Audit d'une base de documents étiquetée

Romain Giot, Romain Bourqui, Nicholas Journet, Anne Vialard

Audit d'une base de documents étiquetée[1]

Romain Giot — Romain Bourqui — Nicholas Journet — Anne Vialard

Univ. Bordeaux, Laboratoire Bordelais de Recherche en Informatique, UMR 5800, F-33405 Talence, France

RÉSUMÉ. *Dans cet article, déjà présenté à ICDAR 2015, nous nous intéressons à l'étiquetage d'une base d'images de documents dans un contexte industriel. Nous travaillons plus particulièrement sur l'évaluation de la qualité d'un étiquetage préexistant. Dans la plupart des cas pratiques, un opérateur étiquette manuellement une base d'images de documents en parcourant séquentiellement les vignettes correspondant aux images. Cette tâche est très répétitive ; de plus le plan de classement définissant les noms et le nombre des classes est souvent incomplet, ce qui conduit à de nombreuses erreurs d'étiquetage. La question est alors d'évaluer si la qualité d'un lot étiqueté est suffisante pour qu'il soit accepté. Notre objectif est de faciliter et d'accélérer cette évaluation qui prend en pratique plus de* $1,5$ *fois le temps nécessaire à l'étiquetage lui-même. Nous proposons un outil interactif pour visualiser les données sous la forme d'un graphe. Ce graphe met en évidence les similarités entre documents ainsi que la qualité de l'étiquetage. Nous définissons des critères sur le graphe qui caractérisent les trois types d'erreur qu'un opérateur peut faire : une image est mal étiquetée, une classe devrait être découpée en sous-classes plus pertinentes, plusieurs classes devraient être fusionnées en une seule. Ceci nous permet de focaliser l'attention de l'utilisateur sur de potentielles erreurs. Il peut alors plus facilement compter les erreurs d'étiquetage et valider (ou pas) une qualité d'étiquetage conforme aux attentes.*

ABSTRACT. *The context of this paper, already presented at ICDAR 2015, is the labelling of a document image database in an industrial process. Our work focuses on the quality assessment of a given labelled database. In most practical cases, a database is manually labelled by an operator who has to browse sequentially the images (presented as thumbnails) until the whole database is labelled. This task is very repetitive; moreover the filing plan defining the names and number of classes is often incomplete, which leads to many labelling errors. The question is then to certify if the quality of a labelled batch is good enough to globally accept it. Our objective is to ease and speed up that evaluation that needs up to* 1.5 *more times than the labelling work itself. We propose an interactive tool for visualizing the data as a graph. That graph enhances similarities between documents as well as the labelling quality. We define criteria on the graph*

1. Traduction d'un article publié en anglais à ICDAR 2015 (Giot *et al.*, 2015)

that characterize the three types of errors an operator can do: an image is mislabelled, one class should be split in more pertinent subclasses, several classes should be merged in another. This allows us to focus the operator attention on potential errors. He can then count the errors encountered while auditing the database and assess (or not) the global labelling quality.

MOTS-CLÉS : *Multimedia analytics, Visualisation, Base d'images de documents, Qualité*

KEYWORDS: *Multimedia analytics, Visualization, Document images database, Quality*

1. Introduction

Le but de ce travail est d'aider les sociétés qui fournissent des solutions de numérisation de documents. Une de leurs tâches est d'étiqueter d'immenses quantités d'images de documents (cartes d'identité, formulaires, tickets, reçus, factures...). Un processus de numérisation standard consiste à scanner les documents physiques et à étiqueter manuellement chaque image suivant un plan de classement. Selon notre partenaire industriel[1], un opérateur humain peut étiqueter manuellement 300 à 500 documents par heure. Une commande standard représente en moyenne 5 à 6 semaines d'étiquetage manuel pour un opérateur. Cette tâche présente trois difficultés principales. Premièrement, ce travail est très répétitif et demande un haut niveau de concentration pour éviter les erreurs d'étiquetage. Deuxièmement, certaines images sont difficiles à étiqueter ; deux images qui appartiennent à deux classes différentes peuvent être très semblables visuellement et présenter seulement des différences minimes (un nom différent dans un formulaire, la taille d'un ticket, ...). La dernière difficulté est liée au plan de classement destiné à l'opérateur et qui contient le nombre et le nom des classes. Comme une campagne d'étiquetage dure plusieurs semaines, le plan de classement peut changer (règles de classification, création ou suppression de classes). Toutes ces difficultés conduisent à des erreurs d'étiquetage que l'on peut séparer en trois types. Le premier type d'erreur correspond à une « erreur de classification » : une image de document est associée à la classe A alors qu'elle devrait être dans la classe B. Les deux autres types d'erreur sont liés aux changements du plan de classement. L'« erreur de fusion » correspond à l'ajout de nouvelles classes au plan de classement (des images qui étaient dans la classe A doivent être classifiées dans les classes A_1, A_2 et A_3). L'« erreur de découpage » se produit lorsque plusieurs classes sont supprimées du plan de classement (les classes A_1, A_2 et A_3 doivent être fusionnées en une unique classe A). Comme le plan de classement peut changer plusieurs fois lors d'une campagne d'étiquetage, les erreurs de fusion et de découpage sont fréquentes.

Les sociétés de numérisation garantissent à leurs clients un taux maximum d'erreurs d'étiquetage. En pratique, l'évaluation de la qualité de l'étiquetage d'un lot d'images de documents se fait par un audit manuel. Généralement, durant les deux ou trois premières semaines de la campagne d'étiquetage, 100% de la partie de la base déjà étiquetée est auditée. Dans les semaines suivantes le pourcentage des images auditées baisse à 10% de la base. Des statistiques menées sur de nombreuses productions montrent que l'audit de 100% d'une base nécessite $1,5$ fois plus de temps que le processus d'étiquetage lui-même. Ce processus d'audit est actuellement réalisé avec une interface homme-machine très basique : les images d'une même classe sont présentées séquentiellement à l'opérateur qui signale les erreurs d'étiquetage. Nous avons précédemment testé ce type d'interface visuelle pour une tâche de classification d'images de documents. Les expérimentations que nous avons menées dans (Augereau *et al.*, 2011) mettent clairement en évidence qu'il est très difficile de traiter de cette façon une grande

1. Nous remercions www.gestform.com pour avoir soutenu ce travail

quantité de documents et de classes. C'est pourquoi nous proposons d'utiliser des techniques de visualisation d'information pour aider l'opérateur.

La visualisation d'information exploite les capacités visuelles humaines pour aider à l'exploration et à l'analyse visuelle d'un ensemble d'informations (Ware, 2000). Elle traite également le problème posé par la profusion d'informations (Thomas et Cook, 2006). Les recommandations de Schneiderman (Schneiderman, 1996) pour l'exploration visuelle de données sont maintenant connues sous de nom de « Mantra de la recherche visuelle d'informations » : « d'abord une vue d'ensemble, zoomer et filtrer, puis les détails à la demande ». Fournir une vue d'ensemble permet à l'utilisateur d'identifier les tendances principales des données et donc de guider son exploration et de focaliser son attention sur les parties intéressantes. Zoomer et filtrer sont des techniques basiques d'interaction en visualisation d'information qui permettent de réduire la quantité d'éléments affichés et donc de réduire la charge cognitive de l'opérateur. Enfin les détails à la demande correspondent à toutes les techniques qui fournissent des informations détaillées sur quelques éléments quand et seulement quand l'opérateur le demande.

Dans cet article, nous proposons d'utiliser Tulip, un framework pour la visualisation d'informations basé graphe (Auber *et al.*, 2010), pour créer un outil visuel interactif. L'originalité de ce travail tient dans la représentation visuelle et multi-échelle d'une base d'images de documents : le premier niveau représente les similarités entre classes, tandis que le second niveau représente les similarités entre images de documents.

En ajoutant une logique métier au processus d'audit nous créons un outil qui permet de parcourir les données efficacement pour détecter rapidement les erreurs d'étiquetage. L'opérateur aura un retour visuel indiquant les similarités entre documents et entre classes, les distances entre classes et à l'intérieur d'une classe. De plus, en focalisant son attention sur les parties du graphe correspondant aux trois types d'erreur (erreur de classification, erreur de découpage, erreur de fusion), le processus d'audit sera facilité et accéléré.

La section 2 présente l'état de l'art sur l'exploration de bases d'images. La section 3 détaille comment utiliser les algorithmes de visualisation de Tulip pour créer une interface visuelle dédiée au processus d'audit. La section 4 détaille 3 mesures créées pour identifier dans un graphe de documents les trois principales erreurs qui peuvent être faites durant une campagne d'étiquetage. La section 5 présente des éléments d'évaluation de notre proposition et la section 6 conclue l'article.

2. État de l'art

Notre problématique est un cas particulier d'exploration de bases d'images. Autant que nous le sachions, il n'y a pas eu de publication spécifique sur la visualisation et l'exploration d'une base d'images de documents. Les propositions les plus proches sont dédiées à la recherche d'images basée sur le contenu.

Les auteurs de (Plant et Schaefer, 2011) listent trois classes principales de méthodes de visualisation d'une base d'images. La première classe regroupe les méthodes basées *mapping* : des similarités entre images sont calculées dans un espace de caractéristiques de dimension élevée et sont préservées autant que possible dans une projection 2D où les images sont présentées à l'utilisateur. La deuxième classe contient les méthodes de visualisation basées *clustering*. Créer des groupes d'images proches devient en effet nécessaire lorsque la taille de la base d'images augmente. Les images sont regroupées en fonction de caractéristiques image ou de méta-données qui leur sont associées et finalement une seule image représentative de chaque groupe est présentée à l'utilisateur. La dernière classe correspond aux méthodes de visualisation basées graphe. En général, les sommets du graphe sont les images et des arêtes entre images traduisent leur similarité. Pour générer la visualisation finale on trouve de nombreuses variantes d'algorithmes modélisant le graphe par un système masses-ressorts (Fruchterman et Reingold, 1991).

Des outils standard de navigation s'appuyant sur une de ces méthodes de visualisation permettent ensuite d'explorer la base d'image : déplacement, changement d'échelle, exploration verticale dans les visualisations hiérarchiques, etc. On peut aussi définir un type de navigation spécifique à une application. Par exemple, dans les applications de recherche d'image, la navigation peut être guidée par des critères de pertinence.

Enfin on peut citer PEx-Image (Eler *et al.*, 2009) comme exemple d'outil générique pour la visualisation et l'analyse d'une base d'images. Il intègre des fonctionnalités complémentaires : calcul de nombreuses caractéristiques image, sélection de caractéristiques, plusieurs projections 2D sont disponibles dont la projection basée distance et les arbres de similarité. Plusieurs vues sur les mêmes données peuvent être synchronisées. De nombreux cas d'utilisation sont décrits dans cet article : comparaison de la pertinence de deux ensembles de caractéristiques pour une base d'images étiquetées, classification guidée par les similarités entre images, intégration d'une information textuelle associée à chaque image. En termes de taille des données, PEx-Image peut traiter jusqu'à 9 000 images.

Les auteurs de (Worring *et al.*, 2012) présentent un outil combinant l'analyse multimédia et des techniques de visualisation avancée pour faciliter la recherche d'images dans le domaine de la police scientifique. Nous pouvons situer notre travail dans le même domaine de recherche appelé *Multimedia Analytics*. Nous générons une visualisation d'une base d'images de documents qui combine un clustering basé sur des méta-données (les étiquettes des images) et une représentation par graphes. L'outil proposé permet à l'utilisateur de détecter efficacement les erreurs d'étiquetage de la base.

3. Construction du graphe et outils de visualisation

Notre logiciel permet d'afficher une base d'images de documents sous la forme d'un graphe de similarité où la couleur, et la position des nœuds et arêtes sont vecteurs

d'informations sur les similarités entre images (ou groupes d'images). Cette section présente en détail la manière dont cette visualisation sous forme de graphe est rendue possible.

3.1. *Extraction de caractéristiques*

Pour construire le graphe, il faut pouvoir comparer les images entre elles. Pour cela, nous calculons deux catégories de caractéristiques. Tout d'abord, à l'aide d'un logiciel d'OCR, nous calculons un histogramme composé des 500 mots les plus fréquemment rencontrés de la base. Deux algorithmes de traitement du langage (lemmatisation et stop-words) permettent de construire un histogramme composé uniquement des 500 mots les plus pertinents. Dans un second temps, nous extrayons des caractéristiques sur l'image elle-même. Nous calculons celles présentées dans (Augereau *et al.*, 2011) : une image est divisée en 12 zones de surfaces équivalentes et pour chacune d'entre elles est calculée la moyenne des niveaux de gris. La hauteur et la largeur des images sont également utilisées comme caractéristique descriptives. Ainsi, chaque document est décrit par un vecteur de 500 caractéristiques de type « texte » et 14 de type « image ».

3.2. *Construction du graphe*

Soit $G^f = (V, E^f)$ le graphe représentant une base d'images sur lesquelles les caractéristiques f ont été extraites. Pour des raisons de concision, nous omettrons ^{f} dans l'ensemble des formules suivantes, mais il faut retenir que la topologie du graphe dépend non seulement des documents visualisés mais également des caractéristiques utilisées.

À chaque élément de l'ensemble des nœuds $V = (v_i)_{i=1...n_v}$ correspond une image de document. $E = (e_i)_{i=1...n_e}$ est l'ensemble des arêtes orientées du graphe. Il y a une arête $e_i = (v_m, v_n)$ si le nœud (document) v_n fait partie des k plus proches voisins du nœud v_m (selon les caractéristiques f). Soit $\mathcal{L} = (l_i)_{i=1...n_l}$ l'ensemble des classes étiquetées. L'application $C : V \rightarrow \mathcal{L}$ correspond à l'étiquetage de chaque document. Un méta-graphe $G_M(G, C) = (V_M, E_M)$ est construit à partir du graphe G. Chaque nœud v_{M_i} du méta-graphe correspond à une classe de documents : $v_{M_i} = \{v_j / C(v_j) = l_i\}$. Il y a une arête orientée (méta-arête) entre deux méta-nœuds (classes) si au moins un nœud (image de document) de la classe source a un de ses k plus proches voisins dans le méta-nœud (classe) de destination. $E_M = \{(v_{M_i}, v_{M_j}) / \exists (v_n, v_m) \in E, C(v_n) = l_i, C(v_m) = l_j\}$.

3.3. *Mesures de qualité du graphe*

Nous présentons ici comment nous calculons différentes mesures de qualité associées à la construction du graphe. Ces mesures permettent de quantifier la qualité de

partitionnement du graphe et donnent donc des indications sur la qualité de la classification manuelle. Dans le cas idéal, les méta-nœuds n'ont pas d'arêtes et le sous-graphe qui représente chaque classe est connexe avec k arêtes sortantes par nœud. Sur la base de travaux de (Mancoridis *et al.*, 1998), nous proposons d'évaluer la cohésion interne de chaque méta-nœud (classe) en comparant le nombre d'arêtes sortant d'un méta-nœud par rapport au nombre maximum théorique d'arêtes pouvant en sortir. Soit $E(v_{M_i}, v_{M_j}) \subset E$ l'ensemble des arêtes reliant un nœud du méta-nœud l_i à un nœud du méta-nœud l_j. Soit $V(v_{M_i}) \subset V$ l'ensemble des nœuds du méta-nœud l_i (*c.-à-d.* : les nœuds de v_{M_i}).

La cohésion interne d'un méta-nœud permet d'indiquer à quel point les images étiquetées de la même manière sont liées, au sens des caractéristiques f. Elle est définie par :

$$IC(v_{M_i}) = \frac{|E(v_{M_i}, v_{M_i})|}{|V(v_{M_i})| * min(k, |V(v_{M_i})| - 1)}$$

Notons que la fonction `min` permet de gérer le cas de figure où il y a moins de k nœuds (images) dans un méta-nœud (classe).

De la même manière, il est possible d'évaluer la cohésion externe de la classification. Cette cohésion externe permet de mesurer à quel point des éléments étiquetés dans deux classes différentes sont similaires au sens des caractéristiques f.

$$EC(e_{M_{ij}}) = \frac{|E(v_{M_i}, v_{M_j})|}{|V(v_{M_i})| * min(k, |V(v_{M_j})|)} \text{ avec } e_{M_{ij}} = (v_{M_i}, v_{M_j}).$$

Idéalement, tous les IC valent 1 et tous les EC valent 0. Ceci signifie qu'idéalement les k plus proches voisins d'un document sont localisés dans la même classe (méta-nœud) que cet élément. En d'autres termes, cela signifie que la classification manuelle est en adéquation avec les caractéristiques f calculées.

À partir de ces mesures de qualité locales, nous sommes en mesure de calculer une valeur de qualité globale (plus cette valeur est forte, meilleure est la qualité globale).

$$Q = \frac{\sum_{v_{M_i}} IC(v_{M_i})}{|\mathcal{L}|} - \frac{\sum_{e_{M_{ij}}} EC(e_{M_{ij}})}{\sum_{v_{M_i}} min\big(|\mathcal{L}| - 1, |V(v_{M_i})| * k\big)}$$

3.4. *Outils interactifs d'exploration de la base d'images*

Cette section décrit les outils que nous avons développés et qui permettent d'interagir avec le graphe. Ces outils visent à faciliter l'exploration du graphe et surtout à faciliter l'identification d'erreurs de classification (la figure 1 présente une capture d'écran de ce logiciel).

Dans le graphe, chaque nœud est représenté par une vignette de l'image du document et chaque arête est représentée par un trait. Toute la difficulté de l'étape de construction du graphe est de réaliser une représentation qui soit visuellement pertinente. La première étape consiste donc à assigner des coordonnées aux nœuds et arêtes. Les approches les plus populaires pour déterminer ces coordonnées sont les

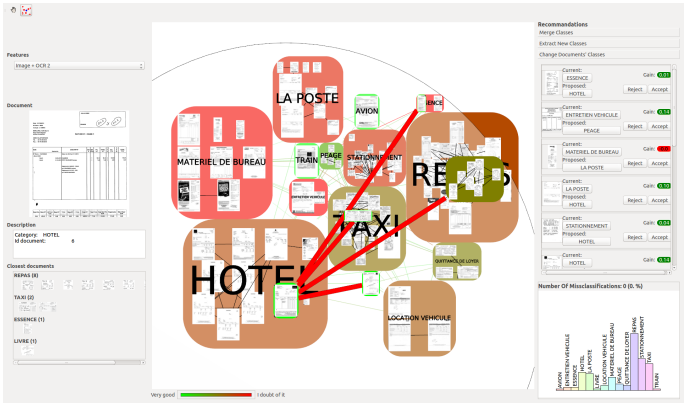

Figure 1. *Capture d'écran de notre logiciel. Le paneau du milieu permet d'afficher la base d'images sous la forme d'un graphe. Chaque document correspond à un nœud. Les méta-nœuds contiennent les nœuds d'étiquettes identiques (étiquetage manuel). Les arêtes en noir relient les nœuds similaires (les k plus proches de chaque document au sens des caractéristiques). Les méta-arêtes relient les méta-nœuds ayant des nœuds en commun. La couleur des méta-nœuds et des méta-arêtes représentent respectivement les valeurs de cohésion interne et de cohésion externe. En passant la souris sur une image de document, s'affiche en sur-imposition le méta-graphe correspondant aux plus proches voisins du document.*

algorithmes de dessin basé sur les forces. Ils sont réputés pour fournir un résultat visuellement agréable et structurellement pertinent. Ces algorithmes se basent sur une analogie physique dans laquelle un nœud est un objet physique et chaque arête est un ressort. Il en résulte un positionnement où les nœuds proches (en termes de distance de graphe) sont positionnés près les uns des autres. Nous avons utilisé l'algorithme FM^3 (Hachul et Jünger, 2005) qui offre l'avantage de fournir un bon ratio entre le résultat visuel généré et le temps de calcul nécessaire.

Un bon algorithme de visualisation doit aussi éviter le recouvrement des nœuds entre eux. Même si FM^3 prend en compte la taille des nœuds dans son calcul de positionnement, il ne garantit pas l'absence de recouvrement. Pour cette raison, nous appliquons l'algorithme *Fast Overlap Removal* (FOR) (Dwyer *et al.*, 2005) en post-traitement à FM^3. Il permet de supprimer les recouvrements de nœuds tout en minimisant le nombre de déplacements.

Dans notre cas, nous avons besoin de gérer à la fois l'affichage des nœuds et l'affichage des méta-nœuds. Pour permettre un affichage de qualité, nous utilisons une approche ascendante. Le sous-graphe qui représente un méta-nœud est dessiné de manière indépendante en utilisant successivement les algorithmes FM^3 et FOR. La taille de la boite englobante d'un sous-graphe est utilisé comme taille de son méta-nœud

représentant. Ceci permet finalement de placer les méta-nœuds les uns par rapport aux autres de manière optimale.

Afin d'afficher des informations facilitant l'exploration de la base et la recherche d'erreurs, nous affichons également les cohésions internes et externes en utilisant un panel de couleurs précis. Comme ces deux mesures sont bornées entre 0 et 1, nous avons mis en place un mapping linéaire de couleurs allant du rouge au vert. Le vert est associé à la notion de « bonne cohésion », le rouge indique au contraire une « mauvaise cohésion ». Pour les méta-nœuds le même code couleur est appliqué. Il indique ainsi quelles sont les classes contenants potentiellement des erreurs de classification. Dans le cas d'une forte cohésion externe, les couleurs permettent d'indiquer les classes potentiellement intéressantes à fusionner. Enfin, toujours pour produire un indice visuel intéressant, la taille des arêtes de G_M sont également liées à la cohésion externe.

Comme cela a été mentionné précédemment, il est possible de générer le graphe selon diverses caractéristiques. L'interface permet à l'utilisateur de définir lui-même s'il souhaite visualiser la base sous la forme d'un graphe dont l'affichage résulte uniquement des caractéristiques textuelles ou uniquement des caractéristiques de type image. L'utilisateur peut aussi décider d'utiliser toutes les caractéristiques simultanément.

Par rapport à une approche classique consistant à visualiser les images de documents les unes à côté des autres (à la manière de Google image), nous proposons un ensemble d'outils améliorant la phase d'exploration de la base. Cependant, pour faciliter encore cette tâche fastidieuse qu'est l'audit d'une base, il est possible d'aider encore plus l'utilisateur. C'est pour cela que nous proposons dans la suite de cet article trois nouvelles mesures calculées sur le graphe. Ces mesures permettent d'aider l'utilisateur à identifier les erreurs d'étiquetage, les classes devant être fusionnées ou encore celles qui doivent être divisées en plusieurs sous classes.

4. Trois mesures pour identifier de potentielles erreurs d'étiquetage et améliorer l'expérience utilisateur

Dans la section 1, nous avons listé trois types d'erreurs récurrentes qui se produisent lors de l'étiquetage d'une base d'images de documents. Pour corriger une erreur, il faut appliquer l'opération inverse (*c.-à-d.* une fusion pour une erreur de découpage ou un découpage pour une erreur de fusion). L'interface utilisateur intègre un module de suggestion d'erreur où sont listées les erreurs potentielles et la correction proposée. C'est à l'opérateur d'accepter ou de rejeter chaque proposition. Si l'étiquetage manuel est effectivement incorrect, l'opérateur peut valider la proposition et un compteur d'erreur est incrémenté. Nous ne modifions toutefois pas la topologie du graphe à chaque validation d'erreur pour éviter que l'utilisateur ne perde ses points de repère visuels. En s'appuyant sur le total des erreurs détectées, l'opérateur peut à tout moment accepter ou rejeter le lot traité.

4.1. *Erreurs d'étiquetage*

Pour chaque classe (méta-nœud), nous proposons une liste de documents (nœuds) qui pourraient être déplacés dans une autre classe. L'idée principale est d'identifier les nœuds qui sont fortement connectés à un ensemble de nœuds appartenant à une autre classe. Nous considérons tous les déplacements possibles de documents de la classe d'intérêt vers les classes auxquelles elle est reliée et nous gardons la modification qui conduit à la meilleure valeur de qualité globale. La complexité dépend du nombre d'arêtes interclasses.

L'opérateur peut valider l'erreur et sa proposition de correction. S'il la valide, le compteur d'erreur est incrémenté. Puis le deuxième nœud le plus isolé de la classe l_n est traité de la même façon et ainsi de suite jusqu'à ce que l'opérateur considère qu'il n'y a plus d'erreur d'étiquetage dans la classe l_n. Le nombre d'erreurs comptabilisé est le nombre total d'erreurs validées par l'opérateur.

4.2. *Erreurs de fusion*

Dans le cas d'une erreur de fusion, la correction doit proposer d'extraire des nœuds d'un méta-nœud pour créer de nouvelles classes (*c.-à.d.* : découpe d'un méta-nœud).

Soit SG_n le sous-graphe de G induit par les sommets de classe l_n. Nous calculons une partition de ce sous-graphe en utilisant l'algorithme MCL (Markov Clustering) (Enright A.J., 2002) qui extrait des clusters compacts dans un graphe. Soit MCL_n la liste des clusters de nœuds calculés par cet algorithme sur le graphe SG_n. Chaque cluster peut constituer une nouvelle classe. Les clusters constitués d'un nœud unique, considérés comme du bruit, sont exclus. Etant donné que MCL découpe souvent un graphe en de trop nombreux clusters de tailles différentes, nous proposons de garder seulement les trois plus grands. Nous estimons que proposer au maximum trois découpages est suffisant pour confirmer (ou pas) l'erreur de fusion détectée. L'opérateur peut alors prendre la décision de créer trois nouvelles classes à partir du résultat du clustering. S'il accepte la suggestion de découpage d'une classe, le nombre d'erreurs associé est incrémenté du nombre de nœuds retirés de la classe de départ.

4.3. *Erreur de découpage*

La correction d'une erreur de découpage consiste à fusionner deux classes existantes en une nouvelle classe. S'il y a un nombre important d'arêtes de G reliant des nœuds d'une classe vers une autre, il est probable que les deux classes devraient être fusionnées. Pour chaque méta-nœud, nous proposons donc un autre méta-nœud avec lequel il pourrait fusionner. Étant donné un méta-nœud source v_{M_m}, le nœud destination est trouvé comme suit : $\arg\max_n |\{(v_i, v_j) \in E / C(v_i) = l_m, C(v_j) = l_n, m \neq n\}|$. En pratique, si la fusion proposée est pertinente, l'opérateur la valide. Le nombre d'erreurs comptabilisé est incrémenté par le nombre de nœuds de v_{M_m}.

5. Évaluation

5.1. *Résultats visuels*

Comme indiqué par Worring (Worring *et al.*, 2012) à propos de « multimédia analytics », il est difficile d'évaluer ce type d'application car « il y a énormément de facteurs qui influencent le résultat ». Les tests que nous avons réalisés sur des documents provenant de campagnes de numérisation industrielles montrent que la taille de la base de données n'est pas le seul problème à résoudre pour fournir une bonne visualisation. Les problèmes classiques de visualisation d'une base d'images sont en partie gérés dans notre logiciel. La visualisation est correcte jusqu'à 300 documents par classe (pas de recouvrement d'image, affichage rapide). Cependant des facteurs spécifiques peuvent limiter les actions de l'opérateur durant le processus d'audit. De façon évidente, l'efficacité de l'opérateur dépend de sa connaissance du contenu de la base et de la taille de chaque classe de document. Notre proposition permet de surmonter ce dernier problème en identifiant les erreurs de classification, de fusion et de découpage. Comptabiliser une erreur de classification est facilité par le fait de présenter à l'opérateur simultanément une unique image de document et son étiquette la plus probable (cf figure 1, partie droite). De la même façon, on peut identifier les erreurs de découpage sans parcourir séquentiellement chaque méta-nœud. En cliquant alternativement sur les deux étiquettes proposées, l'opérateur peut visualiser facilement les deux classes et décider de les fusionner ou pas. Enfin les mesures destinées à identifier les erreurs de fusion aident réellement à évaluer si un méta-nœud doit être découpé en trois nouveaux méta-nœuds. La figure 2.a-b montre le mouvement de caméra effectué lorsqu'on clique successivement sur les boutons « 1/2/3 » à côté de l'étiquette de la classe (zoom puis vue panoramique). Ceci permet de comparer très simplement chaque sous-graphe extrait par l'algorithme MCL.

5.2. *Performance des suggestions*

Les résultats des tests réalisés sur deux bases de données réelles (plus de 100 images pour la première, plus de $3\,000$ documents et 14 classes pour la seconde) sont présentés dans le Tableau 1.

Le paramètre k a été choisi de façon empirique. L'objectif est de vérifier si le système permet à l'utilisateur de ne pas parcourir l'ensemble des images de document pour trouver les erreurs d'étiquetage. Pour chaque base, nous avons généré 100 graphes contenant une seule erreur d'étiquetage. Pour différentes valeurs de k et différentes caractéristiques, nous avons compté combien de fois l'opérateur doit rejeter la proposition avant d'identifier la vraie erreur d'étiquetage, proportionnellement au cardinal de la classe du document erroné. Globalement, les tests montrent que pour la petite ou pour la grande base, 20% à 30% des documents d'une classe doivent être parcourus avant l'identification de l'erreur d'étiquetage (au lieu de 100% sans notre outil). Les grandes valeurs d'écart type montrent que très souvent, la première proposition est la bonne (dans 50% des cas), mais parfois plus de 45% des

Tableau 1. *Pourcentage (moyenne/écart type) de fausses suggestions avant une proposition correcte de ré-étiquetage. Caractéristiques utilisées :* f_1*=ocr+image,* f_2*=ocr,* f_3*=image*

		$k=10$			$k=20$		
		f_1	f_2	f_3	f_1	f_2	f_3
DB 100	μ	0,24	0,20	0,28	0,22	0,23	0,23
	σ	0,87	0,24	0,94	0,84	0,42	0,94
DB 3000	μ	0,30	0,24	0,34	0,28	0,26	0,47
	σ	0,23	0,17	0,27	0,28	0,27	0,28

Tableau 2. *Temps de calcul total (en secondes) pour trois bases d'images de documents.*

$\|V\|$	$\|\mathcal{L}\|$	Affichage	Qualité	Sugg. de découpage	Sugg. de réétiquetage	Sugg. de fusion
115	14	$0,032$	$0,006$	$0,006$	$0,021$	$0,001$
3 224	14	$1,927$	$0,014$	$0,275$	$1,343$	$0,009$
30 394	210	$8,463$	$0,047$	$11,954$	$75,937$	$2,634$

documents de la classe doivent être parcourus. Une démonstration est disponible sur `http://njournet.com/files/DocClass.mp4`. Le Tableau 2 présente les performances en temps de calcul des méthodes implémentées sur trois bases différentes avec $k = 10$ (application en C++, Intel® Core™ i7-3840QM CPU@2.80GHzx8, 32Gb of RAM).

6. Conclusion et perspectives

Cet article constitue une nouvelle proposition pour parcourir efficacement de grandes bases d'images de documents afin de détecter des erreurs de classification. Nous focalisons l'attention de l'utilisateur sur des erreurs potentielles en nous appuyant sur des mesures de qualité basées graphe. Nous avons validé les performances en temps de calcul de cette proposition avec une base composée de 30 000 images de documents étiquetées à la main. Nous avons démontré la pertinence de notre approche pour la modélisation et la visualisation des données. Il reste toutefois des amélioration à apporter à notre méthode. Prioritairement, nous intégrerons de nouvelles caractéristiques image et texte plus fines et plus nombreuses. Les caractéristiques utilisées pour un corpus donné pourraient être sélectionnées automatiquement à partir de la classification manuelle initiale.

Nous évaluerons également s'il serait pertinent d'utiliser une méthode d'apprentissage de fonction de distance ou une distance non linéaire au lieu d'utiliser un simple algorithme k-ppv. Le logiciel devrait aussi permettre la correction des erreurs d'étique-

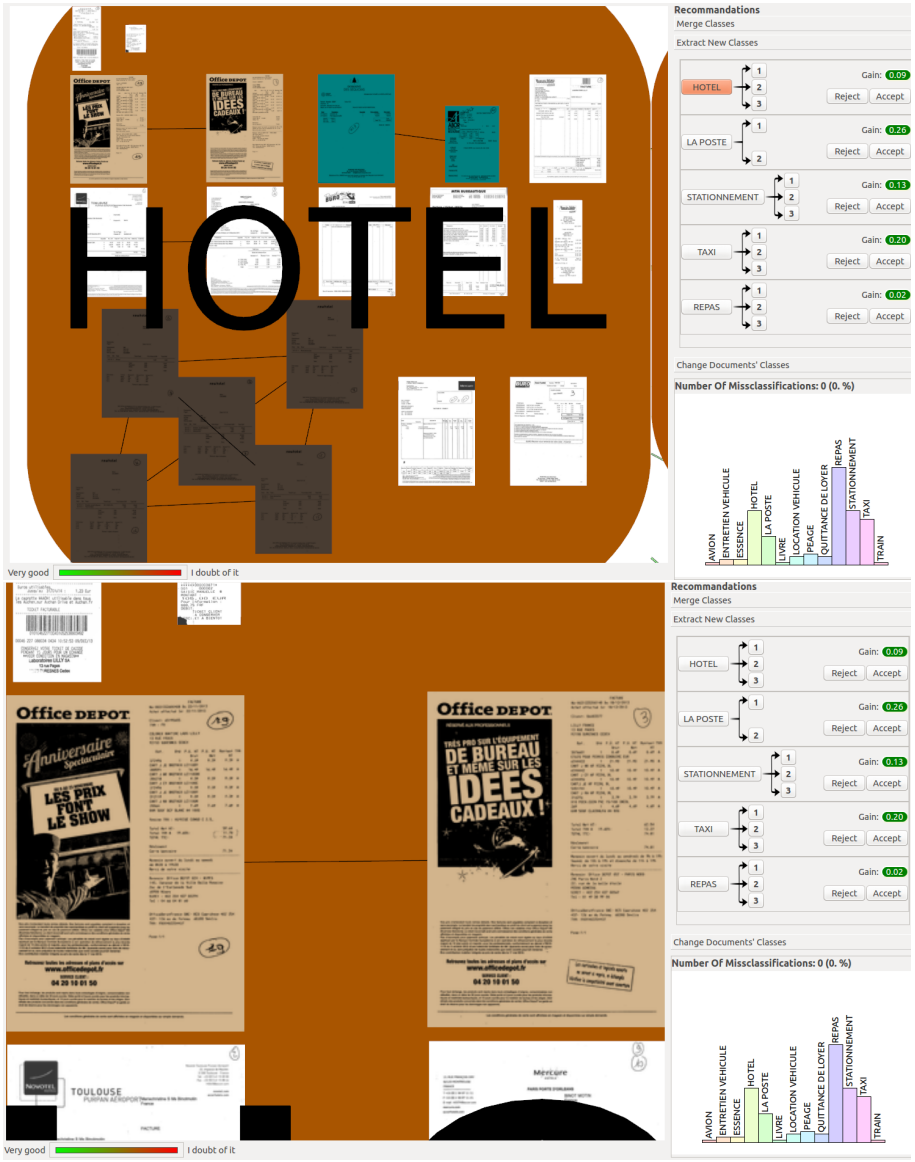

Figure 2. *Un exemple d'erreur de fusion. Dans ce cas, l'algorithme MCL propose de découper la classe « Hôtel » (a) en* 3 *sous-classes principales (en couleur sur l'image) ; une de ces sous-classes correspond aux documents de « Fournitures de Bureau » (b).*

tage au lieu de seulement les trouver. Ce dernier objectif est un problème d'optimisation du recalcul des graphes.

7. Bibliographie

Auber D., Mary P., Mathiaut M., Dubois J., Lambert A., Archambault D., Bourqui R., Pinaud B., Delest M., Melançon G. *et al.*, « Tulip : a Scalable Graph Visualization Framework », *Extraction et Gestion des Connaissances (EGC) 2010*, p. 623-624, 2010.

Augereau O., Journet N., Domenger J. P., « Document Images Indexing with Relevance Feedback : an Application to Industrial Context », *ICDAR*, p. 1190-1194, 2011.

Dwyer T., Marriott K., Stuckey P., « Fast Node Overlap Removal », *Proc. Graph Drawing 2005 (GD'05)*, p. 153-164, 2005.

Eler D. M., Nakazaki M. Y., Paulovich F. V., Santos D. P., Andery G. F., Oliveira M. C. F., Batista Neto J., Minghim R., « Visual Analysis of Image Collections », *Visual Computer*, vol. 25, n° 10, p. 923-937, 2009.

Enright A.J. Van Dongen S. O. C., « An efficient algorithm for large-scale detection of protein families », *Nucleic Acids Research*, 2002.

Fruchterman T. M., Reingold E. M., « Graph drawing by force-directed placement », *Softw., Pract. Exper.*, vol. 21, n° 11, p. 1129-1164, 1991.

Giot R., Bourqui R., Journet N., Vialard A., « Visual Graph Analysis for Quality Assessment of Manually Labelled Documents Image Database », *13th International Conference on Document Analysis and Recognition (ICDAR 2015)*, IAPR, 2015.

Hachul S., Jünger M., « Drawing large graphs with a potential-field-based multilevel algorithm », *Graph Drawing*, p. 285-295, 2005.

Mancoridis S., Mitchell B. S., Rorres C., Chen Y., Gansner E. R., « Using Automatic Clustering to Produce High-Level System Organizations of Source Code », *IEEE Proc. Int. Workshop on Program Understanding (IWPC'98)*, p. 45-53, 1998.

Plant W., Schaefer G., « Visualisation and browsing of image databases », *Multimedia Analysis, Processing and Communications*, Springer, p. 3-57, 2011.

Schneiderman B., « The Eyes Have It : A Task by Data Type Taxonomy for Information Visualizations », *Proc. of the IEEE Symposium on Visual Languages*, p. 336-343, 1996.

Thomas J. J., Cook K. A. (eds), *Illuminating the Path : The Research and Development Agenda for Visual Analytics*, IEEE Computer Society, 2006.

Ware C., *Information Visualization : Perception for Design*, Morgan Kaufmann Publishers, 2000.

Worring M., Engl A., Smeria C., « A Multimedia Analytics Framework for Browsing Image Collections in Digital Forensics », *Proceedings of the 20th ACM International Conference on Multimedia*, p. 289-298, 2012.