Extraction of Ancient Map Contents Using Trees of Connected Components

Jordan Drapeau[1(✉)], Thierry Géraud[3(✉)], Mickaël Coustaty[1(✉)], Joseph Chazalon[1,3(✉)], Jean-Christophe Burie[1(✉)], Véronique Eglin[2(✉)], and Stéphane Bres[2(✉)]

[1] Laboratoire L3i, University of La Rochelle, 17042 La Rochelle Cedex 1, France
{jordan.drapeau,mickael.coustaty,joseph.chazalon,
jean-christophe.burie}@univ-lr.fr
[2] Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, 69621 Lyon, France
{veronique.eglin,stephane.bres}@insa-lyon.fr
[3] EPITA Research and Development Laboratory (LRDE), Le Kremlin-Bicetre, France
{thierry.geraud,joseph.chazalon}@lrde.epita.fr

Abstract. Ancient maps are an historical and cultural heritage widely recognized as a very important source of information, especially for dialectological researches, the cartographical heritage produces first-rate data. However, exploiting such maps is a quite difficult task to achieve, and we are focusing our attention on this major issue. In this paper, we consider the Linguistic Atlas of France (ALF), built between 1902 and 1910 and we propose an original approach using tree of connected components for the separation of the content in layers for facilitating the extraction, the analysis, the viewing and the diffusion of the data contained in these ancient linguistic atlases.

Keywords: Mathematical morphology · Connected components
Map analysis · Text/Graphics separation · Linguistic Atlas

1 Introduction

Ancient maps are a historical and cultural heritage widely recognized as a very important source of information, but not easy to use. In this paper, we are focusing on the Linguistic Atlas of France (ALF), which is a collection of maps in paper format[1]. It comprises 35 booklets, bringing together in 12 volumes, 1920 geolinguistic maps presenting an instantaneous picture of the dialect situation of France at the end of the 19th century. It can be defined as a first-generation atlas publishing raw data and constituting a corpus of more than one million of reliable lexical data, homogeneously transcribed, using the Rousselot-Gilliéron phonetic alphabet.

[1] Dataset available at http://lig-tdcge.imag.fr/cartodialect3/carteTheme.

A. Fornés and B. Lamiroy (Eds.): GREC 2017, LNCS 11009, pp. 115–130, 2018.
https://doi.org/10.1007/978-3-030-02284-6_9

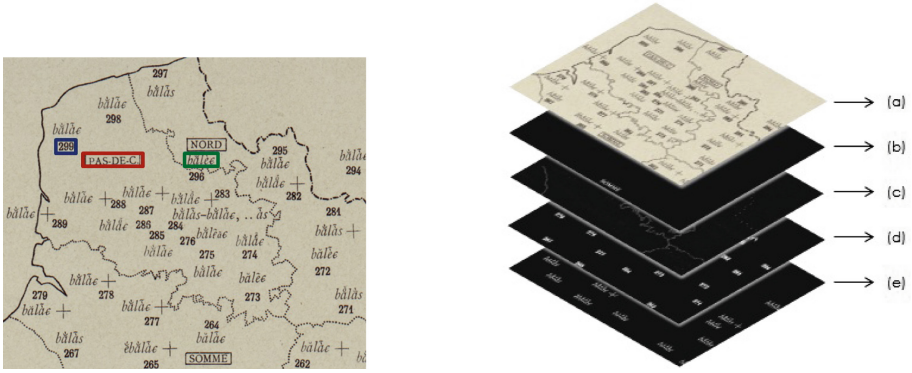


Fig. 1. Left: A French department name in red, a survey point number in blue, and a word in phonetics in green. Right: (a) Map; (b) French departments names; (c) Borders; (d) Survey point numbers; (e) Words in phonetics. (Color figure online)

The ALF maps are mainly composed of three kinds of elements: names of French departments (always surrounded by a rectangle), survey point numbers (identification of a city where a survey has been done), and words in phonetics (pronunciation of the word written in Rousselot-Gilliéron phonetic alphabet). Let us note that each map gathers the different pronunciations of a given word into a single map. An illustration of these components is given in Fig. 1.

This atlas is of prime interest for the researchers in dialectology as it allows to understand how the French language has evolved over the last century. This work takes place in the context of the ECLATS project, a French national research project[2] which aims at automatically extracting this information and generating maps with selected elements (currently, this process is done manually and it takes weeks to build a single map). More specifically, the aim of this paper is to separate each kind of information into layers in order to prepare data for subsequent analysis. The different layers of information are shown in Fig. 1 (right).

2 Related Work

Maps are composed of different layers of informations. Decomposing an image into meaningful components appears as one of major aims in recent development in image processing. The first goal was image restoration and denoising; but following the ideas of Meyer [17], in total variation minimization framework of Rudin, Osher and Fatemi [18], image decomposition into geometrical and oscillatory (i.e., texture) components appears an useful and very interesting way in computer vision and image analysis. There is a very large literature and also

[2] This work is supported by the French National Research Agency under the grant number ANR-15-CE38-0002.

recent advances on image decomposition models, image regularization, texture extraction and modeling or text-graphic separation. Among all the methods that have been proposed in the litterature, we can easily identify three main categories.

The first category of layer decomposition was based on color information. Color-based approaches have been used for separating an image into many layers [2,8,10] by clustering the color present in the document/map. However, the maps from our project can be seen as black and white images (black and white edited documents, worn out by time, then scanned, which makes them grayer and yellower) and layers of informations can not be distinguished using colors.

Looking for a fully generic approach, the second category of approach tried to decompose an image into layers of homogeneous information. The most recent and advanced work used Mathematical Decomposition or Morphological Component Analysis (MCA) [6,9]. MCA allows to separate features in an image which present different morphological aspects based on fast transform/reconstruction operators. Here again, our maps are mainly composed of black connected components on a noisy background, and a lot of overlapping text and graphics exits. Modelling each component in a generic way will impose to model all the different kind of details in all maps to finally obtained an overfitted model, or to manually post-process images like in [3].

Finally, the last category relies on the use of connected components. A lot of work have been done using the connexity of pixels in the litterature and seems to better fit to the features of our maps. Techniques mainly use the properties of the connected components, like [1] which use the generation of connected components and the application of the Hough transform in order to group together components into logical character strings which may then be separated from the graphics. Some bounding boxes (BB) of the components can also be used, like in [23], and used to compute some statistics (size of BB) to separate them. Using some automatic classification process, computed dynamically from the histogram for instance, the large graphical components can be discarded and the smaller graphics and text components kept.

Another work, based on such statistics, proposed to filter the components by their density [12]. Using this information, components were filtered to remove dashed lines. However, to properly filter out connceted components is not an easy task.

As presented before, the maps are composed of dark connected components on a light background (initially white but degraded by time and manipulations). Using connected components then appear as a natural choice where filtering the connected components is a difficult process. A recent subfield of mathematical morphology based on trees of connected components offers some strategies to decompose an image in layers of information [14,26]. This paper will propose to study this last solution.

The using of trees of connected components to separate in uniform layers of information, ancient documents which the layers have been “flattened” at printing, is an original approach. This has never been used for this type of

application. Moreover, even if this method use some filters and the thresholds inside the filters are (for the moment) set up manually, they are the only parameters. So, this method is generic and allows to extract components with an intelligent binarization and especially not a global one.

3 Tree of Connected Components

Mathematical morphology based on trees of connected components offers some strategies for obtaining meaningful hierarchical partitions from any hierarchical representation of an image. Classical connected components filtering techniques can be seen as shape-space filtering. Here, our idea is to apply some morphological operators to the shape graph-space of connected components extracted from the image. Then working on a tree rather than directly on the image will be much more efficient as maps are quite large (resolution of 9808 × 11824 pixels). The proposed method is based on the construction of a tree of all connected components of the input image. Then, on this tree, the components that do not correspond to the expected layers will be filtered out using their intrinsic properties.

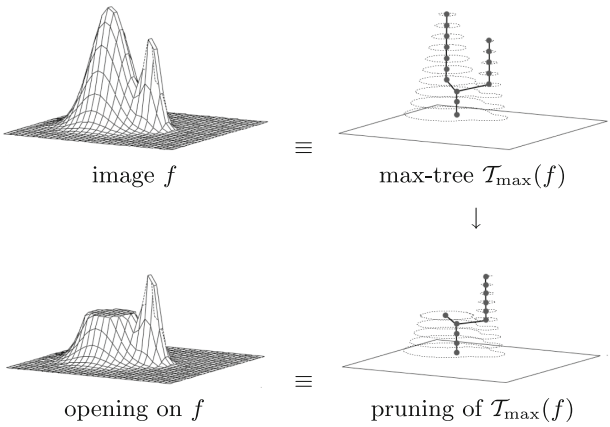


Fig. 2. A morphological connected operator (here an opening) based on a tree-based representation.

3.1 Definition

Whereas the most popular operators of mathematical morphology (MM) relies on structuring elements, the class of "connected operators" does not [14,21]. This class is very interesting because it satisfies the same numerous properties (and invariances) of MM operators, but with an additional property: connected operators do not shift object contours (they cannot create some new contours,

they just suppress some existing ones). Formally, φ is a *connected operator* if, applied on any image f, we have:

$$\forall\, x \mathcal{N} x',\ \ \varphi(f)(x) \neq \varphi(f)(x') \ \Rightarrow\ f(x) \neq f(x'),$$

where $\mathcal{N}$ is a neighborhood relationship. Some connected operators can be easily defined from some tree-based representations of a grey-level image [13,19,20]; such image representations express the inclusion of the connected components obtained by thresholding the image. Note that computing, storing, and processing such a component tree is very efficient [4,16]. In the following, we focus on a particular tree, namely the *max-tree*, that leads to morphological *algebraic openings*, that are, operators γ which are: increasing $(f_1 \leq f_2 \ \Rightarrow\ \gamma(f_1) \leq \gamma(f_2))$, idempotent $(\gamma \circ \gamma = \gamma)$, and anti-extensive $(\gamma \leq \mathrm{id})$. Replacing the *max-tree* by the *min-tree* leads to morphological *algebraic closings*, ϕ, which are increasing, idempotent, and extensive $(\phi \geq \mathrm{id})$. In addition, openings and closings have a strong property, shared by many morphological operators; they are invariant by contrast changes ($\forall g$ non-decreasing, $\gamma \circ g = g \circ \gamma$; the same goes for ϕ). This particular property is of prime importance because it implies that such operators have the ability to filter low-contrasted objects in the same way as they do with high-contrasted ones.

The *upper threshold set* (also called *upper level set*) at a given grey-level λ of a grey-level image f defined on a domain Ω is the set:

$$[\, f \geq \lambda \,] \ =\ \{\, x \in \Omega;\ f(x) \geq \lambda \,\} \ \in\ \mathcal{P}(\Omega),$$

and, from the family of sets $\{\, [\, f \geq \lambda \,] \,\}_\lambda$ we can easily reconstruct f, using: $\forall\, x,\ f(x) \ =\ \arg\max_\lambda \{\, \lambda;\ x \in [\, f \geq \lambda \,] \,\}$. When we consider the inclusion relationship, the set of connected components (obtained with the operator $\mathcal{CC}$) of all the threshold sets of f can be arranged into a tree, called max-tree of f:

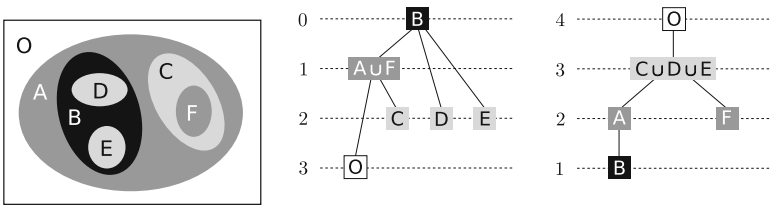
$$\mathcal{T}_{\max}(f) \ =\ \{\, \Gamma \in \mathcal{CC}(\, [f \geq \lambda] \,) \,\}_\lambda.$$

Such a tree is displayed in Fig. 2 (top right). If we prune this tree, such as in Fig. 2 (bottom right), we can reconstruct the function depicted in Fig. 2 (bottom left). Doing so, we have a way to construct an algebraic opening γ_α. This process can be defined thanks to a selector operator:

$$\mathrm{sel}_\alpha(\Gamma) \ =\ \begin{cases} \Gamma & \text{if } \alpha(\Gamma) \text{ is true,} \\ \emptyset & \text{otherwise,} \end{cases}$$

with the following constraint on α to ensure that it is a pruning: $\Gamma_1 \subset \Gamma_2 \ \Rightarrow\ \mathrm{sel}_\alpha(\Gamma_1) \subset \mathrm{sel}_\alpha(\Gamma_2)$. It is easy to see that we can use for α the comparison between an increasing attribute computed on a component and a threshold. For instance, with: $\alpha(\Gamma) \ =\ (\,\mathrm{card}(\Gamma) \leq N\,)$ we filter out any component of the max-tree which size (area, i.e., number of pixels) is below the threshold N, which leads to an *area opening* [24]. Pruning the same way the min-tree leads to an *area closing*.

Last, note that a larger class of filtering operators on trees have been defined in [27], and that there is a third morphological tree defined on threshold sets, called the *tree of shapes* [5,7,11].

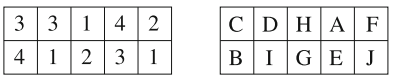


(a) image (b) max-tree (c) min-tree

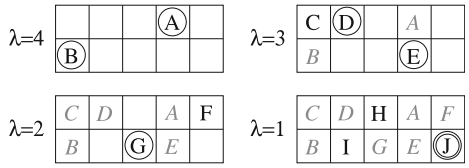
Fig. 3. The dual morphological trees of the same image; light (resp. dark) grey values represent high (resp. low) integer values.

3.2 Building a Tree

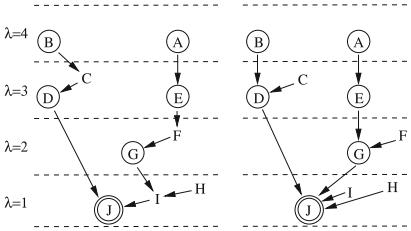
The connected component trees are used to select or prune parts of the images in an efficient manner. The max-tree (Fig. 3b) is a tree where grey values are ranked from the darkest to the lightest, and the min-tree (Fig. 3c) is the dual of the max-tree, as it ranks the grey values from the lightest to the darkest. A component tree can be computed directly on the original grey-scale image or, to be more robust to defects, to the result of some filtering process (for instance, some thin objects can be re-connected beforehand, so that the components of threshold sets are better formed).



(a) Grey-level image (left), and pixels annotated with letters (right).



(b) Annotated pixels with the different thresholds λ.



(c) Max-tree of the image of Figure 4(a): after computation (left) and after canonicalization (right).

Fig. 4. Illustration of a max-tree computation.

To better understand how a component tree is built, we use a simple example to illustrate the process. Let us consider an image of 5×2 pixels having grey levels in the range $[1, 4]$ and, to make the explanations easier, let us name each pixel of the image by a letter going from A to J (see Fig. 4(a)).

For each pixel having a grey level $\lambda \geq 4$, two distinct connected components are obtained, which are {A} and {B}. Note that their surrounding pixels belong

```
FIND-ROOT(x)
 1  if zpar(x) = x  then return  x
 2     else { zpar(x) ← FIND-ROOT(zpar(x)) ; return zpar(x) }

COMPUTE-TREE(f)
 1  for each p, zpar(p) ← undef
 2  R ← REVERSE-SORT(f)    // maps R into an array
 3  for  each p ∈ R in direct order
 4     parent(p) ← p ; zpar(p) ← p
 5     for  each n ∈ N(p) such as zpar(n) ≠ undef
 6        r ← FIND-ROOT(n)
 7        if  r ≠ p  then { parent(r) ← p ; zpar(r) ← p }
 8  DEALLOCATE(zpar)
 9  return pair(R, parent)    // a ``parent'' function

CANONICALIZE-TREE(parent, f)
 1  for  each p ∈ R in reverse order
 2     q ← parent(p)
 3     if  f(parent(q)) = f(q)  then  parent(p) ← parent(q)
 4  return parent    // a ``canonical'' parent function
```

Fig. 5. Code of the algorithm for creating a component tree.

to some other connected components. We then obtain two connected components, which are $\{\mathrm{A}\}$ and $\{\mathrm{B}\}$. In the next step, we move to the lower grey level value, here 3. We keep all pixels with a grey level value greater than or equal to 3. For each pixel having a grey level greater than or equal to 3, there are obtained two distinct connected components which are $\{\mathrm{A}, \mathrm{E}\}$ and $\{\mathrm{B}, \mathrm{C}, \mathrm{D}\}$. Now we need to choose, for each connected component, a new pixel that is not part of the former connected components, and preferably the last pixel is taken in the reading direction of the image (Z-reading). In other words, the pixel D now represents the component $\{\mathrm{B}, \mathrm{C}, \mathrm{D}\}$, obtained at threshold $\lambda = 3$, same thing with E for the component $\{\mathrm{A}, \mathrm{E}\}$. If we continue to apply the same approach for the rest of the image we obtain the results shown in Fig. 4(b). Finally the pixels of the image can be arranged into a rooted tree, shown in Fig. 4(c), where the arrows map a *parenthood* relationship.

In an equivalent manner, the min-tree of an image can be obtained going from the lowest to the highest grey level values.

The full algorithm, depicted in Fig. 5, takes only a few lines of code. There is nothing missing to be able to generate a tree of related components and is very easy to implement.

4 Extracting Map Components

4.1 Isolate Components

Based on the trees of connected components we have extracted, some components can be isolated by identifying their features. Note that the aim is to extract the content of the maps into several information layers, as shown in Fig. 1. So, we browse the created trees by filtering out the connected components that do not correspond to the required profile. Let us mention that computing some attributes related to connected components, and processing such trees to filter

out or identify some particular connected components are very easy [25]. More details about these tree structures and their implementation are given in [4].

4.2 Strategy to Manage the Different Layers

As shown in Fig. 1 (right), fours layers have been extracted: French department names, borders, survey point numbers, and words in phonetics. To extract the different layers and to make the algorithm more robust, the following strategy has been adopted. When a layer has been extracted, the corresponding connected components are removed from the initial image in order to process the next layer. Indeed, ignoring already identified components helps to reduce errors while extracting new components. The French department names and survey point numbers have been extracted using this strategy. However, if something is misidentified in a given layer, it will be deleted for the next filtering. So the strategy consists in processing the easiest layers first. In this work, the choice has been to process the names of French departments (which are always surrounded by a rectangle), then the numbers of survey points (which are of fixed size), then the borders, and finally the words in phonetics (residue of the input image with the previous layers). The Fig. 6 shows the filtering system of the proposed method.

4.3 Extraction of French Department Names

Let us have a look at a concrete example on a map for filtering French department names. For this example, a whole scanned map of France has been chosen. The Fig. 7(a) shows a zoom of this map which is given in its entirety in input. Dark connected components on a light background need to be extracted.

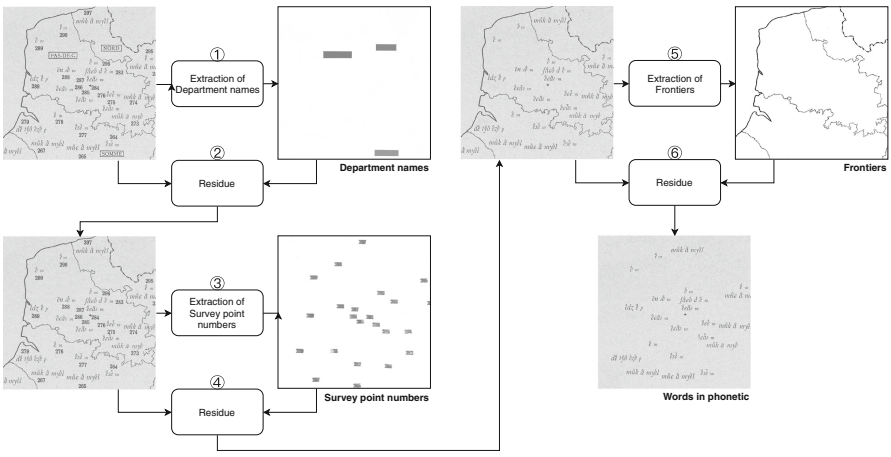


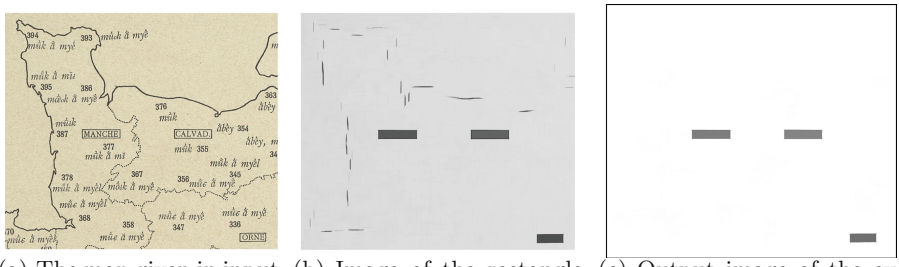
Fig. 6. The complete process of the filtering system for a map. Only a part of the map is shown, but the whole map was given for input and output.

For the French department names' extraction, the given input is the image of the map and there apply many filters to isolate French department names. A sampling of data allowed us to determine that these target objects were always surrounded by a rectangle, which vary in length but not in height. Rectangles are height invariant, but the length varies according to the name it contains. The minimum rectangle length that can be found on our example image is 130 pixels and the maximum is 433 pixels. The minimum rectangle height is 58 pixels and the maximum is 64 pixels, this is why we can consider that the height of the rectangles is invariant with regard to the height of the image (11824 pixels). With this informations, we considered that three types of filters should be implemented: a filter that recognize the vertical and horizontal lines of the picture to detect rectangles, a filter based on the dimensions of the components because rectangles are height invariant, and a filter based on the area of the components to filter the noise.

So the first step consists in isolating the vertical and horizontal lines from the rest. This strategy brings out all the rectangles of the map. Then a max-tree of new related components so formed will be created to try to fill in the rectangles that were highlighted in the previous step. On the min-tree created after the filling of the rectangles, a simple filter will be applied based on the properties of a connected component: if the white component at a height lower than the height of the rectangles we are trying to highlight, it is filled with the color of the parent component. So, all the small white components (the smallest components of the image being the inside of the rectangles) will be filled by a solid color (color of the outline of the rectangles), as shown in Fig. 7(b).

Since the expected properties of the components are known (height, width, area, etc.), another area filtering is applied to remove from the tree anything that does not correspond to what we expect. If the area of the component is smaller than the smallest rectangle, this component is deleted from the tree. This remove the noise that surrounds them (residues that match with very small rectangles).

Finally, we do the opposite filtering that filled the rectangles, using a min-tree instead of a max-tree, to remove the rectangles from the image and leave



(a) The map given in input. (b) Image of the rectangle components of the map. (c) Output image of the extraction of French department names.

Fig. 7. Different steps of French department names extraction (details).

only the outline of the map and the borders. The results, the isolation and extraction of French department names, lies in the residue of the two previous steps. We make the absolute difference of the image after removing the noise surrounding the rectangles with the image leaving only the outline of the map and the borders. The result of this filtering is shown in the Fig. 7(c).

4.4 Extraction of Survey Point Numbers

The filtering principle that was adopted for the French department names is adaptable to other information layers from the moment we know the properties of the connected components that we seek to extract. First of all, our goal is to bring out the desired connected components to the rest, and then we try to remove the noise that surrounds these connected components.

For the extraction of survey point numbers, the given input is the residue of the image of the map with the layer of French department names and there apply some area and dimension filtering to isolate survey point numbers. A sampling of data allowed us to determine that the survey point numbers objects were numbers ranging from 1 to 991 but discontinuously, and they are written in bold with always the same font. There is a maximum of 638 survey points on one map.

Survey point numbers are height invariant (font size), but the length varies according to the number it represents. The minimum survey point number length that can be found on our example image is 20 pixels and the maximum is 78 pixels. The minimum survey point number height is 36 pixels and the maximum is 45 pixels, this is why we can consider that the height of the survey point numbers is invariant with regard to the height of the image (11824 pixels). With this informations, we considered that two types of filters should be implemented: a filter based on the dimensions of the components because survey point numbers are height invariant, and a filter based on the area of the components to filter the noise. More, the mathematical morphology will be useful to group the numbers between them.

To extract this kind of components, the filtering consist to delete all the large components of the image. To isolate components, an area filter and a filter based on the size of the bounding boxes of the component are set up. If the property analyzed is above the defined thresholds (the largest area or dimension of a survey point number), the component take its parent's color in the min-tree (what will remove it from the image). The next step is to group numbers together (like chars to string), using mathematical morphology. This leads the last step of filtering, which is to remove the very small components (diacritics, frontiers made by dots, etc.) that remains on the image to get only the survey point numbers. To do this, as in the first filtering, an area filter and a filter based on the size of the bounding boxes of the component are set up. However this time, if the property analyzed is below the defined thresholds (the smallest area or dimension of a merged survey point number), the component take its parent's color in the min-tree (what will remove it from the image). The result of this

step is an image containing only the survey point numbers, as shown in the right side of the Fig. 6.

4.5 Extraction of Frontiers and Phonetic Words

Three kind of frontiers can be find in the maps: solid lines (frontiers with the seas and oceans), dot-line-dot lines (border with bordering countries), and dot lines (border inside France). Two methods are used to extract this three kind of frontiers. The first one is really simple and based on the area of the connected components. In the atlas, the solid lines which correspond to the frontiers with the seas and oceans are always touching the outline of the map. It means that the area of this component is the bigger area that we can find in the image. So, you just have to look for the widest dark component of our tree, and you can extract this type of frontier pretty easily. The second method consists in extracting the dot lines and the dot-line-dot lines by the nearest neighbor search. This approach allows to draw a line between the dots to regroup them in only one set. This also has the advantage of eliminating all surrounding noise such as diacritics that could be assimilated to border points. To summarize, these two methods will make it possible to extract all the borders of the map in an automatic way. Once the borders have been identified and extracted from the map, it will remain, on the image given in input, only the words in phonetics (residue of the previous filtering steps).

5 Evaluation

In this section, we report performance indicators for the proposed approach. In this work, an open source image processing library was used to build the trees of connected components [15].

5.1 Protocol and Metric

We evaluated the task of detecting individual objects of the following types: names of French departments, and survey point numbers. The method under evaluation is presented with the original and complete image of a map, and produces a set of areas of interest, each of them being annotated with a type. Areas of interest are implemented as series of point coordinates forming polygons.

The evaluation is based on the metrics proposed in [22]: for each content type present in the ground truth and in the results for the method under test, we compute the following indicators:

"correct" (COR): the number of objects which were correctly detected, with the appropriate type (otherwise they are counted as noise for other content types);

"missed" (MIS): the number of objects which were expected in the ground truth for a particular content type, but were not detected by the method under test;

"noise" (NOI): the number of objects which were detected by the method under test but which do not correspond to any expected element in the ground truth.

A given ground truth element $g \in G$ is considered as correctly detected by a resulting element $d \in D$ if g and d verify the following relations, where T_a is an absolute threshold set to 0.5 and T_r is a relative threshold set to 0.2:

$$\frac{area(g \cap d)}{area(g)} > T_a \qquad \frac{area(g \cap d)}{\sum_{g' \in G, g' \neq g} area(g' \cap d)} > T_r$$

$$\frac{area(g \cap d)}{area(d)} > T_a \qquad \frac{area(g \cap d)}{\sum_{d' \in D, d' \neq d} area(g \cap d')} > T_r$$

For completeness, we also report in the results the **total number of expected objects** (NGT) of each type in the ground truth and the **total number of detected objects** (NDE), as well as the **precision** (PRE) and the **recall** (REC) for each content type. Those indicators have the following definitions:

$$NGT = COR + MIS \qquad NDE = COR + NOI$$

$$PRE = \frac{COR}{NDE} \qquad REC = \frac{COR}{NGT}$$

5.2 Data and Ground Truth

Annotated evaluation data was created from the original ALF map dataset. The dataset regroups 1950 maps. For each map, 3 types of information must be annotated: 84 names of departments, 638 survey point numbers, 638 words in phonetics. If all this had to be done by hand, it would take a long time. That is why we have decided to impose a few constraints on ourselves concerning the creation of the ground-truth. In the dataset, there are 7 types of maps (showing the different parts of France), that is why we decided to construct a ground-truth for each type of map to represent the atlas as much as possible while not spending too much time to do it. To save as much time as possible, we also decided to use the results of the current segmentation to avoid redoing everything from scratch and placing the points one by one, but to just move the points as precisely as possible if the segmentation is bad or missing.

One map showing France entirely was manually annotated to produce the ground truth for the task we previously introduced. This evaluation map (named "ALF0101" and visible in Fig. 7(a)) contains a total of 84 names of French departments, and 638 survey point numbers[3]. Each annotation is composed of a region described by a polygon and a content type described by a string: "French department", "survey", etc. Due to resource constraints, we could not annotate more maps. Our work is currently focused on the building of the ground-truth

[3] The ground truth of this evaluation map is available at http://l3i-share.univ-lr.fr/datasets/CarteALF0101.lif.

for the phonetics words on this map. After that, we will extend our ground-truth to other types of maps.

The dataset is composed by one atlas (ALF), which regroup 1950 maps. For each map, 3 types of information must be annotated: 84 names of departments, 638 survey point numbers, 638 words in phonetics. If all this had to be done by hand, it would take a long time. That is why we have decided to impose a few constraints on ourselves concerning the building of the ground-truth. In the dataset, there are 7 types of maps (showing the different parts of France), that is why we decided to construct a ground-truth for each type of map to represent the atlas as much as possible while not spending too much time doing it. To save as much time as possible, the results of the current segmentation will be used to avoid redoing everything from scratch and placing the points one by one, but to just move the points as precisely as possible if the segmentation is bad or missing.

5.3 Results

The results are presented in Table 1. Thanks to the trees of connected components and the filtering of the elements that compose them, the layer of information corresponding to the French department names can be successfully extracted. The position of French department names on the map can be easily determined. Concerning survey point numbers, the method was able to detect 86.36% for the target elements (551 items were well detected) while introducing 128 extra elements (noise).

Table 1. Results obtained on the map ALF0101.

	COR	*MIS*	*NOI*	*NGT*	*NDE*	*PRE*	*REC*
Names of French departments	84	0	0	84	84	100.0 %	100.0 %
Survey point numbers	551	87	128	638	679	81.2 %	86.4 %

The analysis of theses results show that some survey point numbers are missing ("MIS" column) because, in the original map, these components are directly connected to another bigger component, like frontiers. Survey point numbers are expected to be small components of the image, so large components (such as frontiers) are filtered at the beginning of the extraction step for this information layer, which also removes survey point numbers that touch those frontiers. The components wrongly detected as survey point numbers ("NOI" column) are all phonetic word letters that have not been well filtered during the extraction process because of their size similar to numbers.

6 Conclusion

In this paper, an extraction system for content of ancient maps using trees of connected components has been presented. The system take as input an image (scan of the map) and delivers different layers of information. Each layer correspond to a specific kind of information and there positions. The proposed approach uses a tree of connected components based on the grey level of the input image. Working on a tree rather than directly on the image will be much more efficient as maps are quite large. An adapted filtering of this tree allows to extract expected components by using their intrinsic properties. Thus, the method allows to localize the position of the extracted components.

The evaluation have given the number and the actual position of the components that are not correctly detected, the future works will consist on refining our approach in order to detect them more appropriately. Following the results, our approach needs to be modified to filter the numbers of survey points with a better accuracy. If this detection of survey point numbers is improved, this will allow us to perfectly detect phonetic words (residue of the basic image with all filtered layers) without missing or false-alarm components.

From the moment the layers of information are well identified, a system of text recognition in phonetics will be made. It should be based on an optical character recognition (OCR) method, and a dedicated recognition system that could be able to differentiate the many diacritics in phonetic words.

References

1. Fletcher, L.A., Kasturi, R.: A robust algorithm for text string separation from mixed text/graphics images. IEEE Trans. Pattern Anal. Mach. Intell. **10**(6), 910–918 (1988)
2. Bres, S., Eglin, V., Poulain, V.: Semi automatic color segmentation of document pages. CoRR, abs/1609.08393 (2016)
3. Cao, R., Tan, C.L.: Text/Graphics separation in maps. In: Blostein, D., Kwon, Y.-B. (eds.) GREC 2001. LNCS, vol. 2390, pp. 167–177. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45868-9_14
4. Carlinet, E., Géraud, T.: A comparative review of component tree computation algorithms. IEEE Trans. Image Process. **23**(9), 3885–3895 (2014)
5. Carlinet, E., Géraud, T.: MToS: a tree of shapes for multivariate images. IEEE Trans. Image Process. **24**(12), 5330–5342 (2015)
6. Coustaty, M., Dubois, S., Ogier, J.-M., Menard, M.: Segmenting and indexing old documents using a letter extraction. In: Ogier, J.-M., Liu, W., Lladós, J. (eds.) GREC 2009. LNCS, vol. 6020, pp. 142–149. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13728-0_13
7. Crozet, S., Géraud, T.: A first parallel algorithm to compute the morphological tree of shapes of nD images. In: Proceedings of the 21st IEEE International Conference on Image Processing (ICIP), Paris, France, pp. 2933–2937 (2014)
8. Dhar, D.B., Chanda, B.: Extraction and recognition of geographical features from paper maps. Int. J. Doc. Anal. **8**(4), 232–245 (2006)

9. Dubois, S., Péteri, R., Ménard, M.: Decomposition of dynamic textures using morphological component analysis. IEEE Trans. Circuits Syst. Video Techn. **22**(2), 188–201 (2012)
10. Ebi, N., Lauterbach, B., Anheier, W.: An image analysis system for automatic data acquisition from colored scanned maps. Mach. Vis. Appl. **7**, 148–164 (1994)
11. Géraud, T., Carlinet, E., Crozet, S., Najman, L.: A quasi-linear algorithm to compute the tree of shapes of nD images. In: Hendriks, C.L.L., Borgefors, G., Strand, R. (eds.) ISMM 2013. LNCS, vol. 7883, pp. 98–110. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38294-9_9
12. Höhn, W.: Detecting arbitrarily oriented text labels in early maps. In: Sanches, J.M., Micó, L., Cardoso, J.S. (eds.) IbPRIA 2013. LNCS, vol. 7887, pp. 424–432. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38628-2_50
13. Jones, R.: Component trees for image filtering and segmentation. In: Coyle, E. (ed.) Proceedings of the IEEE Workshop on Nonlinear Signal and Image Processing, Mackinac Island (1997)
14. Lazzara, G., Géraud, T., Levillain, R.: Planting, growing and pruning trees: Connected filters applied to document image analysis. In: Proceedings of the 11th IAPR International Workshop on Document Analysis Systems (DAS), IAPR, Tours, France, April 2014, pp. 36–40 (2014)
15. Lazzara, G., Levillain, R., Géraud, T., Jacquelet, Y., Marquegnies, J., Crépin-Leblond, A.: The SCRIBO module of the Olena platform: a free software framework for document image analysis. In: Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR), IAPR, Beijing, China, September 2011, pp. 252–258 (2011)
16. Meijster, A., Wilkinson, M.H.F.: A comparison of algorithms for connected set openings and closings. IEEE Trans. Pattern Anal. Mach. Intell. **24**(4), 484–494 (2002)
17. Meyer, Y.: Oscillating Patterns in Image Processing and Nonlinear Evolution Equations. The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures. American Mathematical Society, Boston (2001)
18. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Phys. D Nonlinear Phenom. **60**(1–4), 259–268 (1992)
19. Salembier, P., Oliveras, A., Garrido, L.: Antiextensive connected operators for image and sequence processing. IEEE Trans. Image Process. **7**(4), 555–570 (1998)
20. Salembier, P., Serra, J.: Flat zones filtering, connected operators and filters by reconstruction. IEEE Trans. Image Process. **3**(8), 1153–1160 (1995)
21. Salembier, P., Wilkinson, M.H.: Connected operators. IEEE Signal Process. Mag. **26**(6), 136–157 (2009)
22. Shafait, F., Keysers, D., Breuel, T.: Performance evaluation and benchmarking of six-page segmentation algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **30**(6), 941–954 (2008)
23. Tombre, K., Tabbone, S., Pélissier, L., Lamiroy, B., Dosch, P.: Text/Graphics separation revisited. In: Lopresti, D., Hu, J., Kashi, R. (eds.) DAS 2002. LNCS, vol. 2423, pp. 200–211. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45869-7_24
24. Vincent, L.: Grayscale area openings and closings, their efficient implementation and applications. In: Proceedings of the EURASIP 1st Workshop on Mathematical Morphology and its Applications to Signal Processing (ISMM), Barcelona, Spain, May 1993, pp. 22–27 (1993)

25. Xu, Y., Carlinet, E., Géraud, T., Najman, L.: Efficient computation of attributes and saliency maps on tree-based image representations. In: Benediktsson, J.A., Chanussot, J., Najman, L., Talbot, H. (eds.) ISMM 2015. LNCS, vol. 9082, pp. 693–704. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18720-4_58
26. Xu, Y., Carlinet, E., Géraud, T., Najman, L.: Hierarchical segmentation using tree-based shape spaces. IEEE Trans. Pattern Anal. Mach. Intell. **39**(3), 457–469 (2017)
27. Xu, Y., Géraud, T., Najman, L.: Connected filtering on tree-based shape-spaces. IEEE Trans. Pattern Anal. Mach. Intell. **38**(6), 1126–1140 (2016)