Approche incrémentale pour la détection des textes de légendes dans des cartes numériques

Arthur Marzinkowski, Salem Benferhat, Anastasia Paparrizou, Cédric Piette

Approche incrémentale pour la détection des textes de légendes dans des cartes numériques

Arthur Marzinkowski[1], Salem Benferhat[1], Anastasia Paparrizou[2], Cédric Piette[1]

[1] Centre de Recherche en Informatique de Lens, CRIL
[2] Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, LIRMM

{marzinkowski,benferhat,piette}@cril.fr paparrizou@lirmm.fr

Résumé

Cet article concerne la détection automatique des textes de légende dans les cartes. Après avoir extrait les textes des images, avec des outils OCR, nous utiliserons un processus itératif de regroupement ("clustering") des textes extraits. Cinq critères principaux, avec différents niveaux d'importance, sont utilisés : l'alignement des textes, la distance entre les zones de texte, la couleur de fond des textes, la couleur des textes et la taille des caractères. Pour chacun des critères, des mesures de similarité appropriées sont définies. Nous proposons une méthode qui combinerait de manière hiérarchique les regroupements obtenus à partir de chaque critère. L'étude expérimentale révèle deux résultats importants. Premièrement, l'utilisation de plusieurs critères donne des résultats supérieurs à ceux d'une simple distance (e.g., euclidienne) entre les zones de texte. Deuxièmement, l'étude expérimentale confirme l'efficacité globale de la relation de priorité que nous avons intuitivement définie entre les critères pour détecter à partir des textes de la légende.

Mots-clés

Cartes numériques, OCR, clustering

Abstract

This paper deals with the automatic detection of legend texts inside maps. After extracting the texts from the images, using OCR tools, we make use of an iterative clustering process of 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, in an incremental way, the partitions obtained by each criterion. The experimental study reveals two important results. First, the combination of several criteria gives better results than just considering a simple distance (e.g., Euclidean distance) between text boxes. Secondly, the overall effectiveness of the priority relation, which we have intuitively defined between the criteria for detect with caption texts, is confirmed.

Keywords

Digital maps, OCR, clustering

1 Introduction

La détection automatique de textes à l'intérieur d'images est étudiée par la communauté de la vision par ordinateur depuis plusieurs décennies. La détection de texte et le regroupement de texte est principalement basée sur des caractéristiques visuelles extraites de l'image (c'est-à-dire la couleur, la forme, etc.) [5]. Ce travail concerne la détection de texte dans des images appartenant à la légende d'une carte numérique arbitraire, et n'est donc pas lié à un domaine d'information spécifique. La nouveauté de ce travail est que non seulement des caractéristiques visuelles sont utilisées, mais également des notions sémantiques qui accompagnent et facilitent la détection des zones de légende. Par exemple, une information sémantique peut être : la position possible de l'objet sur la carte, la relation ou la distance entre les objets voisins, etc.

Les images que nous traitons sont des images qui représentent des cartes numériques. Ce travail constitue une étape primordiale dans la détection d'objets d'intérêt à l'intérieur des cartes, qui sont généralement définis et affichés dans la légende. Pour ce faire, nous commençons par utiliser un algorithme de reconnaissance optique de caractères (OCR) afin d'obtenir les zones de texte qui apparaissent à l'intérieur d'une carte donnée. Nous utilisons les régions de texte fournies par l'algorithme OCR comme entrées dans notre algorithme de regroupement (*clustering*) k-means [16]. Nous introduisons également des critères qui caractérisent de manière symbolique ou visuelle la notion de légende ; par exemple, des phrases de texte alignées verticalement, pouvant former une légende. Nous définissons cinq de ces critères ainsi que les mesures de distance associées à chaque critère pour l'algorithme des k-means. Nous déployons la méthode des k-moyennes de manière incrémentale, où les *clusters* sont construits en fonction d'un critère pris en compte à chaque itération. L'ordre dans lequel les critères sont considérés joue donc un rôle crucial dans l'efficacité de notre algorithme. À chaque itération, notre algorithme doit décider s'il faut ou non diviser (ou éclater) chaque "cluster" obtenu à l'itération précédente. Cette déci-

sion est basée sur l'entropie que nous utilisons pour refléter l'homogénéité d'une partition.

Nous évaluons les résultats de la détection de la légende sur plusieurs cartes de tailles et de styles différents à l'aide de deux mesures d'évaluation. Les résultats montrent que notre algorithme détecte bien la région de la légende sur plusieurs cartes. L'analyse de notre étude exhaustive des critères indique que lorsque plusieurs critères sont pris en compte, nous obtenons une meilleure détection de la région de la légende. D'autres observations seront présentées dans la section consacrée à l'étude expérimentale.

Le reste de l'article est organisé comme suit. La section suivante présente brièvement certains travaux existants. La section 3 décrit le problème considéré dans cet article, définit les cinq critères utilisés pour le regroupement et présente notre algorithme de détection du texte de la légende. La section 4 contient les résultats des études expérimentales réalisées et enfin, la section 5 conclut cet article.

2 Travaux antérieurs

La détection ou la reconnaissance de texte concerne généralement les documents texte et de nombreux algorithmes OCR ont été proposés à cet effet avec une excellente précision [11]. Ces dernières années, la détection de texte dans les images de scènes naturelles a suscité beaucoup d'intérêt en raison d'une variété d'applications : compréhension de scènes, récupération d'images basées sur le contenu, aide à la navigation (pour les voitures autonomes ou les personnes malvoyantes) [1]. Dans [17], les auteurs ont proposé un cadre de détection de chaînes de texte, dans lequel le regroupement des caractères candidats est basé sur des caractéristiques structurelles, telles que les différences de taille des caractères, les distances entre les caractères voisins et l'alignement des caractères. Leur méthode de regroupement de lignes de texte effectue une transformation de Hough pour ajuster la ligne de texte aux centroïdes des candidats de texte au lieu de les regrouper. Ce travail est similaire au nôtre dans le sens où les notions sémantiques sont prises en compte, mais l'objectif et la granularité sont différents (regroupement de caractères et de lignes, et non regroupement de zones de texte). De même dans [6], le système d'assistance proposé reconnaît le texte dans une image en fonction de caractéristiques structurelles telles que la taille, l'orientation et la distance entre les régions d'intérêt successives.

Dans la communauté de la recherche d'informations, il y a un effort considérable pour le regroupement de texte avec des k-means, mais à des fins différentes des nôtres (c'est-à-dire, pour noter et classer la pertinence d'un document compte tenu d'une requête utilisateur [3], pour regrouper des documents de contenu similaire [8] ou pour résumer du texte [7]). Il existe des travaux sur la détection de texte dans les cartes *raster*, mais ils sont davantage liés aux approches OCR, où le défi est dû aux différentes orientations du texte et au chevauchement des étiquettes de texte [2, 9]. À notre connaissance, il n'existe aucun travail traitant de la détection de texte appartenant aux légendes des cartes.

3 Présentation du problème et critères de regroupement

3.1 Présentation du problème

Le problème que nous cherchons à résoudre est celui d'identifier la région d'une carte où se situent les textes de légende. Les entrées de notre algorithme sont des images qui représentent des cartes avec des légendes. Les cartes sur lesquelles nous travaillons sont constituées de figures mais aussi de zones de texte. Dans cet article, nous proposons une approche pour discriminer le texte appartenant à une légende potentielle des autres régions de texte de l'image.

Chaque image sera représentée par une matrice $n \times m$ éléments, que nous noterons par la suite par $\mathscr{I}$. Chaque élément de la matrice, représente une couleur d'un pixel (représentée ici dans le format RGB ; c'est-à-dire un triplet d'entiers compris entre 0 et 255). Nous utiliserons aussi x_i pour désigner un numéro d'un ligne et y_j pour désigner le numéro d'une colonne de la matrice $\mathscr{I}$.
Dans cet article, nous nous concentrons sur les images contenant des légendes. En particulier, nous nous intéressons à la zone de la carte qui contient le texte de la légende et qui sera également représentée par une matrice de couleurs de pixels notée $\mathscr{L}$. La sortie de notre algorithme est une zone de la matrice $\mathscr{I}$ qui est censée représenter la zone de texte de la légende $\mathscr{L}$.
Nous présentons maintenant deux notions qui seront utilisées plus tard par notre algorithme. La première notion, que nous appelons éclatement, consiste à construire une partition d'un ensemble à partir d'une partition plus grande. La seconde est un rappel de la notion de l'entropie qui servira à mesurer l'homogénéité d'une partition.

Definition 1 (Eclatement) *Soit A un ensemble d'éléments. Soit B un sous-ensemble de A et $\mathscr{P}_A$ une partition de A. Nous appelons l'éclatement de B par $\mathscr{P}_A$, noté $B \rhd \mathscr{P}_A$, la partition de B obtenue en intersectant chaque élément de $\mathscr{P}_A$ avec B. Plus formellement :*

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

Definition 2 (Mesure d'homogénéité) *Soit A un ensemble d'éléments et $\mathscr{P}_A$ une partition de A. Nous définissons l'homogénéité (ou l'entropie) de $\mathscr{P}_A$, notée $\mathscr{E}(\mathscr{P}_A)$, par :*

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

où $\|x\|$ représente la cardinalité de x.

3.2 Extraction des textes depuis les cartes

La première étape de notre algorithme consiste à extraire des textes depuis les cartes. Dans cette étape, nous utiliserons simplement les outils OCR existants. Nous utilisons notamment l'outil OCR DocTr [10] qui permet d'obtenir à partir d'une image une liste de zones de texte avec

différents niveaux de granularité (des blocs de texte, une ligne de texte, un mot, etc.).

A partir d'une carte $\mathscr{I}$, l'outil OCR retourne un ensemble de texte, noté $\mathscr{T}_{\mathscr{I}}$. Les éléments de $\mathscr{T}_{\mathscr{I}}$ sont appelés des boîtes de textes et sont dénotés par les lettres calligraphiques minuscules $\mathscr{a}, \mathscr{b}, \mathscr{c}, etc.$

La Figure 1 illustre un exemple de zone de texte $\mathscr{a}$ qui est représentée par un rectangle sur la carte identifié par deux points aux extrémités d'une diagonale $(x^{\mathscr{a}}, y^{\mathscr{a}})$ et $(x_{\mathscr{a}}, y_{\mathscr{a}})$.

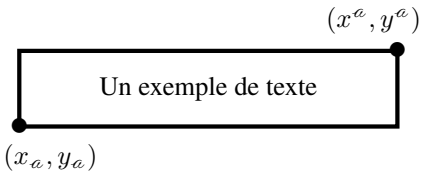

FIGURE 1 – Un exemple de boîte de texte représentée par un rectangle

Remarques :

— Dans le reste de cet article, nous supposons que tous les rectangles associés aux boîtes de texte sont disjoints.
— Nous excluons aussi les rectangles qui ne sont pas verticaux ou horizontaux (par rapport aux axes de l'image) du fait que les légendes ne sont pas inclinés.

3.3 Définitions des critères de regroupement

Cette sous-section présente les cinq principaux critères qui seront utilisés par notre méthode de détection des zones de texte de légende. Pour chacun des cinq critères, nous définissons les mesures de similarité associées pour comparer deux zones de texte. Soit $\mathscr{a}$ et $\mathscr{b}$ deux zones de texte.
Par abus de langage, nous utiliserons le terme distance pour exprimer la similarité entre deux zones de texte, sans exiger de propriété préalable sur la mesure de distance utilisée.

Critère 1 : l'alignement des textes

Le texte d'une légende est souvent aligné. L'alignement du texte peut prendre différentes formes : à gauche, au centre, à droite, etc. Par souci de simplicité, nous nous limitons uniquement à la situation des légendes avec un texte aligné verticalement à gauche. L'extension à d'autres formes d'alignement se fait facilement par symétrie ou par une légère adaptation de la mesure de distance définie ci-dessous. La distance associée à l'alignement gauche, notée d_{ag}, est définie par :

$$d_{ag}(\mathscr{a}, \mathscr{b}) = \mid y_{\mathscr{a}} - y_{\mathscr{b}} \mid . \qquad (3)$$

Critère 2 : la distance entre les textes

Un deuxième critère naturel est celui de la mesure de similarité entre les zones de texte que nous noterons d_e. Les textes des légendes sont en général proches les uns des autres.

Une première idée pour définir la distance entre deux boîtes de texte est de considérer la plus petite distance entre deux points quelconques des périmètres de rectangles associés aux deux boîtes de texte. Cette solution n'est pas satisfaisante dans notre cadre car nous utilisons des rectangles particuliers (disjoints, horizontaux ou verticaux, texte alignés à gauche, etc.).

La définition de la distance entre deux boîtes dépend clairement des résultats des OCR qui peuvent contenir des phrases entières ou simplement des mots. Nous proposons d'analyser chacune de ces deux situations.

Commençons par le cas où les boîtes contiennt des phrases entières. La figure 2 donne la situation "idéale" dans laquelle la distance entre deux boîtes $\mathscr{a}$ et $\mathscr{b}$ est égale à 0. Il s'agit de deux boîtes parfaitement alignées à gauche (nous rappelons pour des raisons de simplicité seul l'alignement à gauche est considérée) et dont la distance est égale à l'unité (un seul pixel dans notre cas).

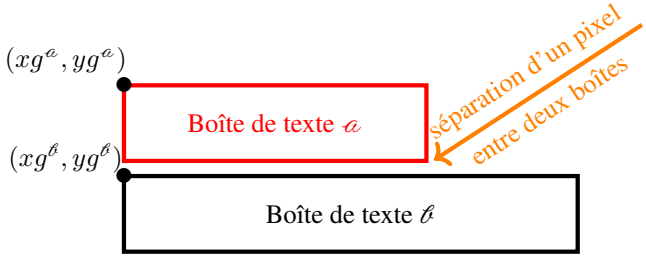

FIGURE 2 – Situations où deux boîtes sont considérées le plus proches possible

Plus formellement, soit $\mathscr{a}$ ey $\mathscr{b}$ deux boîtes de texte disjointes alors :

$$d_e(\mathscr{a}, \mathscr{b}) = 0$$
ssi
$$(y_{\mathscr{a}} = y_{\mathscr{b}})$$
et
$$[(xg^{\mathscr{a}} = x_{\mathscr{b}} + 1) \text{ ou } (xg^{\mathscr{b}} = x_{\mathscr{a}} + 1)].$$

Notons que la distance ne s'applique qu'avec deux boîtes de texte disjointes (c'est le cadre de notre article).

Sur la base de cette définition de situations où deux boîtes de texte sont considérées comme idéalement proches, il suffit alors de définir la distance comme la tranlation nécessaire d'une des deux boîtes pour atteindre cette situation idéale.

Plus formellement, à partir des notations données dans la figure 3, nous avons :

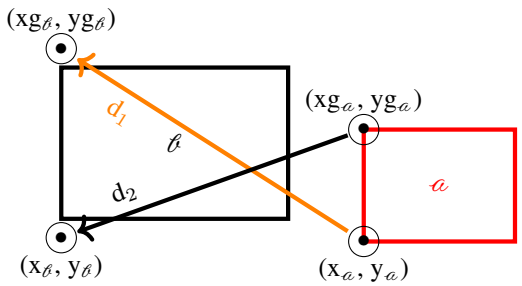

FIGURE 3 – Illustration du calcul de la distance entre deux boîtes de texte

$$d_e(a,b) = min(d_1, d_2), \text{avec}$$
$$d_1 = \sqrt{(x_a - xg_b)^2 + (y_a - yg_b)^2}, et \quad (4)$$
$$d_2 = \sqrt{(x_b - xg_a)^2 + (y_b - yg_a)^2}.$$

Bien sûr, d'autres mesures peuvent être utilisées (comme la distance de Manhattan) à la place de la distance euclidienne.

Maintenant dans l'hypothèse où les boîtes de texte contiennent uniquement des mots, dans ce cas la distance (euclidienne) entre les centres des mots est suffisante. Notons (xc_a, yc_a) et (xc_b, yc_b) les coordonnées des points qui représentent les centres des boîtes a et b respectivement. La distance entre a et b est simplement définie par :

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

Critère 3 : la couleur des fonds des boîtes de texte

Le troisième critère est la couleur de fond de la zone de texte. En effet, dans les légendes le fond utilisé est souvent de couleur homogène. L'utilisation de ce critère soulève deux questions, heureusement bien abordées dans la littérature sur le traitement d'images (e.g., [12]).

La première question est de savoir comment déterminer l'arrière-plan d'une boîte de texte a. Dans notre contexte, on peut simplement utiliser la fréquence de couleur des pixels car on peut raisonnablement supposer que les textes dans les cases à comparer sont de couleur homogène et que les textes occupent la majorité de la case. Une autre façon de procéder, qui est celle utilisée dans l'article, est de calculer la partie la plus connectée (connexes) de la boite (deux pixels sont connectés s'ils sont de couleur homogène, c'est-à-dire que les deux pixels sont de couleur suffisamment similaire) en partant de l'un des bords de la boîte de texte.

La deuxième question est de savoir comment déterminer la couleur représentative du fond du texte. Là encore, nous avons plusieurs possibilités (e.g. [15]). Une possibilité à partir des valeurs de RGB de chaque pixel de la zone de texte, est de i) calculer d'abord la somme des carrées des couleurs (valeur par valeur), iii) puis diviser le résultat par le nombre de pixels et iii) enfin appliquer la racine carrée au résultat (pour rester dans l'ensemble $\{0, ..., 255\}$). Une autre possibilité est tout simplement de prendre la moyenne des couleurs (valeur par valeur) des différents pixels de du fond de l'image. C'est cette méthode qui est utilisée dans cet article.

Notons maintenant (RF_a, GF_a, BF_a) et (RF_b, GF_b, BF_b) les couleurs (en moyenne) du fond de texte des boîtes de texte a et b respectivement. Reste maintenant à calculer la proximité entre ces deux couleurs où plusieurs définitions sont possibles. Dans cet article, nous utilisons la distance euclidienne, notée d_{bg}, et définie par :
$d_{bg}(a,b) =$

$$\sqrt{(RF_a - RF_b)^2 + (GF_a - GF_b)^2 + (BF_a - BF_b)^2}. \quad (6)$$

Critère 4 : la hauteur des boîtes de texte

Les trois critères décrits ci-dessus (alignement, distance et fond de la zone de texte) sont fondamentaux pour déterminer la zone de texte d'une légende. Un autre critère concerne la taille de la police des caractères utilisée qui est ici supposée égale à la hauteur de la zone de texte.
La quatrième distance, notée d_t, donnée par la hauteur des boîtes de texte est simplement définie par :

$$d_t(a,b) = |\ (|\ x_a - x^a\ |) - (|\ x_b - x^b\ |)\ | \quad (7)$$

Critère 5 : la couleur du texte

Ce dernier critère est le dual du critère 3 (la couleur du fond d'une boîte de texte), puis que l'on considère qu'une boîte de texte peut-être divisée en deux partie : la partie qui contient les caractères de la boîte de texte et le reste qui représente le fond de la boîte de texte. Notons (RT_a, GT_a, BT_a) et $(RT_b, GT_b, BT_b\ )$ les couleurs (en moyenne) des textes des boîtes de texte a et b respectivement. Alors la distance par rapport à la couleur des textes, notée d_{ct}, est définie par :
$d_{ct}(a,b) =$

$$\sqrt{(RF_a - RT_b)^2 + (GT_a - GT_b)^2 + (BT_a - BT_b)^2}. \quad (8)$$

3.4 Algorithme par raffinements successifs des résultats de regroupement

Nous avons choisi une approche à base de *clustering* [18, 4]. Comme nos boîtes de texte extraites depuis les images ne sont pas étiquetées, nous utiliserons des méthodes basées sur l'apprentissage non-supervisée. Plus précisément, dans cet article nous avons opté pour l'algorithme k-means (e.g., [16]) largement utilisé dans des problèmes de classification non-supervisé.
Nous cherchons à regrouper des régions de texte de telle manière que ces regroupements représentent le texte d'une

légende. Les données d'entrée de notre algorithme sont avant tout une carte dont nous supposons qu'elle contient une légende. Cette carte sera notée $\mathscr{I}$. A ces données, nous ajoutons deux paramètres. Le premier est un tableau, noté $\vec{c}$, de critères (de taille 1 à 5) qui indique l'ordre dans lequel les critères doivent être utilisés.
Le deuxième paramètre est un fonction seuil, noté $\sigma(\mathscr{A})$ qui indique si une partition d'un ensemble $\mathscr{A}$ est suffisamment homogène ou non. Comme pour toute fonction de collecte, la question difficile est de savoir comment fixer ce seuil. Dans notre étude expérimentale, il est fixé à 80 % de la valeur maximale.

A ces deux paramètres s'ajoutent deux autres fonctions. La première $OCR(\mathscr{I})$ qui retourne $\mathscr{T}_\mathscr{I}$ l'ensemble des boîtes de texte qui se trouvent dans l'image $\mathscr{I}$). La deuxième fonction est la méthode de regroupement utilisée. Comme nous l'avons indiqué plus haut, nous utilisons simplement la méthode de k-means.

Algorithm 1 Algorithme de détection de légendes

Entrées :
- $\mathscr{I}$: une carte avec une légende
- $\vec{c}$: un vecteur de $n \in \{1, ..., 5\}$ critères
- σ : une fonction qui prend une partition et qui retourne un réel
- k-means : la méthode de regroupement utilisée

Sorties :
- $\mathscr{P}$ Un ensemble de *clusters* .

1: *// D'abord l'outil OCR est appliqué sur l'image*
2: $\mathscr{T}_\mathscr{I} \leftarrow OCR(\mathscr{I})$
3: *// On applique l'algorithme k-means sur* $\mathscr{T}_\mathscr{I}$
4: *// et le premier critère* $\vec{c}[1]$
5: $\mathscr{P} \leftarrow k - means(\mathscr{T}_\mathscr{I}, \vec{c}[1])$
6: *// On raffine l'ensemble* $\mathscr{P}$ *itérativement avec les autres critères*
7: **for all** $i \in \{2, ..., n\}$ **do**
8: *// On applique l'algorithme K-means sur* $\mathscr{T}_\mathscr{I}$
9: *// et le ième critère* $\vec{c}[i]$
10: $\mathscr{C} \leftarrow k - means(\mathscr{T}_\mathscr{I}, \vec{c}[i])$
11: *// On raffine chacun des éléments de*
12: *// la partition courante* $\mathscr{P}$
13: *// On stocke les résultats dans* $\mathscr{X}$
14: $\mathscr{X} \leftarrow \emptyset$
15: **for all** $B \in \mathscr{P}$ **do**
16: *// On éclate B* $\mathscr{C}$ *en utilisant Definition 1*
17: $\mathrm{R}_B = B \triangleright \mathscr{C}$
18: *// On vérifie le raffinement de B, i.e. si* R_B *est homogène*
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: $\mathscr{P} \leftarrow \mathscr{X}$
return $\mathscr{P}$

Notre algorithme détaillé ci-dessous est composé de trois étapes principales.

— La première étape (lignes 1 et 2 de notre algorithme) consiste tout simplement à extraire les boîtes de texte depuis l'image grâce à l'outil OCR. Le résultat est un ensemble de boîtes de texte.
— La deuxième étape (ligne 3 et 4) consiste à appliquer l'algorithme k-means sur l'ensemble des boîtes de texte. La partition obtenue est notée.$\mathscr{P}$.
— La troisième étape (lignes 7-21) consiste à raffiner, de manière progressive, le regroupement obtenu avec chacun des critères restants. Au préalable, pour chaque critère, l'algorithme k-means est appliqué sur l'ensemble des boîtes de textes donné par l'algorithme OCR. Ensuite, à chaque élément B de regroupement est éclaté (selon la Définition 1) avec le résultat de regroupement des boîtes de texte avec l'algorithme k-means (ligne 10). Si l'entropie associée au résultat de l'éclatement, alors l'élément B est remplacé par son éclatement (lignes 18-21).

Nous utilisons donc cette mesure d'entropie pour évaluer l'homogénéité des différentes classes entre un *cluster* sur un critère donné et un ensemble de *clusters* d'un autre critère.

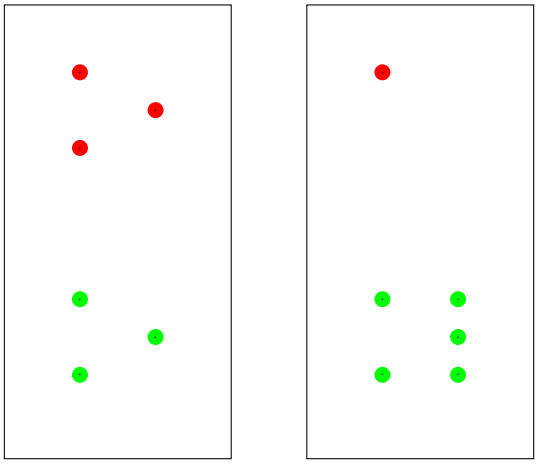

FIGURE 4 – Exemples de deux situations et leurs valeurs d'entropie

Dans l'exemple ci-dessus, chaque point appartient à un même *cluster* pour le critère 1. Les couleurs représentent les différentes classes calculées pour le critère 2.
Pour l'exemple de gauche, qui représente un *cluster* donnée par le critère 1, trois points appartiennent à la classe "rouge" et les trois autres appartiennent à la classe "verte". Les différentes classes du critère 2 au seins de ce *cluster* sont hétérogènes. La valeur d'entropie est maximale (c'est-à-dire de 1).
L'exemple à droite n'a qu'un seul élément appartenant à la classe "rouge". Dans cet exemple, les classes du critère 2 sont beaucoup plus homogènes. En conséquence, la valeur d'entropie de ce *cluster* est plus basse (0,65).

Dans le premier cas, le *cluster* est séparé en deux nouveaux clusters, alors que dans le deuxième cas le *cluster* reste inchangé.

4 Évaluation expérimentale

Les expérimentations conduites dans cet article sont réalisées sur des processeurs `Intel XEON E5-2637 v4` 4 cœurs à 3,6 GHz avec 128 Go de RAM RDIMM, sous `CentOS 7.3 Kernel 3.10`. Concernant la configuration logicielle, nous utilisons la bibliothèque OCR `DocTr version 4.0`[10] et l'implémentation du *clustering* `scikit-learn version 1.3.2` [13].

Nous avons sélectionné 29 images sur lesquelles nous avons exécuté notre procédure de détection de la légende, en utilisant les différents critères présentés plus tôt. Sur chacune de ces cartes, la légende a été étiquetée à la main, afin de comparer le résultat de nos calculs à la "véritable" légende, qui est la réponse optimale. Nous avons pris soin de sélectionner des cartes très différentes les unes des autres, dans leur composition, leur résolution, et la représentation de leur légende. L'ensemble des cartes utilisées dans cette section est disponible à l'adresse `https://www.cril.univ-artois.fr/~marzinkowski/incremental-clustering-results/`

Nous précisons ici que nous comparons des surfaces (ou des zones) de l'image, car ce qui nous intéresse est de savoir si une zone est une légende ou non. L'un des avantages de procéder ainsi, est que si un mot (à l'intérieur d'un groupe de mots) n'est pas détecté par l'OCR, on peut néanmoins récupérer une large partie de la zone (sinon la zone complète si le mot est à l'intérieur d'un groupe de mots) dans notre détection, et ainsi ne pas être trop sensibles aux résultats de l'OCR.

Nous ne reportons pas dans le détail les temps d'exécution, qui sont similaires dans toutes nos expérimentation. Voici toutefois une idée générale du coût calculatoire des différentes étapes de notre algorithme de détection de légende : (i) le temps d'exécution de l'`OCR` sur notre ensemble de données varie entre 10 et 25 secondes, suivant la quantité de textes présents sur la carte donnée en entrée (ii) les cinq ensembles de des partitions obtenues avec k-means sont calculés en environ une seconde (iii) enfin, le temps nécessaire au raffinement n'excède jamais les 100ms. Une synthèse plus précise des temps d'exécution des différentes étapes est donnée dans Table 1.

On observe donc que quelque soit la carte, notre algorithme prend moins de 20 secondes pour effectuer toutes les étapes de la détection.

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

TABLE 1 – Présentation synthétique des temps d'exécution (en secondes)

4.1 Méthode d'évaluation

Nous avons considéré deux métriques d'évaluation :

— Intersection sur Union (IOU)
— Intersection sur Étiquetage (IOE)

Ces métriques sont calculées en comparant 2 zones rectangulaires de la carte. L'une est la zone calculée par notre algorithme, l'autre est la zone étiquetée manuellement comme réponse optimale.

Sommairement, l'*Intersection sur Union* (IOU), également appelée *indice de Jaccard*, consiste à calculer le ratio entre la surface de l'intersection des 2 zones, et celle de leur union. Pour l'*Intersection sur étiquetage* (IOE), il s'agit du ratio entre l'intersection des deux zones, et la zone étiquetée. Ces deux métriques ont l'intérêt d'être insensible à la résolution des cartes ; l'IOU a par ailleurs été utilisé dans de nombreux travaux [14].

Illustrons ces métriques à l'aide de la Figure 5. Celle-ci contient 3 zones de couleur : rouge, bleu et jaune ; nous noterons respectivement leur aire A_R, A_B et A_J.

On considère que le rectangle étiqueté (réponse optimale) est le rectangle composé des surfaces bleues et jaunes, tandis que la zone calculée par l'algorithme est représentée par les rectangles jaunes et rouges. Ainsi, la surface jaune représente la zone correctement identifiée (vrai positif), la surface bleue la zone indûment non identifiée (faux négatif) et enfin, la surface rouge représente la zone identifiée à tort par l'algorithme comme la zone à détecter (faux positif).

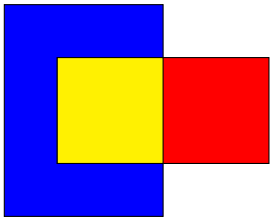

FIGURE 5 – Exemple de différentes surfaces, permettant d'illustrer les métriques IOU et IOE

Dans cet exemple, le score d'IOU est égal à l'aire en jaune divisée par la somme de toutes les aires, soit $(A_J/(A_B + A_R + A_J))$.

Le score d'IOE, quant à lui, est égal à l'aire jaune divisée par la somme des aires bleues et jaunes : $(A_J/(A_B + A_J))$.

Nous avons calculé ces scores pour toutes les permutations des cinq critères pris en compte, ainsi que pour chaque sous-ensemble de critères. Par exemple, si l'on prend en compte l'ordre [Distance, Alignement, Hauteur, Couleur du texte, Couleur de fond], nous calculons un score où les clusters sont calculés avec le seul premier critère (pas de raffinement), un autre avec les deux premiers critères, puis les trois premiers, etc. Ainsi, nous avons été exhaustifs, afin de connaître l'ensemble de critères le plus adapté de manière certaine. Nous obtenons donc pour chaque carte un total de $\sum_{i=1}^{5} \prod_{j=i}^{5} j$ scores, soit 325 scores.

Nous évaluons le résultat en sélectionnant, parmi les *clusters* calculés, ceux dont la métrique d'évaluation est la plus élevée.

4.2 Résultats

Dans la suite de cet article, nous utiliserons les abréviations suivantes pour les 5 critères présentés dans la Section 3.3 :

— D : la distance – équation (5)
— A : l'alignement – équation (3)
— H : la hauteur – équation (7)
— T : la couleur du texte – équation (8)
— F : la couleur de fond – équation (6)

4.2.1 Clustering Mono-Critère

Dans un premier temps, nous avons simplement réalisé un *clustering* en suivant un seul critère, sans aucun raffinement postérieur. Les IOU moyens de ce premier test sont disponibles dans la Table 2.

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

TABLE 2 – IOU moyens sur les 29 cartes d'un *clustering* mono-critère, pour chaque critère

On observe que le critère pourvoyeur des meilleurs résultats est la distance (D). Ce constat n'est pas spécialement surprenant, dans la mesure où une légende regroupe dans la large majorité des cas un ensemble de textes dans une sous-zone spaciale d'une carte ; la proximité de ces textes favorise ce critère de distance.
A l'inverse, le critère le moins favorable -pris individuellement- est la hauteur (H). Encore une fois, ce résultat semble peu surprenant, et peut s'expliquer par le fait que la hauteur n'est pas un élément particulièrement discriminant pour distinguer les éléments textuels d'une légende. Non seulement les textes d'une légende peuvent avoir des tailles différentes (hiérarchie de l'information, etc.), mais d'autres éléments de la carte peuvent avoir la même taille que ces textes particuliers. Il ne semble donc pas pertinent de considérer la hauteur comme seul critère.

4.2.2 De l'importance du raffinement multi-critère

Nous avons poursuivi nos expérimentations en évaluant l'intérêt de combiner nos différents critères via le raffinement successif présenté dans la Section 3.4.
Ici, nous avons voulu montrer l'intérêt du raffinement successif des différents critères, indifféremment de l'ordre dans lequel ceux-ci sont pris. Ainsi, nous avons considéré 5 classes de résultats, suivant le nombre de critères utilisés. Tous les ordres possibles ont été testés, et les résultats suivants indiquent une moyenne de ces résultats.
La Figure 6 se concentre sur l'IOU, et représente le nombre de cartes de notre jeu de données où, en moyenne, un lancement a donné un résultat supérieur à la valeur indiquée en abscisse.
Par exemple, les valeurs pour l'abscisse "10%" indiquent qu'utiliser un seul critère sur les 5 (sans raffinement, comme dans la section précédente) permet d'obtenir en moyenne un IOU supérieur à 10% pour 24 des 29 cartes, tandis qu'utiliser 2,3,4 ou 5 critères porte ce chiffre à 27. Bien sûr, plus les valeurs sur l'axe des abscisses sont élevées, plus le nombre des cartes concernées faiblit.

On note sur cette Figure 6 que le nombre de critères pris en compte a un impact sur la qualité du résultat fourni. En particulier, il est remarquable que chaque raffinement par un nouveau critère permet d'améliorer les résultats. Encore une fois, ceci est vrai indistinctement de l'ordre dans lequel les raffinements sont effectués, puisque ces chiffres indiquent des moyennes de tous les ordres possibles. Il est ici clair qu'exploiter les différentes caractéristiques des éléments textuels de la légende est essentiel à l'efficacité de la procédure de détection.
La Figure 7 reprend la même présentation, mais en s'intéressant cette fois à l'IOE. Assez clairement, les résultats sont ici bien meilleurs. Ceci s'explique aisément par la métrique choisie, IOE étant par construction plus permissive, comme vu dans l'exemple en Figure 5.
Les valeurs élevées représentées par cette Figure illustrent que notre algorithme détecte plutôt correctement les légendes dans de très nombreux cas. Les différences avec l'IOU de la Figure 6 indiquent toutefois que notre technique a tendance à "sur-approximer" le cadre de la légende, cette métrique sanctionnant précisément les approximations trop larges.

4.2.3 Meilleur lancement

Bien que la section précédente fasse abstraction de l'ordre des critères considérés dans les raffinements successifs, celui-ci a bien évidemment une certaine importance dans la performance de notre procédure. Nous nous intéressons ici au lancement qui a obtenu les meilleurs résultats en moyenne.
Conformément à nos observations précédentes, ce lancement utilise l'ensemble des 5 critères pour effectuer sa détection. Intuitivement, on pourrait imaginer que le critère de distance (D), le meilleur dans un *clustering* mono-critère, se trouve en première place dans l'ordre considéré. Or, ce n'est pas le cas. L'ordre optimal que nous avons obtenu est F-A-D-H-T.
C'est donc en regroupant d'abord suivant la couleur de fond (F), puis en raffinant successivement avec l'alignement, la distance, la hauteur et enfin la couleur du texte, que nous obtenons nos meilleurs résultats. Dans cette configuration, l'IOU moyen obtenu sur l'ensemble des cartes est de 52%. Par ailleurs, 7 cartes (sur les 29 de notre jeu de données) obtiennent un IOU supérieur à 80%. l'IOE moyen obtenu pour ce lancement est de 79,48%. Il est à noter que ces scores pourraient paraître faibles, mais nous insistons ici sur le fait que notre jeu de données est hétérogène, et contient des cartes qui sont des défis pour ce type de détection. En particulier, plusieurs cartes ne sont pas conformes aux hypothèses que nous avons faites (tel que l'alignement à gauche).
Nous nous sommes ici concentrés sur un lancement particulier (F-A-D-H-T), cependant, d'autres ordres dans les critères sont également intéressants, et certains *patterns* semblent pouvoir être dessinés. La liste exhaustive des résultats obtenus pendant notre campagne d'expérimentation est disponible à l'adresse `https://www.cril.univ-artois.fr/~marzinkowski/`

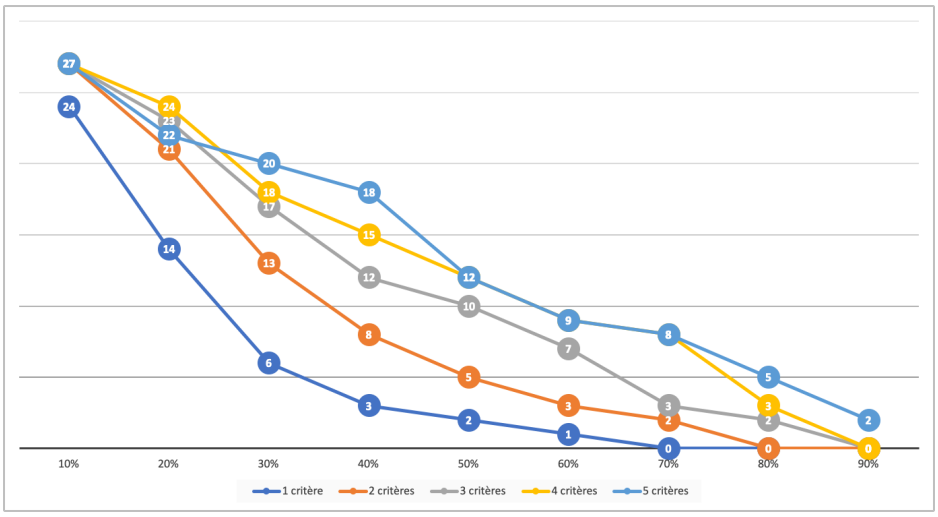

FIGURE 6 – Graphique du pourcentage de carte moyen d'IOU supérieur à l'axes des abscisses

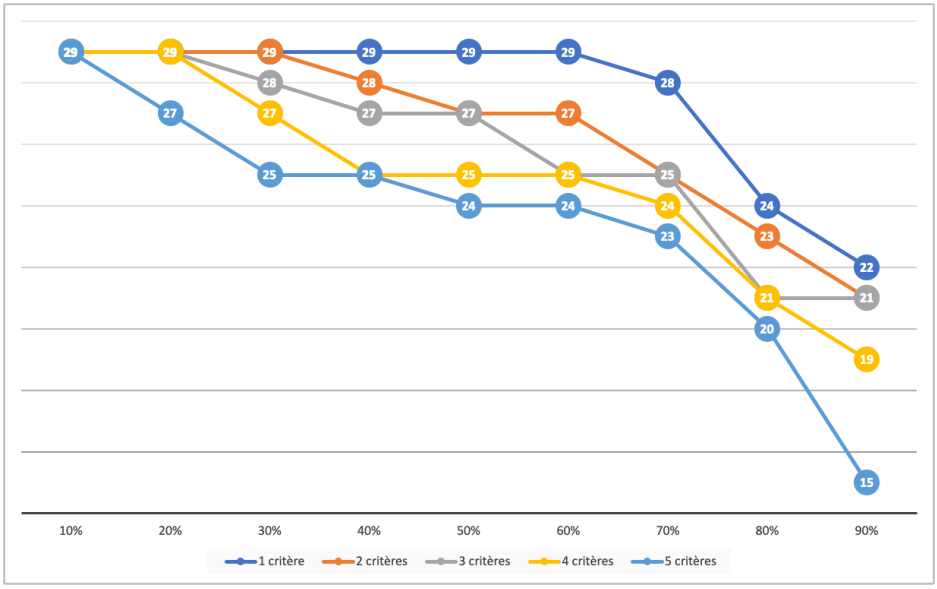

FIGURE 7 – Graphique du pourcentage de carte moyen d'IOE supérieur à l'axes des abscisses

`incremental-clustering-results/`

5 Conclusion et perspectives

Dans cet article, nous avons vu comment les techniques de regroupement, couplées à des critères appropriés, permettent de détecter automatiquement les textes de légende dans des cartes en tout genre : plans de ville, réseaux d'eau, cartes routières, etc.

Pour regrouper efficacement les zones de texte susceptibles de représenter une légende, nous avons établi cinq critères, chacun associé à une mesure de distance spécifique. Par la suite, nous avons introduit un algorithme incrémental, paramétré par un vecteur de critères, nous permettant d'améliorer le partitionnement en exploitant l'entropie comme mesure de l'homogénéité des partitions obtenues à chaque étape. L'étude expérimentale, menée sur un échantillon de cartes très représentatif et utilisant un ou plusieurs critères, a montré que les raffinements successifs des partitions donnent de meilleurs résultats que l'utilisation d'un seul critère.

Il existe plusieurs pistes de recherches futures pour ce travail. Premièrement, nous visons à généraliser les distances proposées pour chaque critère afin de s'adapter aux différentes formes que peuvent prendre les textes de légende. Un autre travail consiste à proposer une fonction d'agrégation globale des différentes distances associées aux cinq critères ; et ainsi appliquer l'algorithme de regroupement une seule fois.

Une autre orientation future consiste à explorer des algorithmes de regroupement alternatifs, avec un accent particulier sur le *clustering* hiérarchique. En parallèle de ces travaux, nous envisageons d'étudier des algorithmes de clustering visant à identifier les objets alignés et associés aux textes de légende. Enfin, nous collectons actuellement un

grand nombre de cartes de légende pour mener une étude expérimentale plus approfondie.

Remerciements

Ce travail a reçu le soutien du programme de recherche Horizon Europe Marie Skłodowska-Curie Actions MSCA (Staff Exchanges) grant agreement 101086252 ; Call : HORIZON-MSCA-2021-SE-01, Projet : STARWARS (STormwAteR and WastewAteR networkS heterogeneous data AI-driven management).
Il a également reçu le soutien de l'Agence Nationale de la Recherche, via le projet ANR CROQUIS (Collecte, représentation, complétion, fusion et interrogation de données de réseaux d'eau urbains hétérogènes et incertaines), grant ANR-21-CE23-0004.

Références

[1] Anurag Agrahari and Rajib Ghosh. Multi-oriented text detection in natural scene images based on the intersection of mser with the locally binarized image. *Procedia Computer Science*, 171 :322–330, 2020.

[2] Yao-Yi Chiang and Craig A Knoblock. An approach for recognizing text labels in raster maps. In *20th International Conference on Pattern Recognition*, pages 3199–3202, 2010.

[3] Youcef Djenouri, Asma Belhadi, Philippe Fournier-Viger, and Jerry Chun-Wei Lin. Fast and effective cluster-based information retrieval using frequent closed itemsets. *Information Sciences*, 453 :154–167, 2018.

[4] Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[5] S. Karthikeyan, Vignesh Jagadeesh, and B. S. Manjunath. Learning bottom-up text attention maps for text detection using stroke width transform. In *2013 IEEE International Conference on Image Processing*, pages 3312–3316, 2013.

[6] D. Kavitha and V. Radha. Text detection based on text shape feature analysis with intelligent grouping in natural scene images. In Somnath Bhattacharyya, Jitendra Kumar, and Koeli Ghoshal, editors, *Mathematical Modeling and Computational Tools*, pages 467–479, Singapore, 2020. Springer Singapore.

[7] Rahim Khan, Yurong Qian, and Sajid Naeem. Extractive based text summarization using kmeans and tf-idf. *International Journal of Information Engineering and Electronic Business*, 11 :33–44, 05 2019.

[8] Rutuja Kumbhar, Snehal Mhamane, Harshada Patil, Sukruta Patil, and Shubhangi Kale. Text document clustering using k-means algorithm with dimension reduction techniques. In *5th International Conference on Communication and Electronics Systems (ICCES)*, pages 1222–1228, 2020.

[9] James Lintern. Recognizing text in google street view images. 2010.

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

[11] Clemens Neudecker, Konstantin Baierer, Mike Gerber, Christian Clausner, Apostolos Antonacopoulos, and Stefan Pletschacher. A survey of ocr evaluation tools and metrics. HIP '21. Association for Computing Machinery, 2021.

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

[13] 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*, 12 :2825–2830, 2011.

[14] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio 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*, pages 658–666. Computer Vision Foundation / IEEE, 2019.

[15] Markus Stricker and Markus Orengo. Storage and retrieval for image and video databases (spie) - similarity of color images. *SPIE Proceedings*, 2420 :381–392, March 1995.

[16] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. Constrained k-means clustering with background knowledge. In *ICML*, pages 577–584, 2001.

[17] Chucai Yi and YingLi Tian. Text string detection from natural scenes by structure-based partition and grouping. volume 20(9), pages 2594—-2605, 2011.

[18] Hui Yin, Amir Aryani, Stephen Petrie, Aishwarya Nambissan, Aland Astudillo, and Shengyuan Cao. A rapid review of clustering algorithms, 2024.