk, & Knoblock, 2011a; Dhar & Chanda, 2006; Duan & Chiang, 2018; Herold, 2017; Herrault, Sheeren, Fauvel, & Paegelow, 2013; Mahmoud, 2012; Martins, Borbinha, Pedrosa, Gil, & Freire, 2007; J. H. Uhl, Leyk, Chiang, Duan, & Knoblock, 2017).

The development of new map georeferencing technologies have received much less attention, and have been mostly limited to manual or semi-automated tools. These include specialized georeferencing software or tools meant to improve traditional georeferencing workflows (Burt, White, Allord, Then, & Zhu, 2020; Jatnieks, 2010; Jenny & Hurni, 2011), as well as new online-based crowdsourcing websites for map georeferencing (Fleet, Kowal, & Pridal, 2012; Knutzen, 2013). The few solutions that are currently available for automating the map georeferencing process have been customized for high-quality maps from specific organizations or maps that contain detailed information about grid-lines and coordinate labels (Burt, White, & Allord, 2014; Burt, White, Allord, Then, & Zhu, 2019; Burt et al., 2020; Jatnieks, 2010).

The lack of automated tools for map georeferencing today is a major bottleneck for the efficient digitizing of map information. Although the time required depends on many factors (e.g. the desired accuracy, scale, and detail of the map), previous estimates based on anecdotal experience places the typical time required for map georeferencing at about 10 minutes (Burt et al., 2019, 10; Titova & Chernov, 2009). According to data from the crowdsourced georeferencing website MapWarper (Waters, 2017), it took a combined effort of 29,030 work hours (or 3.3 years) to georeference approximately 19,000 maps.[2] This means that an average of 92 minutes was spent on each map; or if we exclude potential outliers, a more realistic average of 18 minutes

[2]Based on user contribution data available from the website, for the period 2015-2019

per map.[3] This presents significant time and cost challenges for any individual or organization needing to georeference or digitize a medium- or large-scale map collection.

1.2.1. Current research on automated georeferencing

Existing research into automated georeferencing can be grouped into two distinct categories (Chen, Knoblock, & Shahabi, 2008; Hackeloeer, Klasing, Krisp, & Meng, 2014; Y. Li & Briggs, 2006). The first is raster-to-raster: for example, when comparing an image taken by an unmanned drone or satellite with another reference image to automatically find its location. The second is raster-to-vector: this occurs when information (such as lines representing roads) is automatically extracted from an image and then compared to a reference vector dataset.

The literature around raster-to-raster georeferencing aims to match an image with another known image, and originates in the remote sensing literature (including satellite imagery, aerial photos, and drones). Several approaches have been proposed for selecting matching control points across the images, including: detecting and matching distinctive pixel regions (Bentoutou, Taleb, Kpalma, & Ronsin, 2005; Long, Jiao, He, & Zhang, 2016; Wong & Clausi, 2007), identifying similarities across linear objects (Chen et al., 2008), and finding matching pixel level correlations (Gonçalves, Gonçalves, Corte-Real, & Teodoro, 2012; Gonçalves, Gonçalves, & Corte-Real, 2008; Karel, Doneus, Briese, Verhoeven, & Pfeifer, 2014).

Raster-to-raster approaches are not as well suited, however, for the georeferencing of paper maps, since most of these methods assume and rely on remote sensing imagery which already contains approximate geographic coordinates from the image metadata (as recorded by the satellite or drone), in order to reduce the complexity of finding and matching with existing imagery in the region (Y. Li & Briggs, 2006, 2). Additionally, cartographic map layers can be stylized in multiple arbitrary ways that would be hard to match with equivalent regions in real world imagery.[4]

The type of approaches more commonly used in the map digitizing literature, and that we focus on here is raster-to-vector georeferencing, or the process of matching a raster image with a known vector reference dataset. One approach is to focus on a prespecified theme shown in the raster map or image, extract that thematic layer, and then use that as the basis for matching. Such theme-based raster-to-vector methods have been applied in both the map digitizing and remote sensing literatures, for instance by identifying road intersections, roads, or buildings from the map or imagery, and matching them with an appropriate vector reference dataset (Chen, Shahabi, & Knoblock, 2004; Chiang, 2010; Cléry, Pierrot-Deseilligny, & Vallet, 2014; Hild & Fritsch, 1998; Song, Keller, Haithcoat, & Davis, 2008; Wu et al., 2007).

There are several limitations to the theme-based approach. First, the proposed methods rely on maps that contain specific types of thematic layers such as roads or buildings, and these would have to be predetermined in order choose the appropriate reference dataset. More importantly, since the objects in the map could be anywhere in the world, this not only requires detailed and historically representative global-level data, but also presents a computational problem in how to efficiently compare the

[3]To exclude outliers where a user may have simply been inactive for a long period of time we only considered carefully georeferenced maps (more than 10 control points placed) and that were completed within a reasonable time limit (less than one hour used). This reduced the total number of maps to 3387, georeferenced in 998 hours. Half of the maps were georeferenced in 13 minutes or more, and 80 percent in 6 minutes or more

[4]However, some have tried to identify basic geometric shapes like road intersections in paper maps and match with road intersections identified in aerial imagery (Chen et al., 2008; Chen, Knoblock, Shahabi, Chiang, & Thakkar, 2004).

control points of a particular map with a potentially much larger global reference set (i.e., point set conflation; Diez, Lopez, & Sellares, 2008; Saalfeld, 1988).

Another raster-to-vector approach, more common in informatics literatures, is to search for map grid lines and tick marks that are explicitly marked with coordinate labels, using line-tracing and text recognition to parse and place control points at the grid intersections (Burt et al., 2014, 2019; Rus et al., 2010; Stavropoulou, 2014; Titova & Chernov, 2009). Herold et al. (2011) generalizes this approach to maps with labelled corner coordinates, typical for large-scale local maps. Jatnieks (2010) suggested a semi-automated approach that automatically zoomed to each map corner, but still requiring the user to place the control point. Burt et al. (2019) also developed a semi-automated tool that uses self-learning to automatically detect control points after the user selects an initial pair of control points. A clear limitation of these gridline- and tick-based approaches is that not all maps contain such information.

The algorithm presented in this paper builds on a third approach utilizing toponym map labels as the basis for matching to a reference dataset of toponym coordinates (Pawlikowski, Ociepa, Markowska-Kaczmar, & Myszkowski, 2012; Weinman, 2013, 2017). There are several advantages of this approach. The majority of maps contain toponyms as a point of reference, regardless of the theme or purpose of the map. Furthermore, there are multiple existing sources of global reference data containing the coordinates of toponyms. Lastly, unlike roads and buildings which may change frequently, places are often named after landscapes or events in the distant past and can persist unchanged for hundreds and sometimes thousands of years (Helleland, 2012, 102).

Research on toponym-based georeferencing is relatively nascent, and has so far been focused on isolated parts of the pipeline. Pawlikowski et al. (2012) demonstrated a pipeline that used toponyms for map georeferencing, but was primarily intended for approximate georeferencing of small overview maps, only considering the general location of each toponym with a scale-rotate transform. Weinman (2013) assumed map toponym labels have already been identified and demonstrated how probability models could be used to determine the most likely matches in the reference gazetteer data. Leyk and Chiang (2016) matched detected map labels with gazetteers to provide context, but assumed the general location of the map was already known. Weinman (2017) showed how toponym georeferencing could be augmented by analyzing the label font styles. We build on these previous efforts to implement a complete and generalized toponym-based approach to automated georeferencing, and demonstrate that the approach can be applied to a large sample of maps with accurate results.

2. Materials and Methods

In this section we present the proposed method for the automated georeferencing of maps. Our method, the Toponym-based Approach to Automated Georeferencing (TAAG), georeferences maps based on information about the names and locations of cities, towns, or villages – commonly labeled features in most maps. The method improves on previous toponym-based efforts by using a new approach to text and toponym recognition, and proposes a novel technique for matching these named places to their real world geographic coordinates - this way generating the control points needed for georeferencing without human intervention. We also detail our approach for testing the accuracy of the method.

2.1. The TAAG Algorithm

The overarching approach of the method is detailed in Figure 2, and each step is discussed in more detail in the following sections. We additionally provide detailed implementation notes in our appendices for readers seeking to replicate or extend our approach or findings.

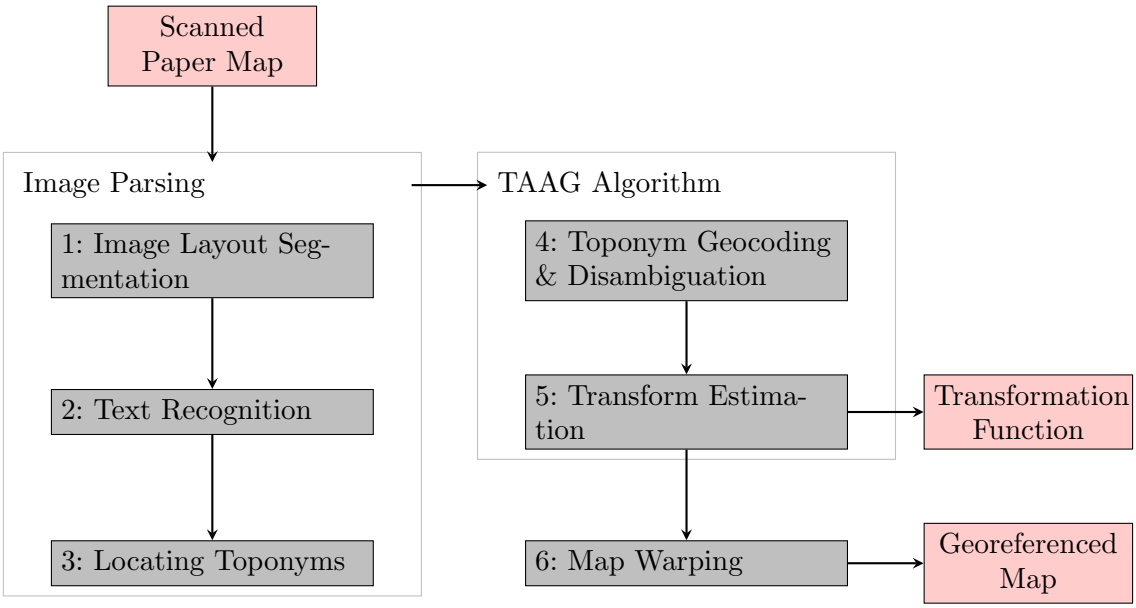

Figure 2. Workflow of Proposed Automated Map Georeferencing

2.1.1. Step 1: Image Layout Segmentation

Image segmentation is a common step in many automated map digitizing methods, and involves identifying collections of pixels that make up the main objects and elements in a map image. Various approaches include grouping pixels based on their color using such methods as K-means clustering (Chiang & Knoblock, 2010; Dhar & Chanda, 2006; Herrault et al., 2013; Roy, Llados, & Pal, 2007), or various step-wise region-growing methods (Centeno, 1997; Chiang, Leyk, & Knoblock, 2011b; Leyk & Boesch, 2010; T. Liu et al., 2016; Roy, Vazquez, Lladós, Baldrich, & Pal, 2007).

For the purpose of this paper, we are only interested in image segmentation as it relates to determining the main graphical layout of the image: specifically a) the outline of the map region, and b) non-geographical information boxes such as the title and map legend. Knowing which parts of the image are used for what purposes will help with narrowing down search regions and interpreting which texts are used as toponyms as opposed to other descriptive text. To identify which pixels to associate with the map itself, we implement a color thresholding approach based on the CIELab Delta E 2000 color difference metric ($\triangle E$); full details on this implementation are provided in Appendix A.

2.1.2. Step 2: Text Recognition

Once we have identified the map portion of the image (i.e. the areas of the image that contain the geographic features we seek to georeference), the next step is to identify the various text labels within this region. Text recognition in map documents presents unique challenges compared to traditional text documents, since the text can occur anywhere in the document and is intermixed and overlaid with various graphics. Many different approaches have been suggested to extracting text from map documents (for an overview, see Chiang, 2016); most approaches first identify and separate individual text pixels from the surrounding graphics, and then iteratively connect these together to form letters, words, and multi-word labels.[5] To accomplish text parsing in our automated context, we implement a three-stage process: (a) the identification of the color of the text, (b) the use of the color information to define some pixels as "text" and other pixels as "not text", and (c) the application of optical character recognition to the output text pixels. We leverage the CIELab color space and Delta E color difference metric ($\triangle E$) in stages a and b; the Tesseract OCR engine is used in step c. Full details on the implementation for each of these steps can be found in Appendix B.

2.1.3. Step 3: Toponym Image Coordinates

Of all text labels identified, we are specifically interested in those that serve as toponyms - placenames such as cities, towns, or villages - to serve as control points for map georeferencing. To determine the exact image coordinate location representative of a given toponym, the text label in the image must be associated with its location marker, generally represented by some type of symbol (typically a black circle or square).

To estimate the correct coordinate for each toponym, we build on past approaches (see, for example, Miao et al., 2017; Reiher et al., 1996; Samet & Soffer, 1996, 1998) by using a contour-based approach that looks for arbitrarily shaped symbols in the neighbourhood of each toponym. For each toponym, a search region is defined that extends 1.5 times the height of toponym font height. The pixels that fall within this search region are then thresholded to isolate collections of black pixels (based on $\triangle E$) that resemble a symbol object (i.e., an equilateral collection of pixels with a size that is approximately half the toponym font height; see Appendix C for more details).

2.1.4. Step 4: Toponym Geocoding & Disambiguation

In this step, we identify the possible geographic coordinates for each of the identified toponyms by searching for each toponym within a set of gazetteers. Because we want to be able to process large batches of maps from all over the world, we need gazetteers that have extensive global coverage, are fast, and that do not have search request limits. For this paper we therefore implemented a local SQLite database that combines the USGS Geographic Names Server (GNS) gazetteer and the GeoNames gazetteer.

A major challenge with multi-gazetteer geocoding, and geocoding more broadly, relates to the disambiguation of commonly used place names. Consider for instance, that we want to create control points for the three towns of "Poli", "Mbe", and "Tchollire" in northern Cameroon (Figure 3a). If we were to search for the coordinates

[5]Although there is some research exploring machine learning-based OCR engines customized for complex multi-colored maps (Pezeshk & Tutwiler, 2011; Weinman, 2017; Weinman et al., 2019), these are still in the early stages and not yet widely available.

of these names in a gazetteer database, we would see that these names are not just native to Cameroon, but are used in many different parts of the world (Figure 3b).

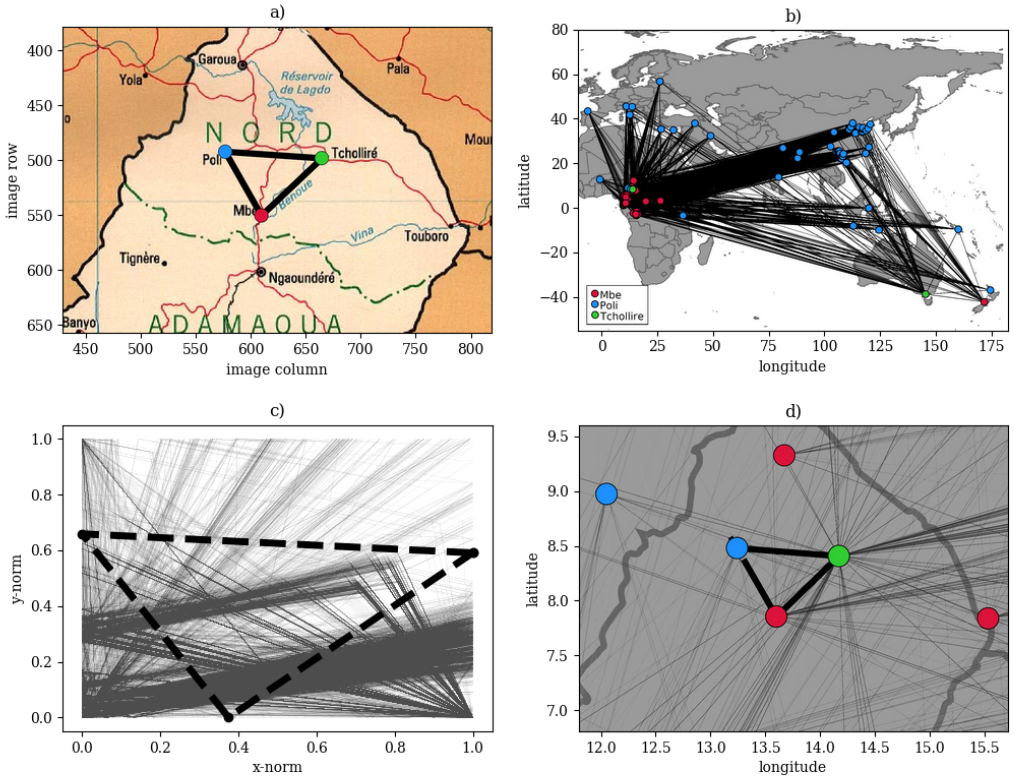

Figure 3. The process of matching image toponyms with candidate gazetteer coordinates: a) the toponyms Mbe, Poli, and Tchollire detected in the image; b) candidate gazetteer matches for the toponyms and lines indicating the spatial patterns formed between them; c) pattern matching of the image toponyms and coordinate combinations in normalized space; d) zoomed-in view highlighting the optimal matching gazetteer combination.

The implication of such place name ambiguity is that there can be many possible coordinate combinations that matches a given set of toponyms (Figure 3b), but only one of those combinations is the one we are interested in (Figure 3d). In the application described here we seek to overcome this ambiguity by matching multiple place names collectively, as opposed to each place name individually. This provides additional information about the entire set of toponyms, which we can exploit by comparing the spatial pattern of the toponyms with the pattern of each possible coordinate combination from the gazetteer.

This approach is illustrated in Figure 3, and can be summarized as an algorithm that identifies groups of toponyms that would form a similar point pattern in both map and geographic space. Those groups that produce the best match, as defined by the overlap in their relative positions in a normalized space (Figure 3c), are selected as "matches" for the toponyms that constitute them. Details on the approach employed to measure pattern similarity and choosing the optimal match, as well as computationally efficient approaches to such matching, are described in Appendix D.

2.1.5. Step 5: Transform Estimation

The set of matching control points (the pixel coordinate x, y, and geographic coordinate $\hat{x}, \hat{y}$) identified from toponyms in the previous sections can be used to estimate a

transform function from image to geographic coordinates. We contrast a small selection of the more commonly used transform functions, specifically the 1st, 2nd, and 3rd order polynomial transforms. Through a leave-one-out RMSE procedure, we determine the optimal subset of control points for each of the transform functions in order to select the best performing one (see Appendix E for a full discussion on how our metrics of accuracy and algorithm choice are implemented). The final transform function provides the necessary information to translate between image and geographic coordinates, and can be used in subsequent steps to warp the map image so it can be overlaid on other geographic data, or used to transform the coordinates of digitized image objects.

2.2. Testing the TAAG Approach

In order to test if the suggested toponym-based approach for map georeferencing will work in practice, we use a novel approach based on computer simulated maps. The typical approach for evaluating accuracy in the map georeferencing literature is to visually inspect the accuracy of a small selection of real-world maps. In our approach, we instead create a large sample of computer-generated maps, and then proceed to georeference these maps. This provides two main benefits:

(1) knowledge of the true coordinates the original map (and hence quantifying the true error from the georeferencing process), and
(2) it allows us control the map characteristics, which can be used to evaluate different types of maps and sources of error.

For the map generation process, we select 26 randomly generated "scenes" (defined by a latitude-longitude coordinate and a map extent or scale) from around the world. Each scene is rendered hundreds of times, using all possible combinations of a set of predefined map parameters and their values (see Table 1). The parameters are chosen based on map characteristics we believe are potential sources of error impacting the georeferencing process.

Following the simulation process outlined above, for each computer generated map we know the true relationship between pixel and geographic coordinates (i.e. the transform function used by the computer when rendering the image). The georeferencing error of our TAAG approach can thus be calculated for any given point as the distance between the true and estimated geographic coordinate. This allows us to calculate how the error fluctuates for any given point in the georeferenced map, and can be given in either pixel distance units or geographic distance (km) units. An example of this error surface is shown in Figure 4, where a georeferenced map is overlaid on the original source map for comparison. In a perfectly georeferenced map, the two maps would be exactly on top of each other.

These continuous error surfaces are useful in that we can derive summary metrics representing the overall error of each georeferenced map. For the purpose of this paper we calculate error as the maximum value in the entire map, which constitutes a conservative accuracy estimate and a more demanding evaluation of the algorithm. As an example, the maximum error metric for the map in Figure 4 would be 141 km based on the upper-right corner where the distortion is highest.

Parameter	Description	Values
Map dimensions		
mapExtent	Map scale (map width in km)	[5000,1000,100,10]
imgResolution	Image resolution (image width in pixels)	[3000,2000,1000]
Toponyms		
numToponyms	Number of toponyms[6]	[80, 40, 20, 10]
toponymDispersed	Distribution of toponyms	[random VS dispersed]
toponymUncertainty	Toponym location uncertainty (km)	[0, 1, 10, 50]
mapProjection	Map projection	[none VS Robinson]
Image noise		
dataNoise	Overlapping map layers	[countries VS countries + roads + rivers + urban]
metaNoise	Non-toponym metadata text	[none VS title + legend + countrynames]
pixelNoise	Random pixel noise (file encoding)	[png VS jpg (quality=10/100)]

Table 1. Simulated map parameters

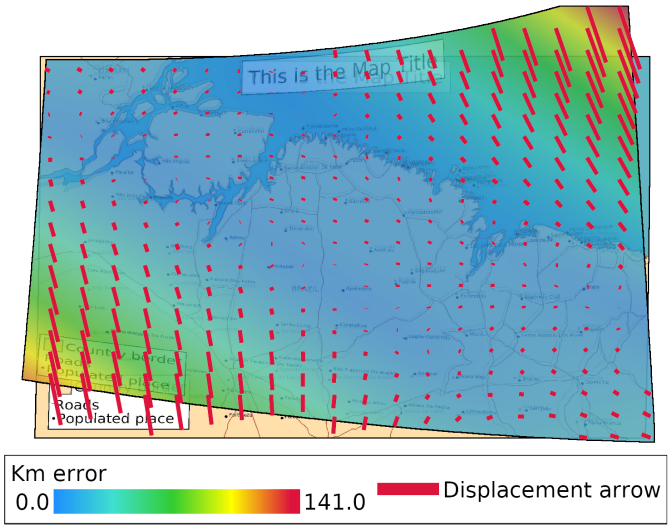

Figure 4. A georeferenced map overlaid on source map with known coordinates. A sample of displacement arrows illustrate the distance between equivalent locations in both maps

3. Results

In total, 24,602 simulated test maps were generated following this process, and subsequently georeferenced using the TAAG algorithm. Of these, 20,506 were used for our analysis, representing the subset that were generated with legible text, recogniz-

ing that the TAAG method is unsuitable for maps with illegible place names.[7] The computing time required to process all 24,602 maps was approximately 539 hours or 22 days, with a median of about 2 minutes per map (see Table 2).[8] In most cases, the majority of the time and computation is spent in the text recognition stage (62%), and the control point matching stage at a distant second (14.8%). For a small selection of outlier cases the control point matching stage takes significantly longer, being responsible for as much as 42% of the average and total time.

Algorithm Stage	Avg	Median	Total time
segmentation	1.5s (1.0%)	1.5s (1.3%)	5.2h (1.0%)
text recognition	82.1s (50.9%)	72.8s (62.6%)	274.2h (50.9%)
toponym candidates	9.6s (5.9%)	7.0s (6.0%)	32.1h (5.9%)
gcp matching	67.7s (42.0%)	17.3s (14.8%)	226.1h (42.0%)
transform estimation	0.2s (0.2%)	0.0s (0.0%)	0.8h (0.2%)
total (excl. warping)	**2.69m (100.0%)**	**1.94m (100.0%)**	**539.0h (100.0%)**

Table 2. Computational cost of the automated georeferencing of 24,602 maps, broken down by the stages of the algorithm.

3.1. Quantifying the Accuracy of the TAAG-based Georeferencing

To enable the comparison of different scale maps in this section, we express the maximum error metric in scale independent units by normalizing the error as a percentage of the map radius, i.e. the number of pixels from the center of the map to any of its corners (Gonçalves, Gonçalves, & Corte-Real, 2009; J. Uhl, Leyk, Chiang, Duan, & Knoblock, 2018). For example, a normalized maximum error metric of 50% would mean a pixel displacement from one of the corners of the map to about halfway towards the map center. Based on this normalized metric, we subset our results into five categories qualitatively representative of the usefulness of the TAAG algorithm under difference cirucmstances: excellent, reasonable, approximate, needs adjustment, and not usable (see Figure 5).

The results of the simulation experiment can be seen in Table 3. Out of a total of 20,506 computer-generated maps, the TAAG algorithm was able to find a solution for 11,372 (58%). Of those it found a solution for, 9,094 (80% of TAAG-georeferenced maps) represented an outcome likely to be of use in a production environment (i.e. accuracy category *Needs adjustment* or better). About 1,961 (17%) would need some form of manual adjustment (20-100% error, relative to image radius); while 2,278 (20%) had errors too high to be usable and would have to be georeferenced manually (>100% error). About half (45%) of all the georeferenced maps were georeferenced to an accuracy similar to or better than what might be expected from a human coder (5% error or less).

Given the exceptionally wide range of simulated map parameters, we further explore a smaller subset more representative of reasonable quality maps that might commonly be found and scanned from a university library. We define this as maps with pixel

[7]For maps rendered at the coarsest resolution (1000 pixel width) with the noisiest file format quality (jpg), we discovered that it was not possible to read the text labels. These were therefore dropped from the analysis, leaving us with 20,506 potentially georeferencable maps.

[8]This is only counting the time required to arrive at the final control points and transform estimation, the steps that would otherwise be performed by a human. It does not count time used warping the final map image in the transformation stage

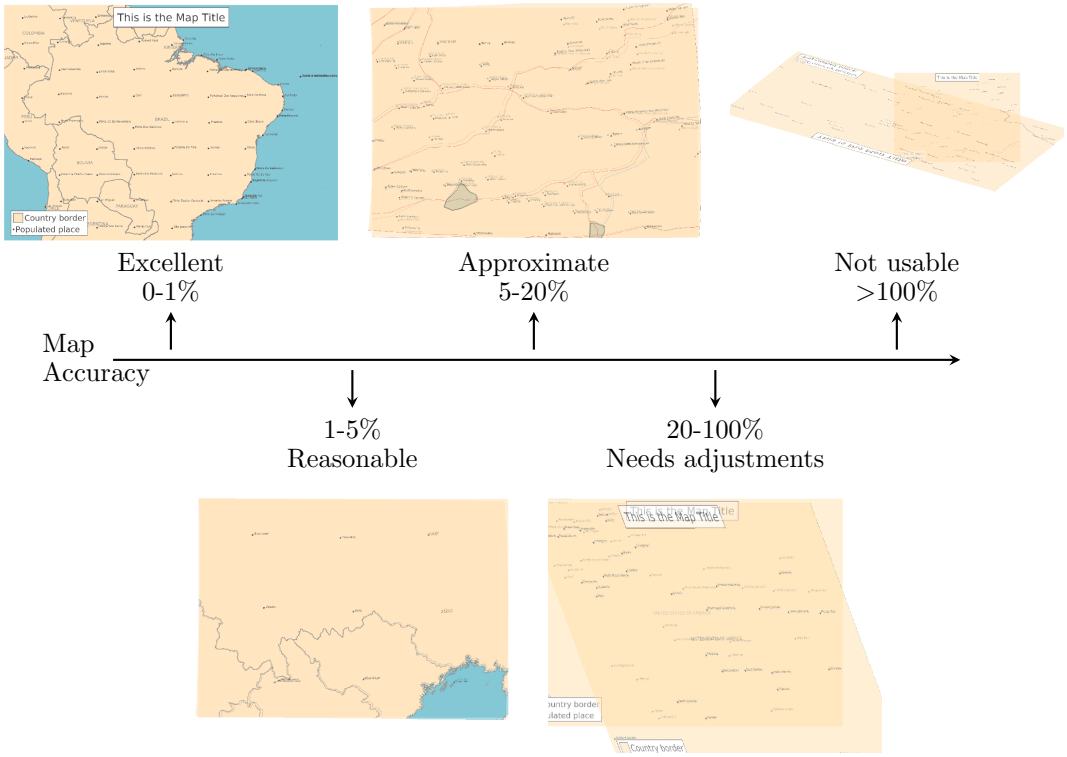

Figure 5. Accuracy categories, with example maps overlaid on source map with known coordinates. Map accuracy is calculated based on normalized maximum map error (as a percentage of image radius).

Accuracy Category	Maps	Percent
Excellent (<1% error)	3,171	15.5%
Reasonable (1-5% error)	1,927	9.4%
Approximate (5-20% error)	2,035	9.9%
Needs adjustment (20-100% error)	1,961	9.6%
Not usable (>100% error)	2,278	11.1%
Failed	8,633	42.1%

Table 3. Accuracy results from the automated georeferencing of simulated maps (n=20,506)

resolutions of 2000 or more, toponym location uncertainty no higher than ca 10 km, at least 40 toponyms, and map extent of at least 100 km. These constraints result in a more realistic sample of maps, and the accuracy results for these are listed in Table 4. Of the resulting subset of 5,760 maps, the TAAG algorithm was able to produce a result for 4861 (84%). 4,656 of these (76% of the total) had map error less than 100%. Over half the maps (52%) were georeferenced at accuracies similar to or better than what might be expected from a human coder (5% error or less). These results give an indication of what we might expect in a real-world scenario.

3.2. Exploring Sources of Error

In this next section, we look in more detail at the observed relationships between simulated map parameters and georeferencing error to explore the main sources of chal-

Accuracy Category	Maps	Percent
Excellent (<1% error)	1987	34.5%
Reasonable (1-5% error)	909	15.8%
Approximate (5-20% error)	749	13.0%
Needs adjustment (20-100% error)	643	11.2%
Not usable (>100% error)	461	8.0%
Failed	976	15.6%

Table 4. Accuracy results for a representative sample of simulated maps with reasonable quality (n=5,760)

lenge for the TAAG algorithm, building on previous approaches to exploring sources of georeferencing errors (Liew, Wang, and Cheah 2012; A. Sisman 2014; Y. Sisman and Sisman 2017; Q. Tan, Lu, Dong, and Zhu 2013). First, we look at the share of maps that were successfully georeferenced across the different map parameters, specifically looking at the share of georeferenced maps that can actually be used (less than 100% error), as well as the share of georeferenced maps with accuracies that equal or outperform human georeferencing (less than 5% error).

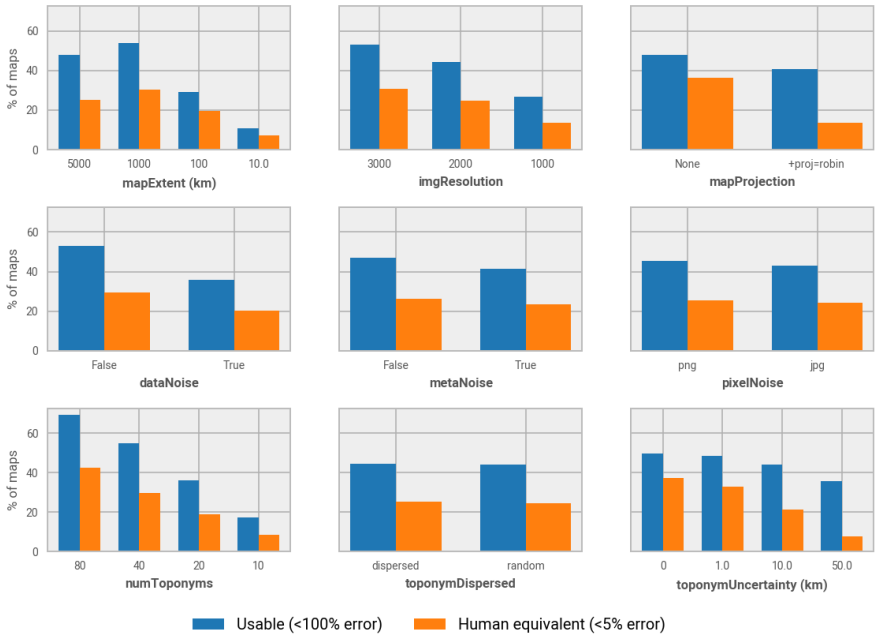

Figure 6. Share of maps that were successfully georeferenced, by broad accuracy grouping.

Based on these results, shown in Figure 6, the most important factor for georeferencing success is the number of toponyms shown in the map, where almost 70% of the maps with 80 toponyms resulted in usable georeferenced maps, decreasing to less than 20% for maps with only 10 toponyms. The map extent is the next most important factor, with usable georeferencing success rapidly dropping from 50% to 20% for smaller-scale maps. Map projection and toponym uncertainty (simulated by random perturbation) also sharply decrease the success rate, but only when defined at the level of human accuracy (<5% error). The rest of the map parameters have only moderate effect (image resolution and data noise) or negligible effect (toponym

dispersal, metadata, and pixel noise) on the success rates.

We further present an analysis of accuracy across different map extents in Figure 7. At large map scales, we observe considerable sensitivity to the physical map projection, the number of toponyms, and amount of data noise. These factors are generally less important at small map scales. Conversely, toponym uncertainty is relatively unimportant for large-scale maps, but rather only if the source map is of a fine scale. While many of these effects are intuitive, we comment on the underlying reasons for these interactions in our discussion.

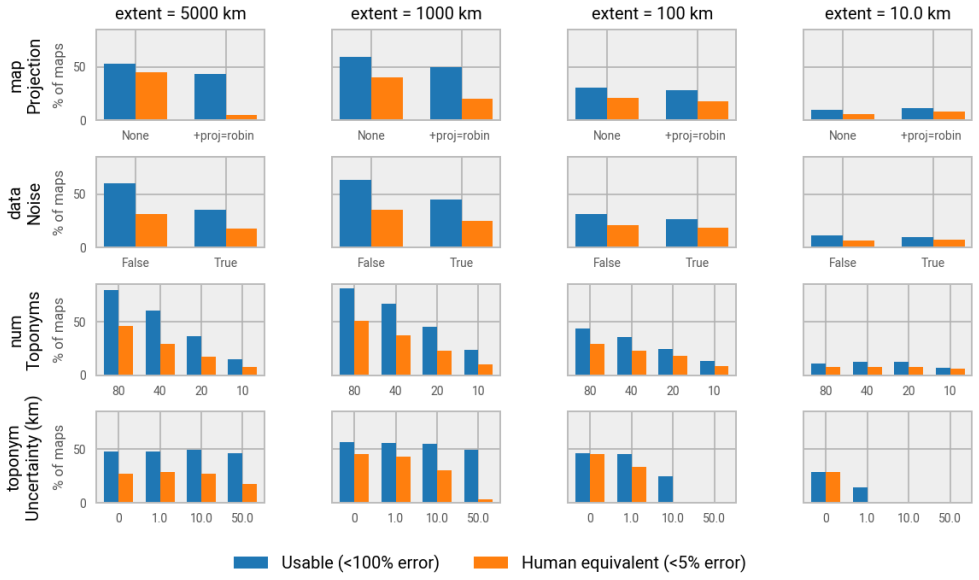

Figure 7. Selection of map parameters whose effect on georeferencing accuracy is mediated by map extent.

To better account for the possible effect of map parameter interactions, we additionally employ a logistic regression on the likelihood of georeferencing success to estimate the relative importance of each source of error (see Equation 1). This helps provide a general understanding of conditions under which the TAAG method is likely to succeed or fail – while controlling for the other map parameters. To clearly distinguish between success and non-success, we define success as human-equivalent accuracy (<5% error) and failure as any map that cannot be used (>100% error or failure), ignoring any of the edge cases in between. We also apply a min-max transformation on the regressors so that we can interpret the results coefficients as the log likelihood increase when going from each parameter's lowest to highest value. Results from this regression are presented in Table 5.

$$\begin{aligned} Success = &\beta_0 + \beta_1 * mapExtent + \beta_2 * imgResolution \\ &+ \beta_3 * numToponyms + \beta_4 * toponymDispersed \\ &+ \beta_5 * toponymUncertainty + \beta_6 * mapProjection \\ &+ \beta_7 * dataNoise + \beta_8 * metaNoise + \beta_9 * pixelNoise \end{aligned} \tag{1}$$

The findings from the regression analysis confirm several of the conclusions from

the initial analysis. Interpreting the coefficients as the log likelihood increase when going from the minimum to maximum value of each parameter, we see that the most influential parameters in the experimental setup for determining georeferencing success remains largely the same (number of toponyms, image resolution, map extent, toponym uncertainty, map projection, and data noise). One important finding is that the map extent loses importance when controlling for the other map parameters, a finding we address in more detail in the discussion section. All of the predictors except toponym dispersal are statistically significant.

Dep. Variable:	Success	**No. Observations:**		15983	
Model:	Logit	**Converged:**		True	
	coef	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
Constant	-1.4567	0.067	-21.813	0.000	-1.588	-1.326
mapExtent	0.5108	0.049	10.332	0.000	0.414	0.608
imgResolution	1.7603	0.062	28.372	0.000	1.639	1.882
numToponyms	2.7515	0.058	47.197	0.000	2.637	2.866
toponymUncertainty	-2.2483	0.071	-31.612	0.000	-2.388	-2.109
toponymDispersed	0.0506	0.042	1.206	0.228	-0.032	0.133
mapProjection	-1.2361	0.045	-27.294	0.000	-1.325	-1.147
dataNoise	-0.9041	0.043	-21.130	0.000	-0.988	-0.820
metaNoise	-0.3132	0.042	-7.441	0.000	-0.396	-0.231
pixelNoise	-0.5138	0.045	-11.400	0.000	-0.602	-0.425

Table 5. Logit Regression Results of Map Parameters on Georeferencing Success. Compares very high accuracy (<5% error) to very low accuracy (>100% or failure). Regressors are min-max normalized to help with interpretation.

4. Discussion & Conclusion

4.1. Discussion

Although we expected to see many of the findings from the previous section, the magnitude of their effects were sometimes lower than we anticipated. For instance, while we expected a major error source to be from inconsistencies across gazetteer sources regarding the true location of a particular toponym, we found that the algorithm was generally robust against toponym placement uncertainty (except at very small scales where small distances matter more). We also expected map projections such as World Robinson to distort the patterns formed between toponyms and make it harder to match with gazetteer latitude/longitude coordinates, but this only turned out to be true for very high accuracy map outcomes. Finally, image and noise related factors turned out less influential than anticipated, with the algorithm handling a wide variety of maps with complex data and metadata layers, including low-resolution maps and lossy file encodings.

One of the more key findings was the importance of map extents as a factor. The results we present in Table 5 and Figure 7 highlight how, as map extent decreases (i.e., the map becomes smaller in scope), so does the success of the TAAG approach. As this effect is less noticeable in the regression results (which controls for the num-

ber of toponyms), we believe this is likely attributable to the decreased likelihood of identifying a sufficiently large number of toponyms across small scales.

Taken as a whole, the findings from the experimental tests suggest that the TAAG algorithm is able to successfully georeference a large number of different types of maps, in many cases at an equivalent or better accuracy than manual human processing might produce. The algorithm is generally robust against poor map quality in terms of pixel resolution, file compression noise, map metadata such as titles and legends, as well as variations in map projections, and the distribution and uncertainty of toponyms. The number of available toponyms, and sufficiently large map extents (although mainly through its effect on the number of available toponyms) appear to be the most important parameters that affect the georeferencing accuracy, with other parameters, such as map projection and toponym uncertainty, having an impact under specific conditions.

4.2. Limitations

There are some important limitations to the suggested TAAG approach. Although many maps contain placename toponyms, the results of this paper show that the TAAG method is generally unsuitable for small scope maps - for instance where the map represents a city. While gazetteers do contain some information on places within cities, additional testing would need to be done to explore whether the method would work or could be modified to work for those cases. The algorithm is also not likely to work very reliably for pre-modern historical maps that use cursive text labels, since these are less likely to be picked up by the text recognition software. Lastly, although not a very common occurrence, rotated maps (facing away from north) would not currently be handled correctly, due to the way TAAG implements point pattern matching. These all represent pathways for improvement of the approach.

Although we sought to test the accuracy of the TAAG algorithm on a wide variety of different maps produced using a computer simulation based approach, real-world maps are likely to contain additional sources of heterogeneity that we were not able to capture in this paper. To further validate the accuracy of the proposed method, in future work we also intend to test it against a large map archive of real-world scanned maps. While this has the disadvantage of providing no information on absolute "truth", it has the advantage of providing a wider range of real-world heterogeneity the TAAG algorithm may be faced with.

4.3. Uses & Future Work

The toponym-based approach presented in this paper has many potential use-cases. First, it can be used by libraries or researchers to efficiently georeference vast collections of maps, using control point error metrics to flag for manual quality checks. Second, the general idea of toponym matching can be used to improve on manual workflows, by: a) providing an initial automated georeferencing which the user can either accept or edit; and b) limiting manual input to entering a toponym location and its text label. Third, TAAG's recognition of toponym patterns could also be used for non-traditional map cases, such as maps of unknown location (e.g. when there are no obvious landmarks or descriptions) or informal hand-drawn maps.[9]

[9]This last use-case could potentially be useful in situations where local geographic knowledge needs to be digitized, for instance through sketch-maps in disaster-situations or participatory mapping.

One important area for future work would be to further improve the accuracy and reduce the rates of failure. One such way would be to iteratively go through multiple stages of georeferencing to refine and improve the results, as some others have done (Pawlikowski et al., 2012; Weinman, 2013, 2017). For instance, after a first round of approximate georeferencing, the estimated map coordinates could then be used to limit the candidates to be searched for a more exhaustive and accurate matching in the next round. Additionally, given the approximate coordinates from the first round, the coordinates could be improved and additional matches added by comparing uncertain labels near known gazetteer coordinates (Gelbukh, Levachkine, & Han, 2003; Pawlikowski et al., 2012; Velázquez & Levachkine, 2003; Weinman, 2013).

Although the paper presents a complete pipeline for automated map georeferencing, the core contribution is the approach for identifying and matching toponyms to gazetteer coordinates. The approach is modular, so that any of the steps, such as image segmentation and text recognition, could be switched out with more optimal approaches to improve the overall results. For instance, the text detection accuracy could likely be improved by switching to the machine learning based text detection approaches of Weinman et al. (2019). Many alternative components are left to be tested.

4.4. Conclusion

In this paper we have presented the Toponym-based Approach to Automated Georeferencing (TAAG) algorithm. By providing a novel approach to the automated georeferencing of physical maps, TAAG represents a step forward towards more efficient map digitizing workflows, including large-scale digitization of large, ungeoreferenced catalogues of historic mapped information. To demonstrate that the method works, we tested the algorithm on thousands of different computer simulated maps, and show that TAAG was able to successfully automate the georeferencing of around 75% of maps it received with acceptable results, and in roughly 50% of cases achieved accuracies equivalent to – or better than – manual human georeferencing. While our test results highlight several areas of potential improvement to refine the TAAG approach, they also provide evidence of the wide range of map conditions under which TAAG can perform.

Acknowledgement(s)

The authors acknowledge William Mary Research Computing for providing computational resources and/or technical support that have contributed to the results reported within this paper. URL: https://www.wm.edu/it/rc

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was not supported by any grants.

Data availability statement

The data and software that support the findings of this study are available with the identifier(s) at the link: https://doi.org/10.5281/zenodo.3895087.

References

Arteaga, M. G. (2013). Historical map polygon and feature extractor. In *Proceedings of the 1st acm sigspatial international workshop on mapinteraction* (pp. 66–71).

Auffret, A. G., Kimberley, A., Plue, J., Skånes, H., Jakobsson, S., Walden, E., ... others (2017). Histmapr: Rapid digitization of historical land-use maps in r. *Methods in Ecology and Evolution*, *8*(11), 1453–1457.

Balletti, C. (2006). Georeference in the analysis of the geometric content of early maps. *e-Perimetron*, *1*(1), 32–39.

Bayer, T. (2016). Advanced methods for the estimation of an unknown projection from a map. *GeoInformatica*, *20*(2), 241–284.

Ben-Haim, G., Dalyot, S., & Doytsher, Y. (2014). Triangulation based topology approach for 2d point sets registration. *Survey Review*, *46*(338), 355–365.

Bentoutou, Y., Taleb, N., Kpalma, K., & Ronsin, J. (2005). An automatic image registration for applications in remote sensing. *IEEE transactions on geoscience and remote sensing*, *43*(9), 2127–2137.

Bolstad, P. (2014). Maps, data entry, editing, and output. In *Gis fundamentals: a first text on geographic information systems.* (4th ed., chap. 4). Eider Press, White Bear Lake, Minnesota.

Budig, B. (2018). *Extracting spatial information from historical maps: Algorithms and interaction.* BoD–Books on Demand.

Burt, J. E., White, J., & Allord, G. (2014). *Quad-g: Automated georeferencing of scanned map images - user manual, version 2.10* (Tech. Rep.).

Burt, J. E., White, J., Allord, G., Then, K. M., & Zhu, A.-X. (2019). Automated and semi-automated map georeferencing. *Cartography and Geographic Information Science*, 1–21.

Burt, J. E., White, J., Allord, G., Then, K. M., & Zhu, A.-X. (2020). Automated and semi-automated map georeferencing. *Cartography and Geographic Information Science*, *47*(1), 46–66.

Centeno, J. S. (1997). Segmentation of thematic maps using colour and spatial attributes. In *International workshop on graphics recognition* (pp. 221–230).

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

Chen, C.-C., Knoblock, C. A., Shahabi, C., Chiang, Y.-Y., & Thakkar, S. (2004). Automatically and accurately conflating orthoimagery and street maps. In *Proceedings of the 12th annual acm international workshop on geographic information systems* (pp. 47–56).

Chen, C.-C., Shahabi, C., & Knoblock, C. A. (2004). Utilizing road network data for automatic identification of road intersections from high resolution color orthoimagery. In *Stdbm* (Vol. 4, pp. 17–24).

Chiang, Y.-Y. (2010). *Harvesting geographic features from heterogeneous raster maps* (Unpublished doctoral dissertation).

Chiang, Y.-Y. (2016). Unlocking textual content from historical maps-potentials and applications, trends, and outlooks. In *International conference on recent trends in image processing and pattern recognition* (pp. 111–124).

Chiang, Y.-Y., & Knoblock, C. A. (2010). An approach for recognizing text labels in raster maps. In *2010 20th international conference on pattern recognition* (pp. 3199–3202).

Chiang, Y.-Y., & Knoblock, C. A. (2015). Recognizing text in raster maps. *Geoinformatica*, *19*(1), 1–27.

Chiang, Y.-Y., Leyk, S., & Knoblock, C. A. (2011a). Efficient and robust graphics recognition from historical maps. In *International workshop on graphics recognition* (pp. 25–35).

Chiang, Y.-Y., Leyk, S., & Knoblock, C. A. (2011b). 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* (pp. 1–4).

Chiang, Y.-Y., Moghaddam, S., Gupta, S., Fernandes, R., & Knoblock, C. A. (2014). From map images to geographic names. In *Proceedings of the 22nd acm sigspatial international conference on advances in geographic information systems* (pp. 581–584).

Cléry, I., Pierrot-Deseilligny, M., & Vallet, B. (2014). Automatic georeferencing of a heritage of old analog aerial photographs. *Photogrammetric Computer Vision*, *2*(3), 33.

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

Diez, Y., Lopez, M. A., & Sellares, J. A. (2008). Noisy road network matching. In *International conference on geographic information science* (pp. 38–54).

Dowman, I. (2005). Encoding and validating data from maps and images. In *Geographical information systems: Principles, techniques, management and applications (abridged edition)* (chap. 31). Wiley New York.

Duan, W., & Chiang, Y.-Y. (2018). Src: a fully automatic geographic feature recognition system. *SIGSPATIAL Special*, *9*(3), 6–7.

Ebi, N., Lauterbach, B., & Anheier, W. (1994). An image analysis system for automatic data acquisition from colored scanned maps. *Machine Vision and Applications*, *7*(3), 148–164.

Felus, Y. A., & Felus, M. (2009). On choosing the right coordinate transformation method. In *Proceedings of fig working week* (pp. 3–8).

Fleet, C., Kowal, K. C., & Pridal, P. (2012). Georeferencer: Crowdsourced georeferencing for map library collections. *D-Lib magazine*, *18*(11/12).

Fuse, T., Shimizu, E., & Morichi, S. (1998). A study on geometric correction of historical maps. *International Archives of Photogrammetry and Remote Sensing*, *32*, 543–548.

Gao, Y. (2017). *Analysis of coordinate transformation with different polynomial models* (B.S. thesis). University of Stuttgart, Institute of Geodesy.

Gelbukh, A., Levachkine, S., & Han, S.-Y. (2003). Resolving ambiguities in toponym recognition in cartographic maps. In *International workshop on graphics recognition* (pp. 75–86).

Gielsdorf, F., Gruendig, L., & Aschoff, B. (2003). Geo-referencing of analogue maps with special emphasis on positional accuracy improvement updates. In *Fig working week* (pp. 13–17).

Gonçalves, H., Gonçalves, J., Corte-Real, L., & Teodoro, A. (2012). Chair: automatic image registration based on correlation and hough transform. *International journal of remote sensing*, *33*(24), 7936–7968.

Gonçalves, H., Gonçalves, J. A., & Corte-Real, L. (2008). Automatic image registration based on correlation and hough transform. In *Image and signal processing for remote sensing xiv* (Vol. 7109, p. 71090J).

Gonçalves, H., Gonçalves, J. A., & Corte-Real, L. (2009). Measures for an objective evaluation of the geometric correction process quality. *IEEE Geoscience and Remote Sensing Letters*, *6*(2), 292–296.

Goodchild, M. J. (1991). The technological setting of gis. In D. J. Maguire, M. F. Goodchild, & D. W. Rhind (Eds.), *Geographical information systems: Principles and applications - volume 1: Principles* (p. 40-53). New York: Longman Scientific and Technical.

Hackeloeer, A., Klasing, K., Krisp, J. M., & Meng, L. (2014). Georeferencing: a review of methods and applications. *Annals of GIS*, *20*(1), 61–69.

Havlíček, J. (2016). Georeferencing applying weighting coefficients for least squares method adjustment. *International Multidisciplinary Scientific GeoConference: SGEM: Surveying Geology & mining Ecology Management*, *3*, 255–261.

Havlíček, J., & Cajthaml, J. (2014). The influence of the distribution of ground control points on georeferencing. *14th International Multidisciplinary Scientific Geoconference SGEM*,

965972.
Havlíček, J., & Cajthaml, J. (2015). Identification of a wrongly assigned ground control point. In *15th international multidisciplinary scientific geoconference sgem 2015 conference proceedings.*
Helleland, B. (2012). Place names and identities. *Oslo Studies in Language*, *4*(2).
Herold, H. (2017). *Geoinformation from the past: Computational retrieval and retrospective monitoring of historical land use.* Springer.
Herold, H., Roehm, P., Hecht, R., & Meinel, G. (2011). Automatically georeferenced maps as a source for high resolution urban growth analyses. In *Proceedings of the ica 25th international cartographic conference* (pp. 1–5).
Herrault, P.-A., Sheeren, D., Fauvel, M., & Paegelow, M. (2013). Automatic extraction of forests from historical maps based on unsupervised classification in the cielab color space. In *Geographic information science at the heart of europe* (pp. 95–112). Springer.
Hild, H., & Fritsch, D. (1998). Integration of vector data and satellite imagery for geocoding. *International Archives of Photogrammetry and Remote Sensing*, *32*, 246–251.
Hill, L. L. (2009). *Georeferencing: The geographic associations of information.* Mit Press.
Hunziker, P., & Cederman, L.-E. (2017). No extraction without representation: The ethno-regional oil curse and secessionist conflict. *Journal of Peace Research*, *54*(3), 365–381.
Inoue, R., Wako, M., & Shimizu, E. (2007). A new map transformation method for highly deformed maps by creating homeomorphic triangulated irregular network. In *Xxiii international cartographic conference. moscow, russia. dvd.*
Jackson, M. J., & Woodsford, P. A. (1991). Gis data capture hardware and software. In D. J. Maguire, M. F. Goodchild, & D. W. Rhind (Eds.), *Geographical information systems: Principles and applications - volume 1: Principles* (p. 239-248). New York: Longman Scientific and Technical.
Jatnieks, J. (2010). Extended poster abstract: open source solution for massive map sheet georeferencing tasks for digital archiving. In *International conference on asian digital libraries* (pp. 258–259).
Jenny, B., & Hurni, L. (2011). Studying cartographic heritage: Analysis and visualization of geometric distortions. *Computers & Graphics*, *35*(2), 402–411.
Joo, J., & Steinert-Threlkeld, Z. C. (2018). Image as data: Automated visual content analysis for political science. *arXiv preprint arXiv:1810.01544*.
Karel, W., Doneus, M., Briese, C., Verhoeven, G., & Pfeifer, N. (2014). Investigation on the automatic geo-referencing of archaeological uav photographs by correlation with pre-existing ortho-photos. In *Isprs technical commission v symposium* (pp. 307–312).
Knowles, A. K., & Hillier, A. (2008). *Placing history: how maps, spatial data, and gis are changing historical scholarship.* ESRI, Inc.
Knutzen, M. A. (2013). Unbinding the atlas: Moving the nypl map collection beyond digitization. *Journal of Map & Geography Libraries*, *9*(1-2), 8–24.
Lawrence, R. L., Means, J. E., & Ripple, W. J. (1996). An automated method for digitizing color thematic maps. *Photogrammetric engineering and remote sensing*, *62*, 1245–1248.
Leyk, S., & Boesch, R. (2010). Colors of the past: color image segmentation in historical topographic maps based on homogeneity. *GeoInformatica*, *14*(1), 1.
Leyk, S., & Chiang, Y.-Y. (2016). Information extraction based on the concept of geographic context. In *Proc. autocarto* (pp. 100–110).
Li, C., Shen, Y., Li, B., Qiao, G., Liu, S., Wang, W., & Tong, X. (2014). An improved geopositioning model of quickbird high resolution satellite imagery by compensating spatial correlated errors. *ISPRS journal of photogrammetry and remote sensing*, *96*, 12–19.
Li, Y., & Briggs, R. (2006). Automated georeferencing based on topological point pattern matching. In *The international symposium on automated cartography (autocarto), vancouver, wa.*
Li, Y., & Briggs, R. (2008). Scalable and error tolerant automated georeferencing under affine transformations. In *Igarss 2008-2008 ieee international geoscience and remote sensing symposium* (Vol. 5, pp. V–232).

Li, Y., & Briggs, R. (2012). An automated system for image-to-vector georeferencing. *Cartography and Geographic Information Science*, *39*(4), 199–217.

Liew, L. H., Wang, Y. C., & Cheah, W. S. (2012). Evaluation of control points' distribution on distortions and geometric transformations for aerial images rectification. *Procedia Engineering*, *41*, 1002–1008.

Liu, D., Toman, E., Fuller, Z., Chen, G., Londo, A., Zhang, X., & Zhao, K. (2018). Integration of historical map and aerial imagery to characterize long-term land-use change and landscape dynamics: An object-based analysis via random forests. *Ecological indicators*, *95*, 595–605.

Liu, T., Miao, Q., Tian, K., Song, J., Yang, Y., & Qi, Y. (2016). Sctms: Superpixel based color topographic map segmentation method. *Journal of Visual Communication and Image Representation*, *35*, 78–90.

Liu, T., Xu, P., & Zhang, S. (2019). A review of recent advances in scanned topographic map processing. *Neurocomputing*, *328*, 75–87.

Long, T., Jiao, W., He, G., & Zhang, Z. (2016). A fast and reliable matching method for automated georeferencing of remotely-sensed imagery. *Remote sensing*, *8*(1), 56.

Mahmoud, M. I. (2012). *Information extraction from paper maps using object oriented analysis ooa.* (Unpublished master's thesis). University of Twente Faculty of Geo-Information and Earth Observation (ITC).

Martins, B., Borbinha, J., Pedrosa, G., Gil, J., & Freire, N. (2007). Geographically-aware information retrieval for collections of digitized historical maps. In *Proceedings of the 4th acm workshop on geographical information retrieval.*

McDonald, J., et al. (1997). History of ohio's oil-and gas-well location maps and their conversion to digital form.

Mendes, J. G. (1995). Cost estimation for the conversion of map-based land-use plans into digital gis databases. *Computers, environment and urban systems*, *19*(2), 99–105.

Miao, Q., Xu, P., Li, X., Song, J., Li, W., & Yang, Y. (2017). The recognition of the point symbols in the scanned topographic maps. *IEEE Transactions on Image Processing*, *26*(6), 2751–2766.

Michelson, M., Goel, A., & Knoblock, C. A. (2008). Identifying maps on the world wide web. In *International conference on geographic information science* (pp. 249–260).

Muller-Crepon, C., Hunziker, P., & Cederman, L.-E. (2019). Roads to rule, roads to rebel: Relational state capacity and conflict in africa..

Musavi, M. T., Shirvaikar, M. V., Ramanathan, E., & Nekovei, A. (1988). A vision based method to automate map processing. *Pattern Recognition*, *21*(4), 319–326.

Panagos, P., Jones, A., Bosco, C., & Kumar, P. S. (2011). European digital archive on soil maps (eudasm): preserving important soil data for public free access. *International Journal of Digital Earth*, *4*(5), 434–443.

Pawlikowski, R., Ociepa, K., Markowska-Kaczmar, U., & Myszkowski, P. B. (2012). Information extraction from geographical overview maps. In *International conference on computational collective intelligence* (pp. 94–103).

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

Pouderoux, J., Gonzato, J. ., Pereira, A., & Guitton, P. (2007, Sep.). Toponym recognition in scanned color topographic maps. In *Ninth international conference on document analysis and recognition (icdar 2007)* (Vol. 1, p. 531-535).

Reiher, E., Li, Y., Delle Donne, V., Lalonde, M., Hayne, C., & Zhu, C. (1996). A system for efficient and robust map symbol recognition. In *Proceedings of 13th international conference on pattern recognition* (Vol. 3, pp. 783–787).

Roy, P. P., Llados, J., & Pal, U. (2007). Text/graphics separation in color maps. In *2007 international conference on computing: Theory and applications (iccta'07)* (pp. 545–551).

Roy, P. P., Vazquez, E., Lladós, J., Baldrich, R., & Pal, U. (2007). A system to segment text and symbols from color maps. In *International workshop on graphics recognition* (pp.

245–256).

Rumsey, D., & Williams, M. (2002). *Historical maps in gis.* na. Retrieved from `https://www.davidrumsey.com/gis/ch01.pdf`

Rus, I., Balint, C., Craciunescu, V., Constantinescu, S., Ovejanu, I., & Bartos-Elekes, Z. (2010). Automated georeference of the 1: 20 000 romanian maps under lambert-cholesky (1916–1959) projection system. *Acta Geodaetica et Geophysica Hungarica*, *45*(1), 105–111.

Saalfeld, A. (1988). Conflation automated map compilation. *International Journal of Geographical Information System*, *2*(3), 217–228.

Samet, H., & Soffer, A. (1996). Marco: Map retrieval by content. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(8), 783–798.

Samet, H., & Soffer, A. (1998). Magellan: Map acquisition of geographic labels by legend analysis. *International journal on document analysis and recognition*, *1*(2), 89–101.

Schölzel, C. A., Hense, A., Hübl, P., Kühl, N., & Litt, T. (2002). Digitization and georeferencing of botanical distribution maps. *Journal of Biogeography*, *29*(7), 851–856.

Sharma, G., Wu, W., & Dalal, E. N. (2005). The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur*, *30*(1), 21–30.

Sisman, A. (2014). An experimental design approach on georeferencing. *Boletim de Ciências Geodésicas*, *20*(3), 548–561.

Sisman, Y., & Sisman, A. (2017). The factors optimization on georeferencing analogue maps. *Arabian Journal for Science and Engineering*, *42*(6), 2471–2478.

Smith, R. (2007). An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (icdar 2007)* (Vol. 2, pp. 629–633).

Song, W., Keller, J. M., Haithcoat, T. L., & Davis, C. H. (2008). Automated geospatial conflation of vector road maps to high resolution imagery. *IEEE Transactions on image processing*, *18*(2), 388–400.

Statuto, D., Cillis, G., & Picuno, P. (2017). Using historical maps within a gis to analyze two centuries of rural landscape changes in southern italy. *Land*, *6*(3), 65.

Stavropoulou, G. (2014). Optical character recognition on scanned maps for information extraction and automated georeference.

Tan, C. L., & Ng, P. (1998). Text extraction using pyramid. *Pattern Recognition*, *31*(1), 63–72.

Tan, Q., Lu, N., Dong, M., & Zhu, L. (2013). Influence of geometrical distribution of common points on the accuracy of coordinate transformation. *Applied Mathematics and Computation*, *221*, 411–423.

Thieler, E. R., & Danforth, W. W. (2012). Historical shoreline mapping (i): improving techniques and reducing positioning errors. *Journal of Coastal Research*, *10*(3).

Titova, O., & Chernov, A. (2009). Method for the automatic georeferencing and calibration of cartographic images. *Pattern Recognition and Image Analysis*, *19*(1), 193–196.

Uhl, J., Leyk, S., Chiang, Y.-Y., Duan, W., & Knoblock, C. (2018). Map archive mining: visual-analytical approaches to explore large historical map collections. *ISPRS international journal of geo-information*, *7*(4), 148.

Uhl, J. H., Leyk, S., Chiang, Y.-Y., Duan, W., & Knoblock, C. A. (2017). Extracting human settlement footprint from historical topographic map series using context-based machine learning.

UNSD. (1996). *Mapscan for windows software package for automatic map data entry - user's guide and reference manual* (Tech. Rep.). United Nations, Department for Economic and Social Affairs - Statistics Division.

Velázquez, A., & Levachkine, S. (2003). Text/graphics separation and recognition in raster-scanned color cartographic maps. In *International workshop on graphics recognition* (pp. 63–74).

Waters, T. (2017). *Map warper.* Retrieved from `https://mapwarper.net/`
Weinman, J. (2013). Toponym recognition in historical maps by gazetteer alignment. In *2013 12th international conference on document analysis and recognition* (pp. 1044–1048).
Weinman, J. (2017). Geographic and style models for historical map alignment and toponym recognition. In *2017 14th iapr international conference on document analysis and recognition (icdar)* (Vol. 1, pp. 957–964).
Weinman, J., Chen, Z., Gafford, B., Gifford, N., Lamsal, A., & Niehus-Staab, L. (2019). Deep neural networks for text detection and recognition in historical maps. In *2019 international conference on document analysis and recognition (icdar)* (pp. 902–909).
White, D., Corson-Rikert, J., & Maizel, M. (1987). Wysiwyg'map digitizing: Real time geometric correction and topological encoding. In *In proceedings from auto-carto 8 (bethesda, md: American congress on surveying and mapping.*
Wong, A., & Clausi, D. A. (2007). Arrsi: Automatic registration of remote-sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, *45*(5), 1483–1493.
Wu, X., Carceroni, R., Fang, H., Zelinka, S., & Kirmse, A. (2007). Automatic alignment of large-scale aerial rasters to road-maps. In *Proceedings of the 15th annual acm international symposium on advances in geographic information systems* (p. 17).
Wucherpfennig, J., Weidmann, N. B., Girardin, L., Cederman, L.-E., & Wimmer, A. (2011). Politically relevant ethnic groups across space and time: Introducing the geoepr dataset. *Conflict Management and Peace Science*, *28*(5), 423–437.
Yilmaz, I., & Gullu, M. (2012). Georeferencing of historical maps using back propagation artificial neural network. *Experimental Techniques*, *36*(5), 15–19.

Appendix A. Image Layout Segmentation

In this appendix, we provide detail on the algorithm used to segment the image into areas which contain either map data or other, descriptive data (e.g., the title). Since we are only interested in relatively large-scale image elements (e.g., the map data region of the image), we first downscale the image to some fixed dimension in order to efficiently handle very high-resolution map images. Next, we detect the edges of image objects based on whether each pixel is perceptibly different from the colors of its immediate neighbouring pixels. More formally this means that a pixel is considered an edge if least one neighbouring pixel in a 5x5 moving window has a CIELab Delta E 2000 color difference value of more than 10 (we describe this metric in more detail in the section on text recognition). Having identified the edge pixels in the image, we can use conventional contour tracing algorithms to trace the outlines of objects in the image.

If the image consists of a map region surrounded by an outer margin, then the inner map region can be identified as the largest contour in the image that is larger than some threshold (e.g. at least one-third the size of the image) and leaves some room around the edge of the image. Text boxes such as legends and titles can be identified by rectangular contours of a particular size (e.g. between one-sixteenth and half the size of the image). The parts of the image that constitute the map region and text boxes are ignored in the later stages when determining which text labels are toponyms. The map region is further used to crop away the margins in the final georeferenced map output.

Appendix B. Automated Text Parsing

This appendix provides full information for the replication of our map-based text parsing procedures, which consists of two stages.

B.1. Extracting Text from Color Pixels

In the first stage, the goal is to identify text strings from the image. This involves (a) the identification of the color of the text, (b) the use of this information to define some pixels as "text" and other pixels as "not text", and (c) the application of optical character recognition to the output text pixels.

First, we must establish a color to use for identifying text in the map. This is important because the map can contain several possible text colors, and the original color of the text can be distorted due to such factors as image quality and blending with background colors. Previous research has explored various alternatives for choosing the text color, including manually specifying colors (Chiang & Knoblock, 2010; Ebi et al., 1994), or semi-automated color detection from user-provided label examples (Chiang & Knoblock, 2015; Chiang, Moghaddam, Gupta, Fernandes, & Knoblock, 2014).

In this paper we implement a new approach that automatically searches the image for samples of labels and extracts the colors from those samples. During this initial sampling stage, we assume that text is generally darker than their surroundings and use the luminance band of the CIELab color space as the basis for an initial OCR text detection. While such grayscale text detection won't pick up all instances of text, we only need a few examples of text to make a determination. We can do this efficiently by sampling small sub-regions of the image until some number of example texts have been identified.

To extract the color from the sampled text labels, we first determine the pixels that make up the text characters based on their luminance level. The weighted average pixel color in the region of each sampled text can then be calculated, using each pixel's luminance level as weights (darker pixels count more towards the average color).[10] We also capture the pixel-level color deviation from the calculated sample color using the same weighted average approach. Color difference is calculated based on the CIELab Delta E 2000 color difference metric ($\triangle E$), a metric that is based on how humans perceive colors (see Sharma, Wu, & Dalal, 2005).

Second, based on the detected sample colors, we then identify the main text colors by grouping together those sample colors that are perceptually similar using the Delta E distance metric (e.g. $\triangle E < 15$). This procedure results in one or more distinct color groups, and we can use the RGB colors that internally minimizes $\triangle E$ to represent each group as a single text color μ. We also record each group's internal color variability δ as the sum of the average $\triangle E$ color deviation from the group color and the average of the previously calculated pixel color deviations.

Once the text colors have been estimated, we employ a second algorithm that searches across the entire image to define some pixels as "text". For each text color μ, we use $\triangle E(\mu) < \delta$ to isolate the relevant pixels of that color, where δ is the previously calculated color variability. Only the pixels below this threshold are kept, and we use the color difference values (scaled to range between 0 and 127) to produce an image of grayscale text pixels to be processed by the OCR engine.

[10] We also apply histogram equalization on the luminance band and focus only on the top 33% most luminant pixels to minimize the influence of peripheral non-text pixels

Third, once text pixels are identified we use a popular open-source neural network based OCR engine, optimized for sparse text recognition (for more detailed information on the OCR tool, see Smith, 2007).[11] This results in a list of identified words and their coordinates, width, height, and confidence level. The text recognition is likely to contain several errors, so we clean the results by dropping text recognized with a low confidence probability ($< 60\%$), and graphics potentially misidentified as text (often recognized as single-character text, and numeric and non-alphabetic characters). To avoid duplicates where multiple text colors pick up some of the same pixels and map labels, we find duplicate texts whose bounding boxes partially overlap and choose the text for the color that best matches the underlying pixel colors (lowest average pixel-level $\triangle E$).

B.2. Connecting text strings

In the second stage, once individual text strings have been extracted from the image, these have to be grouped together, such as multi-part place names. We implement a rules-based algorithm, similar to that proposed in Pouderoux, Gonzato, Pereira, and Guitton (2007) (for alternative approaches, see Chiang, 2010; Chiang & Knoblock, 2010, 2015; Chiang et al., 2011b, 2014; Gelbukh et al., 2003; C. L. Tan & Ng, 1998). First, we group together words horizontally if they are within 1.5 units of distance, where a unit of distance is equal to the estimated font height, as well as having approximately the same font height and vertical alignment. In the second step of the algorithm, we vertically group together text lines that are within 1 times the font height above or beneath. Grouping is done separately for each text color.

Appendix C. Symbol Identification

In the following appendix we provide additional information on the algorithm for identifying the symbol (e.g. a circle or square) marking the location of a toponym. We do this by identifying clusters of black pixels in the vicinity of 1.5 times the toponym font height. To account for potential gaps in hollow shape outlines, we perform a morphological closing operation that fills the interior of any closed chain of black pixels, up to a size approximately one-third the height of the text font. To separate the symbol from potential lines that intersect it, we perform a morphological opening operation that disconnects it from any lines thinner than approximately one-fourth the font size.

Next, we extract the contours of any remaining clusters of black pixels. To further subset these clusters, while allowing for variation in terms of shape type and pixel noise, we identify toponym symbols based on a simple set of criteria. The width and height dimensions of a toponym marker are expected to be approximately square and be of a size that matches the size of the text label. We therefore only keep those contours that have an aspect ratio between 0.75 and 1.25, and whose size is between one-fourth and 1.5 times the font height. We also require that the contour fills more than one-fourth the area of its bounding box. The largest of any remaining contours in the toponym's vicinity is then chosen as the symbol marker for that toponym, and the center point of the contour shape is used as the final toponym coordinate. If no symbol

[11] Upscaling the image with a Lanczos filter prior to thresholding and text recognition, drastically improves the text detection results for small, low-resolution images.

is found around a text label, we assume the label does not represent a toponym.

Appendix D. Point Pattern Matching

Here, we provide more information on how we implement the point pattern matching for the resolution of toponym coordinates. In the approach proposed in this paper, we use an algorithm for point set comparisons of normalized coordinates, assuming all point sets have roughly the same northward-orientation, as is the case for most maps.[12] Consider that the image and text recognition has identified N toponyms in the source image: $\theta_{i=1}, \theta_2, ..., \theta_{i=N}$. In image space, each toponym θ_i has a corresponding pixel coordinate x_i, y_i, and the combined set of these image coordinates connect to form a point pattern: ϕ. Searching a gazetteer for a toponym θ_i will result in M_i possible matching names: $\hat{\theta}_{i,j=1}, \hat{\theta}_{i,2}, ..., \hat{\theta}_{i,j=M}$, with the corresponding geographic coordinates given as $\hat{x}_{i,j}, \hat{y}_{i,j}$.

Each possible match $\hat{\theta}_{i,j}$ forms a point pattern $\hat{\phi}_z$ with every other possible $\hat{\theta}_{i,j}$, where z represents each possible combination. To find the correct matching point set, we iterate over the candidate point patterns in geographic space ($\hat{\phi}_z$), and contrast these to the original point pattern in image space (ϕ). The candidate point pattern $\hat{\phi}_z$ that most resembles the original point pattern ϕ is used as the final set of geographic coordinates $\hat{x}_{i,j}, \hat{y}_{i,j}$ for the toponyms in the image. This is illustrated in the earlier example of the three towns in northern Cameroon, where the triangle shape formed by the correct set of gazetteer coordinates (Figure 3d), is the one that most resemble the triangle shape of the towns in the original image (Figure 3a). We next go through some of the technical details of how we implemented this approach.

D.1. Map Unit Normalization

In order to compare the point patterns of the image points and geographic coordinates, which are given in different units (pixels and decimal degrees), we must first convert them to a common coordinate system. This is done by normalizing their coordinates as values ranging from 0 to 1 between the minimum and maximum x and y coordinates for each point pattern (Figure 3c). To preserve the aspect ratio of the point patterns, the longest of the x or y axis will extend to a maximum of 1, while the shortest axis will only range to some fraction of 1 depending on the ratio between the longest and shortest axes. When normalizing image coordinates, the y-coordinate additionally has to be flipped to account for the fact that the y-axis of image coordinates goes in the opposite direction of geographic coordinates. This approach retains the information in the shapes that are most relevant for the algorithm presented here - relative coordinate positions. This normalization is summarized in Algorithm 1:

D.2. Match Selection Procedure

Having standardized the point pattern coordinates following Algorithm 1, the normalized image pattern (ϕ) is then contrasted to the normalized pattern of each possible combination of matches ($\hat{\phi}_z$) to identify the most likely set of candidate matching

[12]Other approaches are also possible, including the use of scale- and rotation-invariant transformations to find matches from arbitrary viewpoints (Chen et al., 2008; Chen, Knoblock, et al., 2004; Weinman, 2013)

```
Data: A set of points with image or geographic coordinates
Result: The same points with normalized coordinates
xmax,xmin,ymax,ymin ← getBounds(points)
longSide = max(xmax - xmin, ymax - ymin) ;
newpoints ← []
for x,y in points do
    x = (x - xmin) / longSide ;
    y = (y - ymin) / longSide ;
    if imageflag then
        y = 1 - y;
    end
    newpoints.append((x,y)) ;
end
```

Algorithm 1: Coordinate normalization of a point pattern.

points. We do this following a simple metric of pattern similarity, as the average relative distance between coordinates in normalized space:

$$\triangle\hat{\phi}_z = \sum_{i=0,j}^{N} \sqrt{(x_i - \hat{x}_{i,j})^2 + (y_i - \hat{y}_{i,j})^2}/N \tag{D1}$$

where N is the number of toponyms, x_i, y_i is the normalized image coordinate for the toponym at i, and $\hat{x}_{i,j}, \hat{y}_{i,j}$ is the normalized geographic coordinate of the j'th possible match in a particular point pattern combination. The combination of match candidates with the lowest $\triangle\hat{\phi}_z$ contains the potential match points whose pattern most resembles the point pattern in the original image, and is chosen to determine the geographic coordinates $\hat{x}_{i,j}, \hat{y}_{i,j}$ for our map image control points.

In some rare cases, since our normalized point patterns don't carry any information about scale or absolute distances, we may end up matching the wrong point set - especially for smaller point sets, like the example of the three towns mentioned in the main text. However, in practice, we are interested in matching more complex point patterns with N points (where N is the number of toponyms identified in the map). As N grows, the chance for spurious matches decreases to very low levels. Further, by setting an arbitrary pattern similarity threshold σ (e.g. $\sigma = 0.10$), we can preclude matched cases that would fall below meaningful levels of similarity: $\triangle\hat{\phi}_z < \sigma$ (i.e., discluding cases in which the difference between the point patterns in image space and projected space is too large, indicating a potentially spurious match).

D.3. Combinatorial Optimization

Finding the optimal pattern match with the lowest $\triangle\hat{\phi}_z$ requires testing all possible combinations of coordinates for every toponym. Specifically, the number of combinations for which we have to calculate the pattern similarity metric is equal to the sum product of the number of possible coordinates of each toponym:

$$O_{full} = O(\prod_{i=0}^{N} M_i) = O(M_{i=0} * M_{i+1} * ... * M_N) \tag{D2}$$

where M_i is the number of possible matches for toponym θ_i, and N is the number of toponyms. This means that the computational cost grows exponentially as N increases. For instance, if there are exactly 5 possible coordinates for every toponym, then the performance for a mere 10 toponyms is $O(5^{10}) \approx O(10 * 10^6)$. For 20 placenames this goes up to $O(5^{20}) \approx O(95 * 10^{12})$. Most maps will likely have on the order of 30-40 placenames, resulting in an algorithm that would only be practical in a cluster environment if an exhaustive search was used.

One way to solve the combinatorial problem has been suggested in the georeferencing of road intersections in remote sensing imagery, whereby the reference road network dataset is transformed during a pre-processing stage to enable faster pattern matching (Y. Li & Briggs, 2006, 2008, 2012). However, these matching solutions tend to be optimized for the relatively simple transformations required for remote sensing imagery, and the pre-processing techniques can take a significant amount of space and processing time (e.g. in the case of Y. Li and Briggs (2006), 13 minutes for just 10,000 points). Other approaches used to reduce the number of match comparisons in map georeferencing include DBScan point clustering (Pawlikowski et al., 2012) and probability-based RANSAC matching (Weinman, 2013).

In this paper, we focus on a piece-wise, growth-based method, similar to more general aggregative voting algorithms for matching triangulation structures (Ben-Haim, Dalyot, & Doytsher, 2014). The proposed method consists of five steps:

Step 1 Construct a set **C**, which is made up of all possible combinations of three non-repeating, unsorted elements from the set of all toponyms **T**. For instance, in the case of four toponyms, set **C** would contain three possible combinations:

$$C = \{\{\theta_1, \theta_2, \theta_3\}, \{\theta_1, \theta_2, \theta_4\}, \{\theta_2, \theta_3, \theta_4\}\} \tag{D3}$$

Step 2 Select one of the triplets from set **C**. The goal is then to identify the optimal geographic coordinates - as described in the previous section - for these three toponyms, i.e. the combination of matched toponyms $\theta_{i,j}$ whose coordinates best matches the original 3-point pattern in the image. The performance for finding the optimal set of coordinates for these three toponyms is:

$$O_{triplet} = O(\prod_{k=1}^{3} M_k) \tag{D4}$$

where M_k is the number of possible matches for each toponym θ_k in the triplet. If the optimal combination of coordinates for the triplet satisfies some minimum threshold of similarity σ, we add the triplet to a result set **R** and proceed to step 3. If not, we repeat step 2 with the next triplet of **C**.

Step 3 We remove from the list of toponyms **T**, any toponym that has already been matched. We then consider any of the toponyms θ_i that remain to be matched from **T**, and determine its geographic coordinate based on the match candidate $\hat{\theta}_{i,j}$ that

best improves the pattern similarity score $\triangle\hat{\phi}_z$ of the result set **R**. If the similarity score for the optimal coordinate remains under the similarity threshold σ, we add this toponym to the result set **R** - e.g. changing it from the original 3-point triplet to a 4-point set.

Step 4 Repeat step 3 until all - or as many as possible - toponyms θ_i in set **T** are associated with a matching toponym $\hat{\theta}_{i,j}$ in the result set **R**. The added cost of expanding the initial result triplet **R** with all the remaining elements in **T** is then only:

$$O_{expand} = \sum_{i=0}^{|T|} O(M_i) \tag{D5}$$

where $|T|$ is the length of the remaining set of toponyms, and M_i is the number of possible matches for each toponym θ_i that remains. The expanded result set **R** can be used as the basis for the control points.

Step 5 Repeat steps 1 to 4 a given number of trials τ. This is necessary in order to avoid cases where the initial triplet added to the result set **R** in step 2 was based on a spurious match. In order avoid such cases and find the optimal set of control points we run τ repetitions of steps 1 to 4, and choose only the result set **R** with the highest match similarities and that expands to some minimum number of λ matches.

While this approach is not guaranteed to find the optimal match - in contrast to the the full exhaustive search which iterates through all possible combinations of place name coordinates - it results in a tradeoff that drastically reduces the computational cost while still finding a matching combination that meets some threshold. This threshold is determined by hyperparemeters that define a minimum similarity σ, the minimum number of matches λ, and the number of trials τ.

Appendix E. Selection of Image Transformation Algorithm

E.1. Choosing the transform function

After identifying a set of control points, the next challenge for automated georeferencing is in how to choose the optimal polynomial transform function. In traditional map georeferencing it is up to the human coder to qualitatively accept or reject a transform function by visually comparing the accuracy of a transformed map with an underlying basemap (Bolstad, 2014, 156). Visual aids can also be used to help in the decision making, such as control point displacement arrows or distortion grids (Jenny & Hurni, 2011; J. Uhl et al., 2018).

Instead, we need a way for the algorithm to quantify the accuracy of the model. The typical way to quantify model accuracy is using the root mean square error (RMSE) of the control point residuals (Bolstad, 2014, 155):

$$RMSE = \sqrt{\sum_{i=1}^{N} dist(p_i, p_i')/N} \tag{E1}$$

where N is the number of control points; p_i and p_i' are the observed and predicted

coordinates (given in the unit of choice, as either pixel or geographic coordinates) of control point i; and *dist* is the function used to calculate the distance between the two coordinates. The RMSE measures the degree to which the transform function predictions deviates from the observed control points, with a squared term that penalizes large deviations.

The traditional RMSE metric however is not suitable for comparing across different models, since higher-order polynomial transforms by definition fits the model closer to the control points and therefore result in lower control point residuals (Bolstad, 2014, 158). A lower RSME value only means that the error near the control points are low, but this often comes at the cost of higher errors further away from the control points, depending on the number and distribution of control points (Bolstad, 2014; Dowman, 2005; Felus & Felus, 2009; Gao, 2017).

Several alternative model accuracy metrics have been suggested in the literature (Felus & Felus, 2009; Gonçalves et al., 2009). A particularly promising suggestion is to switch away from using model-fit residuals - which measures errors *at* the control points - towards leave-one-out residuals - which measures out-of-sample errors *between* the control points. This is done by withholding one control point at a time from the model estimation stage, and calculating the residual for each point being withheld (Bolstad, 2014, 159; Havlíček & Cajthaml, 2014, 5). After comparing a wide variety of error metrics, Gonçalves et al. (2009) recommended that the literature switch to this leave-one-out method. In our approach, we therefore use an RMSE metric based on leave-one-out residuals, $RMSE_{LOO}$, which is more suitable for comparing and choosing between the accuracy of different types of transforms.

E.2. Control point outliers

Another challenge for automated georeferencing is that estimating transform functions is highly sensitive to outliers, where a single faulty control point can negatively affect the accuracy of the transform. In traditional map georeferencing it is up to the human coder to identify an acceptable level of residual error, and to identify and drop potential outliers (Bolstad, 2014, 156). One approach to automate this would be to drop any points beyond a specified number of standard deviations from the mean (Havlíček & Cajthaml, 2015). When the quality of control points cannot be guaranteed - such as satellite and drone image georeferencing - more sophisticated techniques for refining the list of control points include RANSAC and related matching techniques (Long et al., 2016; Weinman, 2013).

In this paper, we use a simple iterative approach where we drop one control point at a time. To determine which control point to drop, we go through each of the control points and estimate the $RSME_{LOO}$ model error that would result if that point were to be dropped. The point whose exclusion best improves the model (results in the lowest model error) is then dropped, provided the new model improves above some threshold (e.g. $> 10\%$). We keep dropping control points until the model stops improving beyond the specified threshold. Dropping any further control points beyond this point would only marginally improve or even decrease the accuracy of the model (i.e. this is the inflection point where *adding* any more points would stop improving the model). By applying this method we can calculate the optimal set of control points for each of the transform functions being compared, and can choose the optimal transform function based on this.

After dropping outlier control points for each model and choosing the best perform-

ing transform function, we can finally transform and resample the map image to a geographic coordinate system (although this step is optional). First, the transform is used to calculate the bounds and aspect ratio of the transformed (geographic) coordinate system. These are also used to define a new image grid with the same aspect ratio. Second, for each geographic coordinate in the new georeferenced image we use the inverse transform function to determine the corresponding pixel(s) in the original image used for resampling. This step completes the proposed method and gives us the final output: a georeferenced image based on a fully automated process that requires no manual intervention.