An incremental approach for the detection of legend text in digital maps

Arthur Marzinkowski, Salem Benferhat, Anastasia Paparrizou, Cédric Piette

An incremental approach for the detection of legend text in digital maps

Arthur Marzinkowski
CRIL - University of Artois
CNRS - UMR 8188
Lens, France
marzinkowski@cril.fr

Salem Benferhat
CRIL - University of Artois
CNRS - UMR 8188
Lens, France
benferhat@cril.fr

Anastasia Paparrizou
LIRMM
Montpellier, France
anastasia.paparrizou@lirmm.fr

Cedric Piette
CRIL - University of Artois
CNRS - UMR 8188
Lens, France
piette@cril.fr

***Abstract*—The presented work concerns the automatic detection of legend texts inside maps. After extracting the texts from the images using OCR tools, we use an iterative clustering process on the extracted texts. We consider five main criteria, with different levels of importance: text alignment, distance between text boxes, background color of the text, font color, and font size. For each criterion, we define appropriate similarity measures. We propose a method that combines, incrementally, the partitions obtained by each criterion. The experimental study reveals two important results. First, combining several criteria gives better results than considering a single distance metric (e.g., Euclidean distance) between text boxes. Secondly, the overall effectiveness of the priority relation, which we intuitively defined among the criteria for detecting caption texts, is confirmed.**

***Index Terms*—Digital maps, OCR, clustering**

I. Introduction

The automatic detection of texts inside images has been studied by the computer vision community for several decades. The text detection and text grouping is mainly based on visual features that are extracted from the image (i.e., color, shape etc.). Text recognition inside images and videos with arbitrary and cluttered background has attracted a lot of interest, especially for helping visually impaired people. This work deals with the text detection inside images that belongs to the legend, a specific area of information that interests the reader. The novelty of this work is that not only visual characteristics are used, but also semantic notions that accompany and aid the legend area detection. For example, semantic information might be: the possible position of the object in the map, the relation or distance among neighboring objects, etc.

The images that we deal with are images that represent digital maps. This work is a primary step in the detection of objects of interest inside maps, that are usually defined and displayed in the legend. For this purpose, we start by calling an optical character recognition algorithm (OCR) in order to obtain the text boxes (i.e., streams of characters included in a rectangle) that appear inside a given map. We use the text regions provided by the ORC algorithm as an input to a clustering algorithm we propose, based on the well-known k-means algorithm. We also introduce criteria that characterise either in a symbolic or visual way the notion of legend. For instance, text phrases that are aligned vertically may formulate a legend (symbolic/qualitative criterion), the background color of text boxes is a solid color (visual criterion). We define five such criteria as well as distance measures associated to each criterion for the k-means algorithm. We deploy k-means in an incremental way, since clusters are being built according to a single criterion that is considered at each iteration. Therefore, the order in which the criteria are considered affects the efficiency of our algorithm. The decision of merging or splitting inside a cluster is based on the entropy we compute to reflect the inner homogeneity.

Among the several text regions provided, we construct and deconstruct clusters of them iteratively until we converge to a set that is most probably the legend region we seek for. The originality of this work is the incremental way we build clusters, in order to consider the various criteria we impose. Additionally, we can give a different importance to each criterion by prioritizing it, resulting in a hierarchical management of the clusters.

We evaluate the results of the legend detection on several maps of different dimensions, analyses, and styles using two evaluation metrics. The results show that our algorithm detects quite well the legend region on several maps, having the tendency to slightly ”over-approximate” the target region. Fruitful observations are derived by the exhaustive study we conducted on the criteria. Namely, when more than one criterion are considered, we obtain a better approximation of the legend region. When changing the priority order on the criteria, the detection deteriorates.

The rest of the paper is organized as follows. The next section briefly presents some related work. Section III describes the problem considered in this paper, defines the five criteria used for clustering and presents our legend text detection algorithm. Section V contains the experimental results, and finally, Section VI concludes the paper.

II. Related work

Text detection or recognition usually concerns text documents, and many OCR algorithms have been proposed for this purpose with excellent accuracy [1]. Last years, text detection in natural scene images has attracted a lot of interest due to a variety of applications: scene understanding, content-based image retrieval, assistive navigation (for autonomous cars or visually impaired people) [2]. In [3], the authors proposed a

framework of text string detection, where the character candidate grouping is based on structural features, such as character size differences, distances between neighboring characters, and character alignment. Their text line grouping method performs Hough transform to fit text line among the centroids of text candidates instead of clustering. This work is similar to ours in the sense that semantic notions are considered but the purpose and granularity in different (character and line grouping vs text boxes grouping). Similarly, in [4], the proposed assistance system recognizes text in an image based on structural features like size, orientation, and distance between the successive regions of interest.

In the information retrieval community, there is a considerable effort for text clustering with the well-known k-means algorithm, but for different purposes than ours (i.e., for scoring and ranking a document's relevance given a user's query [5], grouping documents of similar content[6], text summarization [7]). There exist works on text detection in raster maps, but they are more related to ORC approaches, where the challenge originates from the varying text orientations and the overlap of text labels [8], [9]. To our knowledge, there are no works that deal with text detection that belongs to legends of maps.

III. Problem description and grouping criteria

A. Problem description

The problem we are dealing with is the identification of the region of a map where the legend text is located. The data input of our algorithm are images representing maps with legends. The maps are made up of figures, but also of text areas. In this paper, we propose an approach to discriminate text belonging to a potential legend from other text regions in the image.

Each image will be represented by a matrix of $n \times m$ elements, which will be denoted in the following by $\mathscr{I}$. Each element of the matrix represents a pixel color (represented here in RGB format, i.e. a triplet of integers between 0 and 255). We will use the index x_i to designate a given row number and y_j to designate a given column number of the matrix $\mathscr{I}$.

In this paper, we focus on images containing legends. In particular, we are interested in the area of the map containing the legend text, which will also be represented by a pixel color matrix denoted $\mathscr{L}$. The output of our algorithm is an area of the matrix $\mathscr{I}$, which is intended to represent the text area of the legend $\mathscr{L}$.

We now introduce two concepts that will be used by our algorithm in the following. The first notion, which we call splitting, consists in building a partition of a set from a larger partition. The second is a refinement on the notion of entropy, which will be used to measure the homogeneity of a partition.

Definition 1 (Splitting): Let A be a set of elements. Let B be a subset of A and $\mathscr{P}_A = \{C_i : i = 1, .., t\}$ be a partition of A (namely, $\forall C_i, C_i \neq \emptyset$, $\bigcap_{i=1,..t} C_i = A$ and $\forall j \in \{i = 1, ..t\}, k \in \{i = 1, ..t\}$ with $j \neq k$, we have $C_j \cap C_k = \emptyset$). We call the partition of B by $\mathscr{P}_A$, denoted $B \rhd \mathscr{P}_A$, the partition of B obtained by intersecting each element of $\mathscr{P}_A$ with B. More formally:

$$B \rhd \mathscr{P}_A = \{B \cap C_i : C_i \in \mathscr{P}_A\} \tag{1}$$

Definition 2 (Homogeneity measure): Let A be a set of elements and $\mathscr{P}_A$ be a partition of A. We define the homogeneity (or entropy) of $\mathscr{P}_A$, denoted $\mathscr{E}(\mathscr{P}_A)$, by:

$$\mathscr{E}(\mathscr{P}_A) = -\sum_{B_i \in \mathscr{P}_A} \frac{\|B_i\|}{\|A\|} * \log_2 \frac{\|B_i\|}{\|A\|}, \tag{2}$$

where $\|x\|$ represents the cardinality of x.

B. Extracting texts from maps

The first step of our algorithm is to extract text from the maps. In this step, we simply use existing OCR tools. In particular, we are using the DocTr OCR tool [10], which enables us to obtain a list of text zones with different levels of granularity (blocks of text, a line of text, a word, etc.) from an image.

Given a $\mathscr{I}$ map, the OCR tool returns a set of texts, denoted by $\mathscr{T}_{\mathscr{I}}$. The elements of $\mathscr{T}_{\mathscr{I}}$ are called text boxes and are denoted by the lower-case calligraphic letters $\mathscr{a}, \mathscr{b}, \mathscr{c}, etc.$.

Figure 1 gives an example of a text box area, which is represented by a rectangle on the map identified by the two end points of its secondary diagonal, denoted by $(x^{\mathscr{a}}, y^{\mathscr{a}})$ and $(x_{\mathscr{a}}, y_{\mathscr{a}})$.

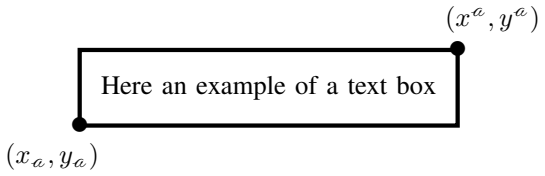

Fig. 1. An example of a text box with the representation of its coordinate notations

Remarks:

- In the rest of this paper, we assume that all rectangles associated with text boxes are disjoint.
- We also exclude rectangles that are not vertical or horizontal (in relation to the image axes), as the legends are not supposed to be inclined.

C. Definitions of clustering criteria

This subsection presents the five main criteria that will be used by our method for detecting legend text areas. For each of the five criteria, we define the associated similarity measures for comparing two text zones. Let $\mathscr{a}$ and $\mathscr{b}$ be two text boxes. For the sake of simplicity, we will use the term distance to express the similarity between two text boxes, without requiring any prior property on the distance measures used in clustering algorithms.

Criterion 1: text alignment : Legend text is often aligned. Text alignment can take various forms: left, center, right, etc. For the sake of simplicity, we limit ourselves to the situation of legends with text aligned vertically on the left. Extension to other forms of alignment can easily be done by symmetry or by a slight adaptation of the distance measure defined below. The distance associated with text left alignment, denoted d_{ag}, is defined by:

$$d_{ag}(a, b) = \mid y_a - y_b \mid . \tag{3}$$

Criterion 2: distances between text boxes : A second natural criterion is the similarity measure between text zones, which we will denote by d_e. This is a natural criterion, since legend texts are generally close to each other.

A primary idea for defining the distance between two text boxes is to consider the smallest distance between any two points on the perimeters of the rectangles associated with the two text boxes. This solution is not satisfactory in our context, as we are using particular rectangles (disjoint, horizontal or vertical, text aligned on the left, etc.).

The definition of the distance between two boxes clearly depends on the results of OCR, which may return whole sentences or simply isolated words. We propose to analyze each of these two situations.

Let us start with the case where the boxes contain whole sentences. Figure 2 shows the "ideal" situation in which the distance between two text boxes a and b is equal to 0. These are two boxes perfectly aligned on the left (recall that, only alignment on the left is considered) and whose distance is equal to the basic unit for distances (a single pixel in our case).

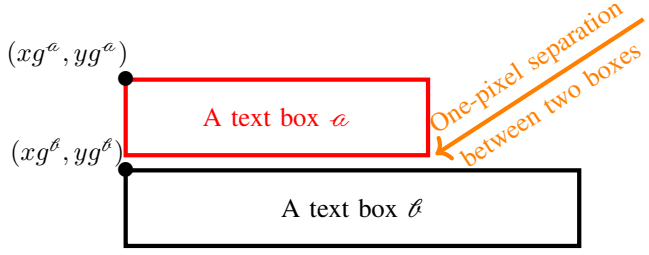

Fig. 2. Situations where two text boxes are considered as close as possible

More formally, let a and b be two disjoint text boxes. Then:

$$d_e(a, b) = 0$$
iff
$$(y_a = y_b)$$
and
$$[(xg^a = x_b + 1) \text{ or } (xg^b = x_a + 1)].$$

Note that the distance only applies to two disjoint text boxes (this is the context of our paper).

Based on this definition of situations where two text boxes are considered ideally close, it is then sufficient to define the distance as the number of basic movements required for one of the two boxes to move to reach the aforementioned ideal situation.

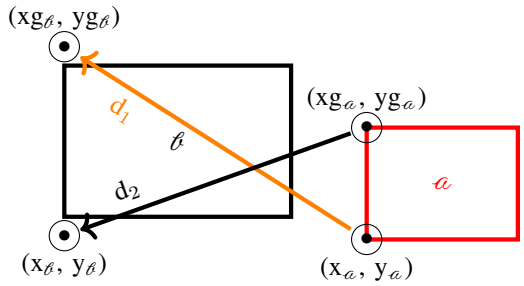

Fig. 3. Illustration of the computation of the distance between two text boxes

More formally, from the notations given in figure 3, we have:

$$\begin{aligned} d_e(a, b) &= min(d_1, d_2), \text{with} \\ d_1 &= \sqrt{(x_a - xg_b)^2 + (y_a - yg_b)^2}, and \\ d_2 &= \sqrt{(x_b - xg_a)^2 + (y_b - yg_a)^2}. \end{aligned} \tag{4}$$

Of course, other measures (such as the Manhattan distance) can be used alternatively to the Euclidean distance.

Now, assume that the text boxes contain only words. In this case, the (Euclidean) distance between the word centers is sufficient. Let us denote (xc_a, yc_a) and (xc_b, yc_b) the coordinates of the points that represent the centers of the a and b boxes respectively. The distance between a and b is simply defined by:

$$d_e(a, b) = \sqrt{(xc_a - xc_b)^2 + (yc_a - yc_b)^2} \tag{5}$$

Criterion 3: Background colors of text boxes : The third criterion is the background color of the text zone. The background color used in legends is often homogeneous (usually solid). The use of this criterion raises two questions, both of which are well addressed in the image processing literature (e.g., [11]).

The first question is how to determine the background of a text box a. In our context, we can simply use the pixel color frequency, as we can reasonably assume that the texts in the boxes to be compared are of homogeneous color and that the texts occupy the majority of the box. Another way of proceeding, which is the one followed in the paper, is to calculate the most connected (related) part of the box (two pixels are connected if they are of homogeneous color, i.e. both pixels are of sufficiently similar color) starting from one of the edges of the text box.

The second question is how to determine the color representative of the text background. Once more, we have several possibilities (e.g. [12]). One possibility, based on the RGB values of each pixel in the text area, is to i) first calculate the sum of the squares of the colors (value by value), ii)

then divide the result by the number of pixels and ii) finally apply the square root to the result (to stay within the set $\{0, ..., 255\}$). Another possibility is simply to take the average of the colors (value by value) of the different pixels of the image background. The later method, is the one used in this paper.

Let us now denote $(RF_{\mathscr{a}}, GF_{\mathscr{a}}, BF_{\mathscr{a}})$ and $(RF_{\mathscr{b}}, GF_{\mathscr{b}}, BF_{\mathscr{b}})$ the average background colors of the $\mathscr{a}$ and $\mathscr{b}$ text boxes respectively. Now, let us calculate the proximity between these two colors, where again several definitions are possible. In this paper, we use the Euclidean distance, denoted d_{bg}, and defined by:

$$d_{bg}(\mathscr{a}, \mathscr{b}) = \sqrt{(RF_{\mathscr{a}} - RF_{\mathscr{b}})^2 + (GF_{\mathscr{a}} - GF_{\mathscr{b}})^2 + (BF_{\mathscr{a}} - BF_{\mathscr{b}})^2}. \quad (6)$$

Criterion 4: Height of text boxes : The three criteria described above (alignment, distance and background color of the text area) are fundamental in determining the text area of a legend. Another auxiliary criterion, concerns the size of the font used, which here, is assumed to be equal to the height of the text box. The fourth distance measure, denoted d_t, represents the height of the text boxes and is simply defined by:

$$d_t(\mathscr{a}, \mathscr{b}) = \mid (\mid x_{\mathscr{a}} - x^{\mathscr{a}} \mid) - (\mid x_{\mathscr{b}} - x^{\mathscr{b}} \mid) \mid \quad (7)$$

Critère 5: Text color : The last criterion is the dual of criterion 3 (the background color of a text box), since we consider that a text box can be divided into two parts: the part that contains the text characters of the text box and the rest that represents the background of the text box. Let us denote $(RT_{\mathscr{a}}, GT_{\mathscr{a}}, BT_{\mathscr{a}})$ and $(RT_{\mathscr{b}}, GT_{\mathscr{b}}, BT_{\mathscr{b}}$) the (average) text colors of the $\mathscr{a}$ and $\mathscr{b}$ text boxes respectively. Then the distance with respect to the text color, denoted d_{ct}, is defined by:

$$d_{ct}(\mathscr{a}, \mathscr{b}) = \sqrt{(RF_{\mathscr{a}} - RT_{\mathscr{b}})^2 + (GT_{\mathscr{a}} - GT_{\mathscr{b}})^2 + (BT_{\mathscr{a}} - BT_{\mathscr{b}})^2}. \quad (8)$$

IV. A CLUSTERING ALGORITHM BASED ON SUCCESSIVE CLUSTER REFINEMENTS

In this section, we propose an approach based on *clustering* [13], [14] in order to group the text boxes that belong to the legend area. Since the text boxes extracted from images are not labeled, we opt for methods based on unsupervised learning. Precisely, in this paper we opted for the k-means algorithm (e.g., [15]) which is widely used in unsupervised classification problems.

We are interested in grouping regions of text in such a way that these groupings represent the text of a legend. The input data of our algorithm is a map which we assume that it contains a legend. This map will be denoted $\mathscr{I}$. To this input, we associate two parameters. The first is a vector, denoted $\vec{c}$, of criteria (of size 1 to 5) which indicates the order in which the criteria must be used. The second parameter is a threshold function, denoted $\sigma(\mathscr{A})$ which indicates whether a partition of a set $\mathscr{A}$ is sufficiently homogeneous or not. Similarly to any collection function, the difficult question is how to set this threshold. In our experimental study, it is set to 80 % of the maximum value.

In addition to the two aforementioned parameters, two other functions are considered in the input data. First, the $OCR(\mathscr{I})$, which returns the set of text boxes $\mathscr{T}_{\mathscr{I}}$ that are in the image $\mathscr{I}$. Second, the function for the text grouping. As we stated above, we simply use the k-means method.

Algorithm 1 Legend detection algorithm

Inputs:
- $\mathscr{I}$: a map containing a legend
- $\vec{c}$: a vector of $n \in \{1, ..., 5\}$ criteria
- σ: a function that takes a partition and returns a real number
- k-means: the clustering method used

Outputs:
- $\mathscr{P}$ A set of *clusters*.

1: *// First an OCR tool is applied to the image*
2: $\mathscr{T}_{\mathscr{I}} \leftarrow OCR(\mathscr{I})$
3: *// The k-means algorithm is applied to* $\mathscr{T}_{\mathscr{I}}$
4: *// with the first criterion* $\vec{c}[1]$
5: $\mathscr{P} \leftarrow k - means(\mathscr{T}_{\mathscr{I}}, \vec{c}[1])$
6: *// We refine the set* $\mathscr{P}$ *iteratively with the other criteria*
7: **for all** $i \in \{2, ..., n\}$ **do**
8: *// Apply the k-means algorithm to* $\mathscr{T}_{\mathscr{I}}$
9: *// with the i-th criteria* $\vec{c}[i]$
10: $\mathscr{C} \leftarrow k - means(\mathscr{T}_{\mathscr{I}}, \vec{c}[i])$
11: *// We refine each element of*
12: *// the current partition* $\mathscr{P}$
13: *// Results will be saved in* $\mathscr{X}$
14: $\mathscr{X} \leftarrow \emptyset$
15: **for all** $B \in \mathscr{P}$ **do**
16: *// We split B from* $\mathscr{C}$ *using Definition 1*
17: $\mathrm{R}_B = B \rhd \mathscr{C}$
18: *// We check the refinement of B, i.e. if* R_B *is homogeneous*
19: **if** $\mathscr{E}(R_B) <= \sigma(R_B)$ **then**
20: $\mathscr{X} \leftarrow \mathscr{X} \cup \{B\}$
21: **else**
22: $\mathscr{X} \leftarrow \mathscr{X} \cup R_B$
23: **end if**
24: **end for**
25: $\mathscr{P} \leftarrow \mathscr{X}$
26: **end forreturn** $\mathscr{P}$

The proposed algorithm (Algorithm 1) consists of three main steps.

- The first step (lines 1 and 2 of Algorithm 1) is simply to extract the text boxes from the image by utilizing an OCR tool. The result of this step is a set of text boxes.

- The second step (lines 3 and 4) consists in applying the k-means algorithm to the set of text boxes. The resulting partition is denoted by $\mathscr{P}$.
- The third step (lines 7-21) consists in progressively refining the clusters obtained with each of the remaining criteria. First, for each criterion, the k-means algorithm is applied to the set of text boxes given by an OCR algorithm. Then, each cluster element B is splitted (according to Definition 1) by applying the k-means algorithm on the result of clustering obtained in line 10. Whether the entropy associated with the splitting result is better or not than $\sigma(R_B)$, then element B is replaced by its splitting result or remains unchanged respectively (lines 18-21).

We therefore use this entropy measure as a criterion for evaluating the homogeneity of the different clusters obtained by applying one criterion compared to the clusters obtained by a different criterion.

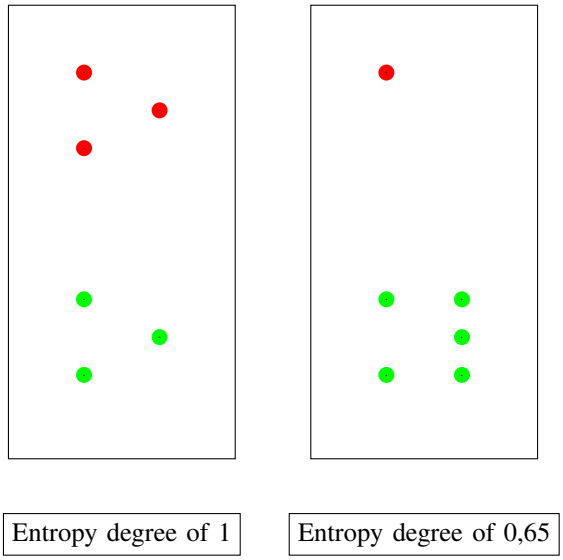

Fig. 4. Two examples of clustering results with their entropy values

Figure 4 depicts an example of entropy degree on two different clusters. Each element, displayed by a dot, belongs to the same *cluster* of criterion 1. The colors represent the different clusters calculated by criterion 2. For the left example, which represents a *clustering* given by criterion 1, three points belong to the "red" class and the other three belong to the "green" class. The entropy degree here is maximum (i.e., 1).

The example on the right has only one element belonging to the "red" class. Thus, the different classes of criterion 2 are much less homogeneous. As a result, the entropy value of this *clustering* is lower (with an entropy equal to 0.65).

In the first case, the *cluster* is split into two new clusters, while in the second case the *cluster* remains unchanged.

V. Experimental studies

The experiments carried out in this paper were performed on processors *Intel XEON E5-2637 v4* of 4 cores at 3.6 GHz with 128 GB RAM RDIMM, under *CentOS 7.3 Kernel 3.10*. In terms of software configurations, we used the OCR library *DocTr version 4.0*[10] and the implementation of *clustering scikit-learn version 1.3.2* [16].

We selected 29 images on which we conducted our legend detection procedure, using the various criteria studied in this paper. On each map, the legend was labeled by hand, in order to compare the result of our calculations with the "real" legend. We took care to select maps that were very different from one another in terms of content, resolution and legend representation. The maps used in our experiments are available at https://www.cril.univ-artois.fr/~marzinkowski/incremental-clustering-results/.

We specify here, that we are comparing areas (or zones) of the image, because what we are interested in, is whether a part of the map is a legend or not. One of the advantages of proceeding by this way is that, even if a word (within a word group) is not detected by the OCR tool, we can nevertheless recover a large part of the area (if not the entire area when the word is within a word group). Therefore, our detection becomes less sensitive to the OCR results.

We do not report in detail the running times, as they are quite similar in all our experiments. However, we provide a general idea of the computational cost of the various stages of our legend detection algorithm: (i) the running time of the `OCR` on our dataset varies between 10 and 25 seconds, depending on the amount of text appears on the map that is given as input (ii) the five sets of partitions obtained with k-means are computed in about one second and finally, (iii) the time required for refinement never exceeds 100ms. A more precise summary of the processing times for the various stages is given in Table I.

We can therefore see that, whatever the map used, our algorithm takes less than 20 seconds to complete all its steps for detecting the texts belonging a legend.

	Moyenne	Écart type
OCR	17,231	2,095
Clustering	1,036	0,790
Raffinement	0,005	0,002

TABLE I
Summary of running times (in seconds)

A. Evaluation method

We have considered two evaluation metrics:

- Intersection Over Union (IOU)
- Intersection Over Labels (IOE)

These metrics are used to compare two areas of the map. One is the area produced by our algorithm, and the other is the manually labeled area (considered as the actual area label).

Roughly speaking, *Intersection over Union* (IOU) consists in calculating the ratio between the surface of the intersection of two zones, and that of their union. For *Intersection over label* (IOE), it represents the ratio between the intersection of the two zones, and the labeled zone. These two metrics have

the advantage of being insensitive to map resolution; IOU has also been used in a number of works [17].

Let us illustrate these metrics in Figure 5. It contains three color zones: red, blue and yellow; whose areas are denoted by A_R, A_B and A_J respectively.

We consider the labeled rectangle (true answer) to be the rectangle including both blue and yellow surfaces, while the area calculated by the algorithm is represented by the yellow and red rectangles. Thus, the yellow surface represents the correctly identified zone (true positive), the blue surface the wrongly unidentified zone (false negative) and finally, the red surface represents the zone wrongly identified by Algorithm 1 as the zone to be detected (false positive).

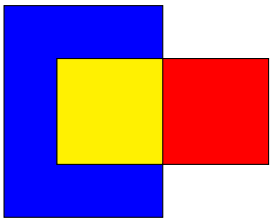

Fig. 5. Example of different areas, illustrating the IOU and IOE metrics

In this example, the IOU score is equal to the yellow area divided by the sum of all areas, i.e. $(A_J/(A_B + A_R + A_J))$.

The IOE score, on the other hand, is equal to the yellow area divided by the sum of the blue and yellow areas: $(A_J/(A_B + A_J))$.

We have computed these scores for all permutations of all the five criteria, as well as for each subset of criteria. For example, if we consider the order [Distance, Alignment, Height, Text Color, Background Color], we calculate a score where clusters are calculated using the first criterion alone (no refinement), another with the first two criteria, then the first three, and so on. In this way, we have been exhaustive, so that we can validate which set of criteria is the most suitable. For each map, we obtain a total of $\sum_{i=1}^{5} \prod_{j=i}^{5} j$ scores, i.e. 325 scores.

Among the computed *clusters*, we select those ones with the highest evaluation metric score.

B. Results

In the rest of this paper, we will use the following abbreviations for the 5 criteria presented in Section III-C:

- D: distance – equation (5)
- A: alignment – equation (3)
- H: height – equation (7)
- T: text color – equation (8)
- F: background color – equation (6)

1) Single-Criteria Clustering: Initially, we simply performed a *clustering* following a single criterion, without any further refinement. The average IOUs for this first test are shown in Table II.

We observe that the criterion providing the best results is distance (D). This result is not particularly surprising, given

D	A	H	T	F
42,8%	28,2%	9,8%	14,6%	15,1%

TABLE II
IOU AVERAGES OVER THE 29 MAPS OF A SINGLE-CRITERIA *clustering*

that, in the large majority of cases, a legend groups together a set of texts in a spatial sub-area of a map. Obviously, the proximity of these texts favors the distance criterion.

Conversely, the least favorable criterion -when considered individually- is the one of height (H). This result though, is less expected, but can be explained by the fact that height is not a particularly discriminating factor for distinguishing the textual elements of a legend. Indeed, the texts in a legend may have different sizes, but other map elements may have the same size as well. Hence, it seems inappropriate to consider height as a single criterion.

2) The importance of multi-criteria refinement: We proceeded our experimental study by evaluating the effect of combining different criteria via the successive refinement procedure presented in Section IV.

Here, we aim to show the value evolution of successively refining the cluster considering various criteria, regardless of the order in which they are taken. For that purpose, we have considered 5 classes of results, depending on the number of criteria used. All possible orders were tested, and the following results show an average of these results.

Figure 6 focuses on IOU metric, and represents the number of maps in our dataset where, on average, a run has produced a result greater than the value displayed on the abscissa.

For example, the values for the abscissa "10%" indicate that using just one criterion out of the 5 (without refinement, as in the previous section) yields an average IOU greater than 10% for 24 of the 29 maps, while, when using 2,3,4 or 5 criteria we detect 27 maps. Of course, the higher the values on the x-axis, the lower the number of maps concerned.

Figure 6, shows that the number of criteria taken into account has an impact on the quality of the result. In particular, it is important to note that each refinement by a new criterion improves the results. Once again, this is true independently of the order in which the refinements are done, as the figures are drawn on the averages of all possible orders. It is worth noticing that, exploiting the different characteristics of legend text elements is essential for the effectiveness of the legend text detection procedure.

Figure 7, follows the same principle, but for the IOE metric this time. Evidently, the results here are much better. This is easily explained by the metric chosen, since IOE, by construction, is more permissive (see Figure 5).

The greater values displayed the latter figure, illustrate that our algorithm detects legends rather correctly in a large number of cases. The differences with the IOU in Figure 6 indicate, however, that our technique has a tendency to

Fig. 6. Chart of the percentage of average IOU map above the x-axis

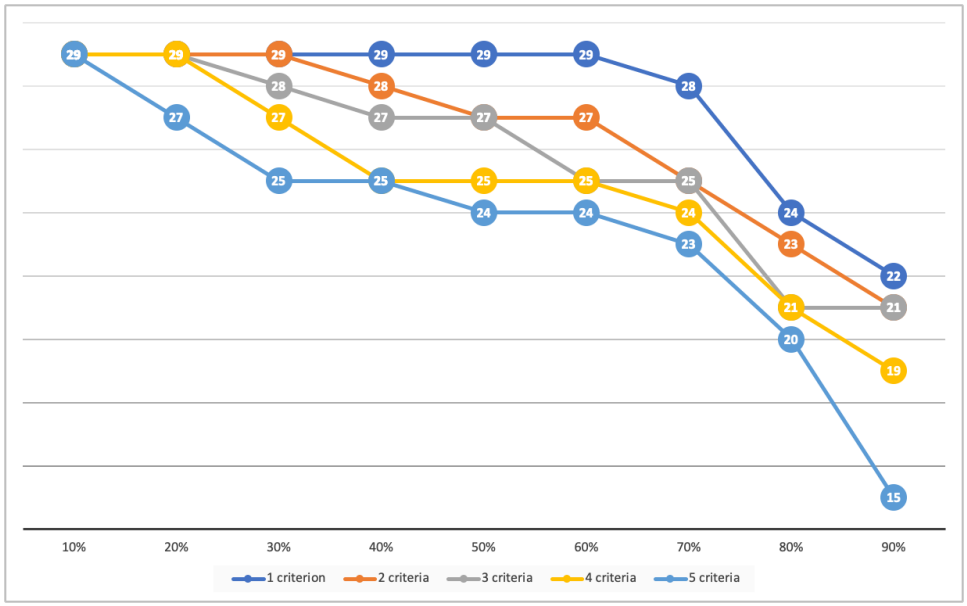

Fig. 7. Chart of the percentage of average IOE map on the x-axis

"over-approximate" the legend frame, as this metric precisely punishes overly broad approximations.

3) Best combination of criteria: Although the previous section disregarded the order of the considered criteria in successive cluster refinements, this operation impacts also the performance of our procedure. Here, we focus on the run that has had, on average, the best results.

In line with our previous observations, this run uses all 5 criteria to perform its detection. Intuitively, we might imagine that the distance criterion (D), the best in a mono-criteria *clustering*, would be in first place in the order considered. Surprisingly, this is not ttue. The optimal order we obtained by the order F-A-D-H-T. So, grouping first according to background color, then refining successively with alignment, distance, height and finally text color, we obtain our best results. In this configuration, the average IOU obtained across all maps is 52%. On the other hand, 7 maps (out of the 29) achieve an IOU above 80%. The average IOE obtained for this test is 79.48%. Recall at this point that, our dataset is chosen to be heterogeneous, containing maps that are challenging for this type of detection. This choice was done in order to show that our algorithm is generic and independent of the nature of the data and explains why the detection scores may seem low. In particular, several maps do not conform to the assumptions we have made (such as left-hand alignment).

We have shown results of a particular order (i.e., F-A-D-H-T), however, other orders on the criteria are also in-

teresting, and thus, some *patterns* can to be extracted. The full list of results obtained during our experimental phase is available at https://www.cril.univ-artois.fr/~marzinkowski/incremental-clustering-results/.

VI. Conclusion and future work

In this paper, we have presented how clustering techniques, coupled with appropriate criteria, can be used to automatically detect legend texts in maps of all kinds: city maps, water networks, road maps, etc.

To efficiently group text areas that could represent texts in a legend, we established five criteria, each one associated with a specific distance measure. We then introduced an incremental algorithm, parameterized by a vector of criteria, enabling us to improve partitioning by exploiting entropy as a measure of the homogeneity of the partitions obtained at each stage. The experimental study, carried out on a highly representative sample of maps using one or more criteria, showing that successive refinements of the partitions yielded better results than the use of a single criterion.

A future task would be to study methods for ordering the different clusters obtained by our clustering algorithm. The ultimate goal is to indicate, among a set of clusters returned by our clustering algorithm, which ones are most likely to represent legend texts. This amounts to ranking the clusters and selecting the best one(s). If we consider only text boxes (and not other elements of the images), we can identify three main factors that influence the selection of the best cluster to represent a legend. The first is the number of text boxes contained within a cluster. There is a typical range for the number of text boxes that a legend usually contains. If a cluster contains only one text box, it is unlikely to be a legend, and the same applies if the cluster exceeds a certain threshold. This threshold depends on the total number of text boxes in the image and the nature of the image itself. It can also be approximated by the number of representative objects that appear frequently enough in the image. The second factor is the alignment of the text boxes. If the boxes are well-aligned, this suggests an arrangement that often corresponds to a legend. In this paper, we have mainly considered left alignment and we aim to maintain this assumption. The alignment distance we defined for the clustering algorithm was initially designed for only two text boxes. Our objective is to extend this distance measure in order to be applied to clusters consisting of multiple text boxes. The third criterion is the distance between successive text boxes: text boxes that are too far apart are less likely to form a coherent legend than those that are close together.

Other future work, includes generalizing the proposed distances to accommodate various legend text formats, developing a global aggregation function for distances across the five criteria to enable single-run clustering, and exploring alternative clustering algorithms, particularly hierarchical clustering. We also plan to collect a large number of legend maps for further experimental study.

Acknowledgments: This work was supported by the Horizon Europe Marie Skłodowska-Curie Actions MSCA (Staff Exchanges) grant agreement 101086252; Call: HORIZON-MSCA-2021-SE-01, Project: STARWARS (STormwAteR and WastewAteR networkS heterogeneous data AI-driven management). This work has also received support from the Agence Nationale de la Recherche, via the ANR CROQUIS project (Collecte, représentation, complétion, fusion et interrogation de données de réseaux d'eau urbains hétérogénes et incertaines), grant ANR-21-CE23-0004.

References

[1] C. Neudecker, K. Baierer, M. Gerber, C. Clausner, A. Antonacopoulos, and S. Pletschacher, "A survey of ocr evaluation tools and metrics," ser. HIP '21. Association for Computing Machinery, 2021.

[2] A. Agrahari and R. Ghosh, "Multi-oriented text detection in natural scene images based on the intersection of mser with the locally binarized image," *Procedia Computer Science*, vol. 171, pp. 322–330, 2020.

[3] C. Yi and Y. Tian, "Text string detection from natural scenes by structure-based partition and grouping," vol. 20(9), 2011, pp. 2594—-2605.

[4] D. Kavitha and V. Radha, "Text detection based on text shape feature analysis with intelligent grouping in natural scene images," in *Mathematical Modeling and Computational Tools*, S. Bhattacharyya, J. Kumar, and K. Ghoshal, Eds. Singapore: Springer Singapore, 2020, pp. 467–479.

[5] Y. Djenouri, A. Belhadi, P. Fournier-Viger, and J. C.-W. Lin, "Fast and effective cluster-based information retrieval using frequent closed itemsets," *Information Sciences*, vol. 453, pp. 154–167, 2018.

[6] R. Kumbhar, S. Mhamane, H. Patil, S. Patil, and S. Kale, "Text document clustering using k-means algorithm with dimension reduction techniques," in *5th International Conference on Communication and Electronics Systems (ICCES)*, 2020, pp. 1222–1228.

[7] R. Khan, Y. Qian, and S. Naeem, "Extractive based text summarization using kmeans and tf-idf," *International Journal of Information Engineering and Electronic Business*, vol. 11, pp. 33–44, 05 2019.

[8] Y.-Y. Chiang and C. A. Knoblock, "An approach for recognizing text labels in raster maps," in *20th International Conference on Pattern Recognition*, 2010, pp. 3199–3202.

[9] J. Lintern, "Recognizing text in google street view images," 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:8464903

[10] Mindee, "doctr: Document text recognition," https://github.com/mindee/doctr, 2021.

[11] J. R. Parker, *Algorithms for Image Processing and Computer Vision*, 2nd ed. Wiley Publishing, 2010.

[12] M. Stricker and M. Orengo, "Storage and retrieval for image and video databases (spie) - similarity of color images," *SPIE Proceedings*, vol. 2420, pp. 381–392, March 1995.

[13] H. Yin, A. Aryani, S. Petrie, A. Nambissan, A. Astudillo, and S. Cao, "A rapid review of clustering algorithms," 2024.

[14] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[15] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained k-means clustering with background knowledge." in *ICML*, 2001, pp. 577–584. [Online]. Available: http://www.litech.org/~wkiri/Papers/wagstaff-kmeans-01.pdf

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[17] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. D. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 658–666. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Rezatofighi_Generalized_Intersection_Over_Union_A_Metric_and_a_Loss_for_CVPR_2019_paper.html