HAL open science

Optimisation de l'analyse en largeur pour une approche structurelle d'interprétation de croquis

Achraf Ghorbel, Eric Anquetil, Aurélie Lemaitre

To cite this version:

Achraf Ghorbel, Eric Anquetil, Aurélie Lemaitre. Optimisation de l'analyse en largeur pour une approche structurelle d'interprétation de croquis. RFIA 2012 : Congrès francophone sur la Reconnaissance des Formes et l'Intelligence Artificielle, Jan 2012, Lyon, France. hal-00646764v2

HAL Id: hal-00646764

https://hal.science/hal-00646764v2

Submitted on 5 Dec 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimisation de l'analyse en largeur pour une approche structurelle d'interprétation de croquis

Achraf Ghorbel[1] Eric Anquetil[1] Aurélie Lemaitre [2]

[1] IRISA/INSA-Rennes
[2] IRISA/Université Rennes 2

UMR IRISA, Campus de Beaulieu, F-35042 Rennes
Université Européenne de Bretagne, France
{achraf.ghorbel, eric.anquetil, aurelie.lemaitre}@irisa.fr

Résumé

Dans ce papier, nous présentons une approche optimisée d'analyse en largeur pour la reconnaissance interactive de documents structurés. Les méthodes d'analyse à base d'exploration en largeur permettent de confronter plusieurs hypothèses d'interprétation. Malheureusement, en pratique ces méthodes sont peu utilisées car elles engendrent souvent une trop grande combinatoire, difficile à maîtriser. Ce papier présente une stratégie d'optimisation afin de limiter la combinatoire. L'objectif est de démontrer la pertinence de l'exploration en largeur dans les méthodes d'analyse de documents notamment lorsqu'elles sont interactives, c'est à dire qu'elles intègrent l'utilisateur dans la boucle d'analyse. Ces optimisations ont été testées sur la reconnaissance d'images de plans d'architecture tracés à main levée.

Mots Clef

Exploration en largeur, document structuré, analyse interactive, interprétation de plans d'architecture.

Abstract

In this paper, we present an optimized approach, based on a competitive breadth-first exploration of the analysis tree, for an interactive interpretation of off-line sketch. The competitive breadth-first exploration of the analysis tree, allows to compare several hypotheses of interpretation to deal with confusion. Unfortunately, in practice these methods are rarely used because they often induce a large combinatory. This paper presents an optimization strategy to minimize the combinatory. The aim is to demonstrate the relevance of a competitive breadth-first exploration in off-line document analysis, in particular when the approach is interactive, ie the user is involved into the loop analysis. This paper demonstrates this optimized interactive analysis method on off-line handwritten 2D architectural floor plans.

Keywords

Breadth-first exploration, structured document, interactive recognition, 2D architectural floor plans interpretation.

1 Introduction

On identifie généralement, deux grandes approches d'analyse de document : les approches structurelles et statistiques. Le choix de l'une de ces deux approches dépend du type de document. L'approche structurelle [1] [2] [3] [4] exploite et s'appuie sur les connaissances a priori de la structure du document pour piloter l'analyse. Elles sont souvent basées sur des langages visuels qui permettent de décrire cette connaissance et de générer un analyseur. Malheureusement, les approches structurelles gère difficilement l'incertitude. Par contre, les approches statistiques [5] [6] sont particulièrement bien adaptées pour prendre en compte l'incertitude mais elles ne conservent pas la structure du document et nécessitent souvent une large base homogène d'apprentissage. Chaque type d'approche a des avantages et des inconvénients. L'interprétation des documents structurés manuscrits nécessite à la fois de conserver la structure du document mais aussi de bien gérer l'incertitude.

Nos récents travaux [7] (la méthode IMISketch) ont porté sur la conception d'une nouvelle approche de reconnaissance structurelle de documents, caractérisée par son interactivité. L'intégration de l'utilisateur dans la boucle de reconnaissance évite une correction a posteriori fastidieuse des erreurs de reconnaissance tout en permettant d'avoir un système auto-évolutif au fur et à mesure de l'analyse. Alors que la plupart des approches structurelles s'appuient sur une analyse en profondeur [4], nous nous sommes orientés sur la conception d'une analyse en largeur permettant de mettre en confrontation, pour un contexte d'analyse ciblé, plusieurs hypothèses d'interprétation concurrentes. Cette stratégie permet d'intégrer explicitement l'utilisateur dans la boucle d'analyse pour lever, si besoin, des ambiguïtés de reconnaissance [7], ou pour enrichir les connaissances ini-

tiales du système [8]. Ce système générique d'analyse peut s'appliquer aussi bien aux documents structurés hors-ligne, en-ligne et vectoriels.

Plusieurs travaux ont porté sur la reconnaissance des croquis. Cette reconnaissance nécessite une description de la structure du document. Hammond et Davis [9] [10] ont proposé la méthode LADDER pour l'interprétation en-ligne a posteriori des documents structurés manuscrits. Le langage LADDER a été exploité dans plusieurs domaines tels que les diagrammes UML [10], électrique [11] et les graphes complexes [12]. Yamamoto [16] et Bunke [17] utilisent des grammaires mono-dimensionnelles. D'autres techniques sont utilisées pour décrire les documents structurés bidimensionnels. Fahmy [18] et Bunke [19] ont proposé les grammaires de graphe. Ces grammaires ont été utilisées pour la modélisation des documents hors-ligne tel que les formules mathématiques. Même si les grammaires de graphes offrent un mécanisme intéressant pour la reconnaissance de forme, ces grammaires montrent des limites au niveau de l'implémentation, spécialement quand le nombre de production devient important. De plus, ces graphes sont mal adaptés pour la gestion du bruit et de l'incertitude. Dans la méthode IMISketch, nous adoptons le formalisme des grammaires à multi-ensembles à contraintes pilotée par le contexte (GMC-PC) [20] qui au départ ont été conçues pour la reconnaissance *en-ligne*.
Dans ce papier, nous développerons la méthode IMISketch pour l'analyse des documents structurés hors-ligne de type plans d'architecture manuscrits. La reconnaissance des plans d'architecture a été abordé dans plusieurs travaux de recherche. Lladós [13] a proposé une méthode pour la reconnaissance des plans d'architecture dessinés à main levée en utilisant la transformé de Hough et l'isomorphismes de graphes. Aoki [14] a proposé aussi une méthode d'interprétation des plans d'architecture qui consiste à transformer ces plans en une représentation CAO (conception assistée par ordinateur). Ahmed [15], de son coté, a proposé une technique à base de graphes pour la reconnaissance des plans d'architecture imprimés. Contrairement à toutes ces méthodes qui nécessitent une phase de vérification a posteriori fastidieuse, IMISketch, grâce à l'interactivité avec l'utilisateur, évite, ou en tout cas réduit fortement, cette phase de vérification.

IMISketch est caractérisé par une exploration en largeur qui peut engendrer une combinatoire importante. Dans ce papier, nous nous focalisons sur l'optimisation de la stratégie d'analyse afin de réduire la combinatoire pour aboutir à un analyseur qui soit réellement exploitable dans le cadre d'une analyse interactive ; c'est à dire avec un utilisateur dans la boucle qui peut être sollicité par l'analyseur en cas d'ambiguïtés trop complexes à résoudre automatiquement. Cette optimisation repose sur une limitation dynamique de la profondeur des branches de l'arbre d'analyse, qui est construit à chaque étape de l'exploration du document. Nous avons validé cette approche sur le système d'analyse interactive de document structuré décrit dans [7]. Les expérimentations ont comme support applicatif les plans d'architecture, dessinés à main levée.

Ce papier est organisé comme suit. La section 2 introduit les principes de base du système d'analyse interactive IMISketch pour l'interprétation de croquis. La section 3 présente la stratégie d'optimisation de l'analyseur en largeur pour contenir la combinatoire. Les résultats expérimentaux sont décrits dans la section 4 et finalement, la section 5 conclut ce papier.

2 Principes du système de reconnaissance IMISketch

Dans cette section, nous résumons les principes du système de reconnaissance interactive de document structuré IMISketch présentée dans la Figure 1. Ce système est basé sur les caractéristiques suivantes :

- les connaissances structurelles a priori du document sont exprimées à travers un langage visuel basé sur l'écriture de règles de production ;
- l'analyse repose sur une exploration en largeur ;
- l'incertitude est formalisée par l'attribution d'un score à chaque hypothèse sous tendue par une branche de l'arbre d'analyse ;
- l'analyseur travaille sur un contexte local de recherche pour limiter la combinatoire.
- si les ambiguïtés ne peuvent pas être résolues dans ce contexte local d'une manière automatique, l'utilisateur sera sollicité par l'analyseur pour lever l'ambiguïté.

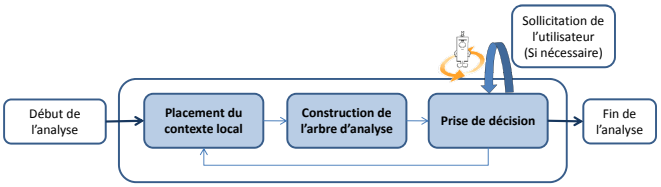


FIGURE 1 – Le processus d'analyse

Ces caractéristiques ont été définies afin d'assurer la meilleur interactivité entre le processus d'analyse et l'utilisateur. Cette interactivité permet notamment d'éviter la phase de vérification a posteriori qui peut devenir fastidieuse sur des documents complexes. En effet, en sollicitant à bon escient l'utilisateur sur des phases critiques de l'analyse du document, nous évitons une propagation en cascade d'erreur de reconnaissance, qui sont très lourdes à corriger a posteriori. La Figure 1 illustre les différentes phases de l'analyseur ainsi que la relation entre ces phases. L'analyseur commence par choisir le contexte local de recherche. Ce contexte est une zone d'intérêt spatiale du document, centré par l'élément se trouvant à droite de la règle de production appliquée à la racine, qui a pour but de limiter la combinatoire due à l'exploration en largeur de l'arbre d'analyse.
Une fois ce contexte défini, l'analyseur passe à l'étape de la construction de l'arbre d'analyse. Cet phase explore en

largeur toutes les hypothèses possibles en limitant leur profondeur grâce au contexte local de recherche défini dans la première étape. Une hypothèse correspond à une branche de l'arbre. Un score est associé à chaque branche (hypothèse). Ce score est déduit à partir des pré-conditions et des contraintes de la grammaire GMC-PC pour chaque règle de production appartenant à cette hypothèse. L'équation 1 décrit le calcul du score d'une règle de production. L'utilisation de la racine carrée correspond à une normalisation à l'aide d'une moyenne géométrique. L'équation 2 décrit le degré d'adéquation (score) d'une hypothèse. |PS| est le nombre de productions dans la branche considérée (la branche PS). Le nombre d'arbres considérés correspond au nombre de nouvelles règles de production déclenchées par la dernière règle validée.

$$\rho_P = \sqrt{\mu_{preconditions} \cdot \mu_{contraintes}} \quad (1)$$

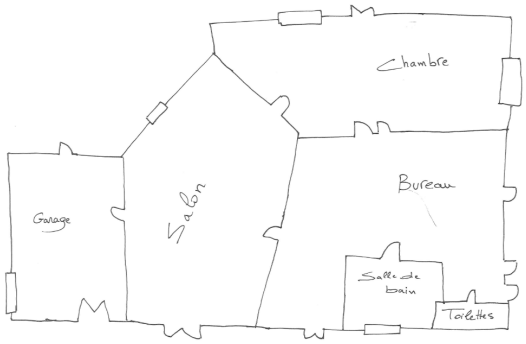
$$\rho_{PS} = \left(\prod_{P_i \in PS} \rho_{P_i} \right)^{\frac{1}{|PS|}} \quad (2)$$

Chaque hypothèse contient un ensemble d'objets complets ou incomplets. Un objet est dit complet si et seulement si cet objet peut se trouver dans le résultat final de l'interprétation de document. Par exemple, dans le cas de l'analyse de plan d'architecture, les objets complets peuvent être les murs, les portes, les fenêtres... Les objets incomplets sont des objets intermédiaires, qui ne peuvent pas exister dans le résultat final de l'analyse. Dans le cas des plans d'architecture, nous pouvons trouver des objets incomplets tel que le segment représentant le début d'un ouvrant.

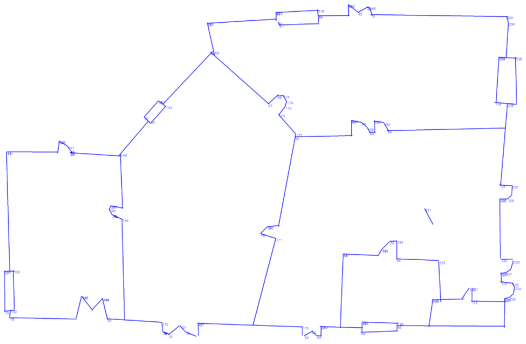
Dès que les arbres d'analyse sont explorés, l'analyseur entame la dernière phase : la phase de prise de décision. Le rôle de cette étape est de valider la racine de la meilleur hypothèse (branche) parmi l'ensemble des hypothèses en concurrence générées dans la deuxième phase.
Cette approche permet de prendre une décision sur la validation de la règle de production formant la racine, étant donné son contexte.
Il arrive que le processus d'analyse décide qu'il n'est pas capable de prendre la décision tout seul en validant la meilleur hypothèse (parce qu'elle a un score trop faible ou qu'elle entre en confusion avec d'autres hypothèses). Plus précisément, ce cas arrive par exemple si la différence de score entre les deux meilleures hypothèses est trop faible et que ces deux hypothèses sont contradictoires (elles partagent la(s) même primitive(s) mais interprétée(s) de manière différente). Le processus de prise de décision sollicite alors l'utilisateur et ce dernier validera la bonne hypothèse. Une fois la phase de prise de décision établie, le processus d'analyse reprend la première phase, en déplaçant le contexte d'analyse, pour continuer l'interprétation du document. L'analyse se poursuivra jusqu'à stabilité : aucune interprétation n'est plus possible.
Pour bien assimiler cette approche, nous présentons dans le paragraphe suivant un exemple d'analyse d'une partie de plan d'architecture.

Exemple d'analyse en largeur interactive

Nous exposons ici un exemple de reconnaissance d'images de plans d'architectures sommaires (composés de murs, portes et fenêtres) mais qui ont été dessinés à main levée. La description de la structure de ces documents est formalisées par des règles de production résumées dans la Figure 3. Ces règles traduisent qu'à l'extrémité d'un mur, nous pouvons trouver soit un mur soit un ouvrant (porte ou fenêtre). Un ouvrant est situé entre deux murs colinéaires. Une séquence désigne un ouvrant non complet. Nous trouvons dans ces règles, trois éléments complets : mur, porte et fenêtre, et deux objets incomplets : séquence et ouvrant. L'analyse dans cet exemple est faite sur un ensemble de primitives extrait d'un plan d'architecture. Ces primitives sont des segments extraits de l'image. Les Figures 2(a) et 2(b) montrent un exemple de plan d'architecture et l'ensemble des primitives (segments) associées.



(a) Image origine



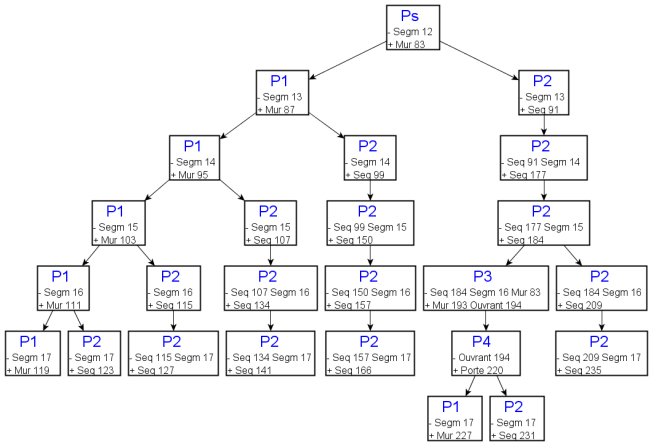
(b) Primitives extraites

FIGURE 2 – Exemple de primitives (segments) extraites (b) d'un plan d'architecture manuscrit (a)

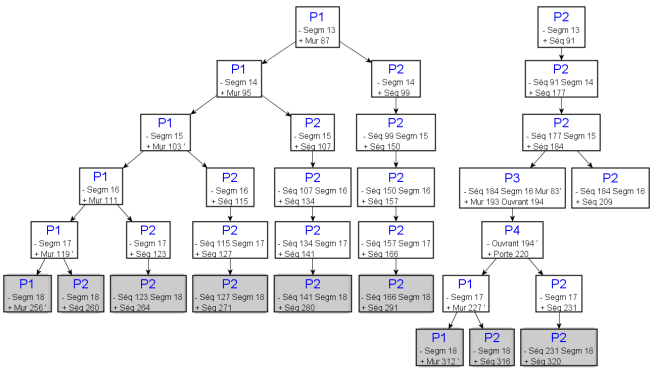
Après la phase d'extraction de segment, l'analyse définit le contexte local. Le positionnement de ce contexte dans le document dépend de l'élément consommé par l'interprétation racine. La Figure 5 montre le déplacement du contexte local entre deux phases de placement du contexte successives. Les phases de construction de l'arbre déduites de ce déplacement sont présentées dans la Figure 4. L'analyseur développe l'arbre d'analyse tant que les conditions suivantes (dénommé "Conditions IMISketch") sont vérifiées :

Nom de la règle	Éléments créés	Éléments consommés	Contexte
Ps	Mur de départ	Un segment	Dans le document
P1	Mur	Un segment	Extrémité d'un mur
P2	Séquence	• Un segment • Une séquence et segment	• Extrémité d'un mur • Extrémité d'une séquence
P3	Ouvrant et Mur	• Séquence et deux murs • Une séquence, un mur et un segment	• Deux murs colinéaires • Un mur et un segment sont colinéaires
P4	Porte	Un ouvrant	Interprété par le classifieur
P5	Fenêtre	Un ouvrant	Interprété par le classifieur

FIGURE 3 – Synthèse des règles de production pour les plans d'architecture



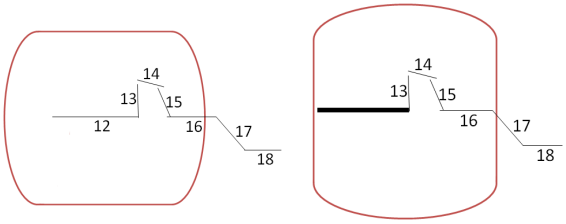
(a) Phase de construction de l'arbre d'analyse de l'étape n (segment 12), déduite de la Figure 5(a)



(b) Phase de construction de l'arbre d'analyse de l'étape n+1 (segment 13), déduite de la Figure 5(b). Les nœuds grisés indiquent les nouvelles règles de production appliquées en déplacement le contexte local.

FIGURE 4 – Arbres d'analyse associés aux Figures 5(a) et 5(b). (-) et (+) désignent respectivement l'élément consommé et l'élément créé.

1. l'élément nouvellement consommé est dans le contexte local de recherche ;
2. le nombre d'éléments consommés dans chaque hypothèse (branche) est inférieur à un seuil ;



(a) Placement du contexte local à l'étape n (centré sur le segment 12)

(b) Placement du contexte local à l'étape n+1 (centré sur le segment 13)

FIGURE 5 – Déplacement du contexte local de recherche au cours de l'analyse

Malgré l'utilisation du contexte local de recherche dans IMISkech, le problème de la combinatoire, et par conséquent des temps de calcul engendrés, est un problème important qu'il faut adresser pour répondre aux critères d'acceptabilité et d'utilisabilité de ce type d'analyse interactive avec un utilisateur. Ce problème est dû principalement au nombre de règles de production appliquées dans le contexte local. Nous remarquons que le développement de certains nœuds dans l'arbre d'analyse est inutile. Nous proposons dans la section suivante différentes stratégies d'optimisations au niveau du processus d'analyse pour remédier au problème de combinatoire dû au développement en largeur de l'arbre d'analyse.

3 Stratégies pour la réduction de la combinatoire : IMISktech optimisé

La construction de l'arbre en se basant sur une exploration en largeur permet d'avoir plusieurs hypothèses en concurrence. Malheureusement certaines règles de production appliquées dans un arbre sont invalidées par le système. Dans cette section, nous proposons un nouvel algorithme appliqué au niveau de la construction de l'arbre d'analyse afin de ne développer que les nœuds utiles pour la bonne prise de décision. Cet algorithme est décrit ci-dessous :

– Si le nombre d'arbre d'analyse est égal à 1 :
  1. se limiter sur le développement des fils directs de la racine. Puisque, il existe un seul arbre, le processus de décision valide systématiquement la racine de cet arbre. Le développement de l'arbre d'analyse jusqu'aux fils directs de la racine servira à déterminer les nouvelles racines à explorer.
– Si le nombre d'arbre d'analyse est supérieur à 1 :
  1. regrouper les arbres d'analyse par l'élément consommé dans la racine (l'élément se trouvant

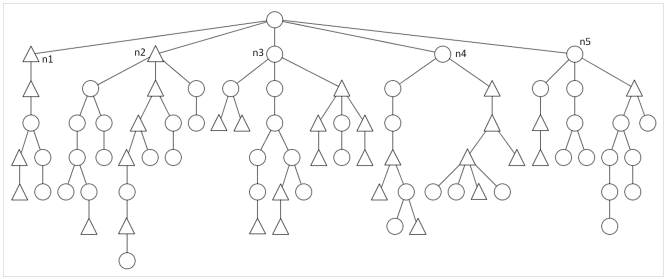
dans la partie droite d'une règle de production). Chaque groupe représente les différentes interprétations d'un élément consommé. Nous ne mettons donc en concurrence que les hypothèses d'un même groupe.

2. ordonner ces groupes par leur score. Ce score correspond au score de la meilleure hypothèse (branche) se trouvant dans le groupe.
3. ne développer que les racines appartenant au groupe ayant le meilleur score.
4. construire les arbres d'analyse tant que les conditions suivantes sont vérifiées :
   – l'élément consommé dans feuille de l'arbre se trouve dans le contexte local de recherche (déjà utilisé par IMISketch) ;
   – le nombre d'éléments consommés dans chaque hypothèse (branche) est inférieur à un seuil (déjà utilisé par IMISketch) ;
   – le nombre d'éléments complets créés dans une branche est inférieur à un seuil. Nous fixons ce seuil à 3 pour les plans d'architecture.

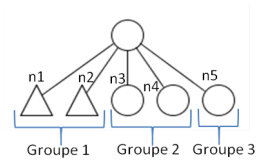
Cette optimisation peut engendrer une insuffisance d'information sur les hypothèses et donc une éventuelle ambiguïté. Mais grâce à l'interactivité, cette insuffisance impacte peu sur le résultat final dans la mesure où l'utilisateur peut être sollicité pour valider la bonne hypothèse.
L'ensemble de ces optimisations permet d'élaborer une stratégie interactive d'analyse de documents structurés basée sur une exploration en largeur de l'arbre d'analyse, en remédiant au problème majeur de l'analyse en largeur qui est l'explosion de la combinatoire.

3.1 Comparaison : IMISketch Vs IMISketch optimisé

Dans cet exemple, nous voulons montrer les améliorations découlant du nouvel algorithme pour la construction des arbres d'analyse. Pour faciliter cette comparaison, nous présentons un exemple d'arbre d'analyse synthétique. Le but est de comparer, pour chaque étape, le nombre de nœuds développés par IMISketch et celui induit par le nouvel algorithme de construction d'arbre d'analyse (IMISktech optimisé). La Figure 6(a) montre le résultat de l'exploration selon la méthode IMISketch. Le nombre d'interprétation (nœuds) est égal à 80 interprétations. La construction de l'arbre en se basant sur le nouvel algorithme engendre uniquement 6 interprétations (Figure 6(b)).
Les Figures 7(a) et 7(b) illustrent les nouveaux arbres d'analyse en suite au déplacement le contexte local.
Sur deux étapes successives de construction des arbres d'analyse pour les deux métodes (IMISketch et IMISketch optimisé), nous constatons qui nous sommes passés de 111 interprétations (IMISketch) à 28 interprétations (IMISketch optimisé), ce qui présente un gain d'environs 75% du temps de calcul. Ces optimisations sont génériques et ne dépendent pas de la catégorie de document structurés à analyser. La section suivante reporte une synthèse des résultats quantifiés sur un lot de 80 images de plans manuscrits.



(a) Méthode de construction de l'arbre d'analyse de **IMISketch**. Le nombre de nœuds nouvellement créés = 80



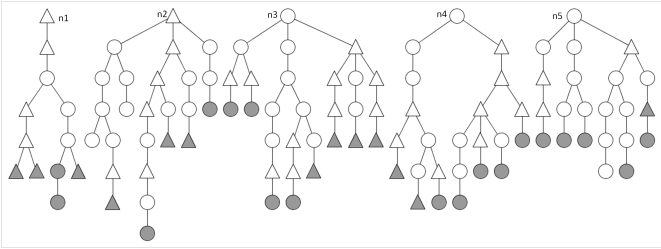
(b) Méthode de construction de l'arbre d'analyse de **IMISketch optimisé**. Le nombre de nœuds nouvellement créés = 6. Les fils directs de la racine sont groupés par éléments consommés.

FIGURE 6 – Arbre d'analyse à l'étape n. Les nœuds modélisés par des cercles désignent des objets complets. Les nœuds modélisés par des triangles désignent des objets incomplets.

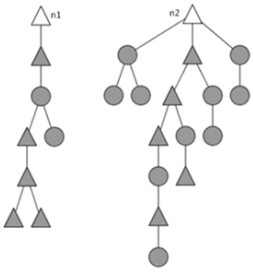
4 Résultats expérimentaux

Dans cette section, nous reportons les différents résultats obtenus suite aux optimisations décrites dans la section 3 sur la méthode interactive d'analyse de document. Le but de cette optimisation est de réduire les temps d'analyse du document pour répondre aux critères d'acceptabilité et d'utilisabilité du système. Les expérimentations ont portées sur 80 plans d'architecture de complexité différente dessinés par une dizaine de personnes différentes. Chaque plan d'architecture est composé d'un ensemble de murs, de portes, de fenêtres à frappe et de fenêtres coulissantes. La Figure 9 illustre le nombre de symboles trouvés. La Figure 8 montre quelques exemples de plans d'architecture analysés. Nous avons lancé le processus d'analyse sur les 80 plans deux fois. La première fois avec la méthode de base (IMISketch) et la deuxième fois avec les optimisations présentées dans la section 3 (IMISketch optimisé). Pour chaque plan d'architecture, nous comparons le résultat final de l'interprétation et les temps de calcul obtenus.

Au niveau du résultat final de l'interprétation, nous constatons que pour ces plans, les optimisations effectuées n'affectent pas les performances du système. En effet, nous



(a) Méthode de construction de l'arbre d'analyse de **IMISketch**. Le nombre de nœuds nouvellement créés = 31



(b) Méthode de construction de l'arbre d'analyse de **IMISketch optimisé**. Les racines n1 et n2 développées appartiennent au même groupe (Groupe 1) (elles partagent les mêmes éléments consommés). Le nombre de nœuds nouvellement créés = 22

FIGURE 7 – Arbre d'analyse à l'étape n+1. Les nœuds modélisés par des cercles désignent des objets complets. Les nœuds modélisés par des triangles désignent des objets incomplets. Les nœuds grisés correspondent aux nouvelles règles de production appliquées en déplaçant le contexte local de recherche.

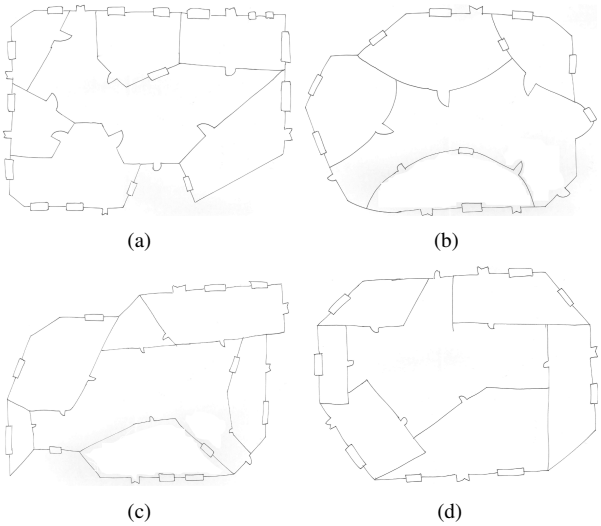


FIGURE 8 – Exemples de plans d'architecture analysés

Nombre de plans d'architecture	80
Nombre de murs	3179
Nombre de portes	681
Nombre de fenêtres	439
Nombre de fenêtres coulissantes	509
Nombre de symboles	4808

FIGURE 9 – Les symboles des plans d'architecture

remarquons une très légère différence entre IMISketch et IMISketch optimisé au niveau de la reconnaissance. Nous atteignons un taux brut de reconnaissance (sans intervention de l'utilisateur) avec IMISketch de 95.25% et avec IMISketch de 96.17% (cf. Figure 10). La sollicitation de l'utilisateur durant la phase d'analyse permet d'améliorer les performances de la reconnaissance. Nous atteignons alors 97.15% avec IMISketch et 97.31% avec IMISketch optimisé (cf. Figure 11). La Figure 12 illustre le pourcentage de la sollicitation de l'utilisateur lors de la boucle d'analyse par rapport au nombre total des racines interprétées ainsi que le pourcentage des sollicitations utiles par rapport au nombre total de sollicitation, c'est à dire les sollicitations qui permettent l'amélioration de la reconnaissance. À chaque interprétation d'un élément, les hypothèses proposées par les deux méthodes (IMISketch et IMISketch optimisé) ne sont pas les mêmes (Figures 6 et 7). IMISketch produit plus d'hypothèses que IMISketch optimisé 3.1. Même si le nombre d'hypothèses mises en concurrence est plus important pour IMISketch que pour IMISketch optimisé ce qui pourrait laisser penser qu'il y a avons plus de chance d'avoir la bonne hypothèse avec IMISketch, les confusions engendrées sont aussi potentiellement plus nombreuses. Au final, l'un dans l'autre, les résultats en terme de performance sont tout à fait comparables.

Les erreurs trouvées (environs 3%) sont dues soit à un mauvais calibrage du contexte local de recherche, soit au dessin approximatif de certains symboles. La Figure 13 montre quelques exemples de symboles "mal" dessinés. La Figure 14 illustre la différence de temps de calcul entre la méthode sans optimisation (IMISketch) et avec optimisation (IMISketch optimisé) pour chaque plan d'architecture analysé. Selon la complexité des plans, le gain en temps de calcul peut atteindre les 86% par image (image d'indice 62). La Figure 15 synthétise les temps cumulés de calcul sur le lot des 80 documents. Nous sommes passés de 16 heures et 22 minutes de temps de calculs cumulés à 7 heures et une minute ; ce qui représente un gain de 57%.

Les résultats expérimentaux sont très encourageants. Ils laissent entrevoir qu'il est tout à fait envisageable de mettre en place des analyses en largeur en maîtrisant la combinatoire engendrée. Cela conforte l'idée de concevoir des systèmes de reconnaissance interactive de document. Les sollicitations à l'utilisateur, pilotées par l'analyseur, garantissent l'obtention de très fort taux de fiabilité même si l'on

	IMISketch	IMISketch optimisé
Nombre de symboles non reconnus	228	184
Taux de reconnaissance	95.25%	96.17%

FIGURE 10 – Taux de reconnaissance des plans d'architecture manuscrits **sans** sollicitation de l'utilisateur

	IMISketch	IMISketch optimisé
Nombre de symboles non reconnus	137	129
Taux de reconnaissance	97.15%	97.31%

FIGURE 11 – Taux de reconnaissance des plans d'architecture manuscrits **avec** sollicitation de l'utilisateur

	IMISketch	IMISketch optimisé
Sollicitation de l'utilisateur	4.58%	3.77%
Sollicitation utile de l'utilisateur	16.42%	12.06%

FIGURE 12 – Le pourcentage de sollicitation de l'utilisateur pour les 80 plans d'architecture

(a) (b) (c)

FIGURE 13 – Exemple de symboles mal dessinés

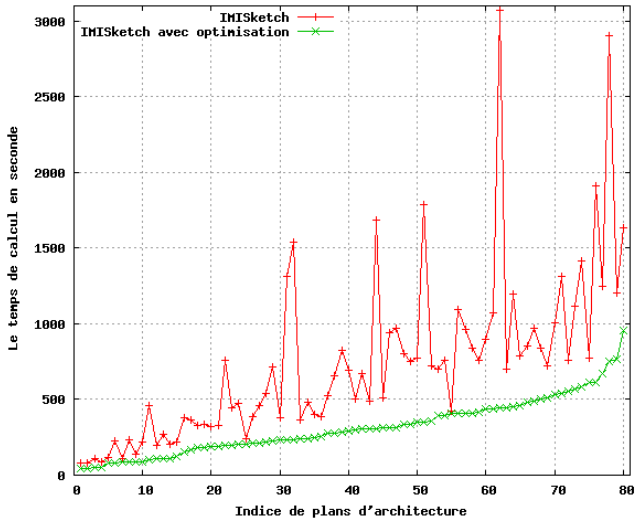


FIGURE 14 – Gain en temps de calcul par image

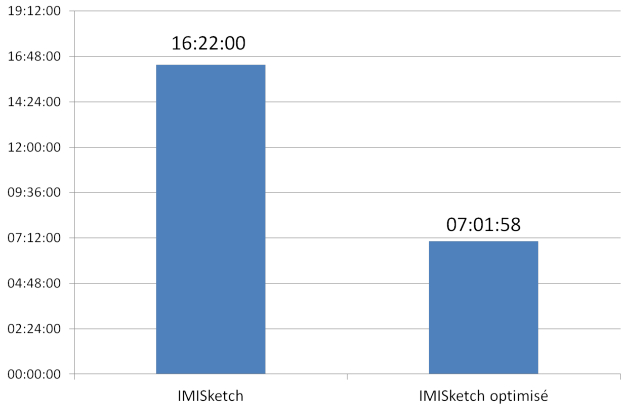


FIGURE 15 – Gain en temps de calcul cumulé

envisage le traitement de documents complexes.

5 Conculsion

Dans ce papier, nous avons présenté une stratégie d'optimisation de la combinatoire engendrée par une analyse interactive et en largeur de documents structurés. Cette stratégie repose sur une construction dynamique de l'arbre d'analyse en maîtrisant la profondeur de chacune des branches vis à vis d'un ensemble de critères défini dans la zone d'intérêt courante du document. Cette stratégie d'optimisation a été validée et testée sur un analyseur dans le contexte des plans d'architecture. Soulignons que cette approche d'optimisation est générique et par conséquent qu'elle pourrait facilement s'appliquer à d'autres types de documents structurés, ainsi qu'à d'autres analyseurs en largeur.

La prochaine étape de nos travaux portera sur le passage à l'échelle. L'objectif est d'étendre les résultats expérimentaux sur une base d'images plus large contenant des plans d'architecture plus complexes (intégration du mobilier, des cotations, etc.). Nous nous attacherons aussi à valider les critères d'acceptabilité et d'utilisabilité du système par des tests d'usage qui seront menés en collaboration avec les experts du Laboratoires des Usages "Loustic" (http ://www.loustic.net/rennes).

Remerciement

Les auteurs de ce papier remercient toutes les personnes qui ont participé aux différentes expérimentations. Ce travail est financé par le projet ANR Mobisketch.

Références

[1] K. Chan and D. Yeung, "An efficient syntactic approach to structural analysis of on-line handwritten mathematical expressions," *Pattern Recognition*, vol. 33, no. 3, pp. 375–384, 2000.

[2] J. Fitzgerald, F. Geiselbrechtinger, and T. Kechadi, "Mathpad : A fuzzy logic-based recognition system

for handwritten mathematics," in *ICDAR 2007*, vol. 2, 2007, pp. 694 –698.

[3] S. Mao, A. Rosenfeld, and T. Kanungo, "Document structure analysis algorithms : a literature survey," in *Proc. SPIE Electronic Imaging*, vol. 5010, 2003, pp. 197–207.

[4] B. Coüasnon, "Dmos, a generic document recognition method : Application to table structure analysis in a general and in a specific way," *IJDAR 2006*, vol. 8, no. 2, pp. 111–122.

[5] M. Lemaitre, E. Grosicki, E. Geoffrois, and F. Preteux, "Preliminary experiments in layout analysis of handwritten letters based on textural and spatial information and a 2d markovian approach," in *ICDAR 2007*, vol. 2, 2007, pp. 1023 –1027.

[6] F. Montreuil, E. Grosicki, L. Heutte, and S. Nicolas, "Unconstrained handwritten document layout extraction using 2d conditional random fields," *ICDAR 2009*, vol. 0, pp. 853–857.

[7] A. Ghorbel, S. Macé, A. Lemaitre, and E. Anquetil, "Interactive competitive breadth-first exploration for sketch interpretation," *to appear in ICDAR 2011*.

[8] A. Ghorbel, A. Almaksour, A. Lemaitre, and E. Anquetil, "Incremental learning for interactive for sketch reconition," *Ninth IAPR International Workshop on Graphics RECognition - GREC 2011*.

[9] T. Hammond and R. Davis, "Ladder, a sketching language for user interface developers," *Computers & Graphics*, vol. 29, no. 4, pp. 518–532, 2005.

[10] Hammond and Davis, "Ladder : A language to describe drawing, display, and editing in sketch recognition," 2003.

[11] C. Alvarado and R. Davis, "Sketchread : a multi-domain sketch recognition engine," in *ACM SIGGRAPH 2007 courses*. ACM, 2007, p. 34.

[12] T. Hammond and B. O'Sullivan, "Recognizing free-form hand-sketched constraint network diagrams by combining geometry and context," *Proceedings of the Eurographics Ireland 2007*, 2007.

[13] J. Lladós, J. López-Krahe, and E. Martí, "A system to understand hand-drawn floor plans using subgraph isomorphism and hough transform," *Machine Vision and Applications*, vol. 10, no. 3, pp. 150–158, 1997.

[14] Y. Aoki, A. Shio, H. Arai, and K. Odaka, "A prototype system for interpreting hand-sketched floor plans," in *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, vol. 3. IEEE, 1996, pp. 747–751.

[15] a. M. L. Sheraz Ahmed, M. Weber, and A. Dengel, "Improved automatic analysis of architectural floor plans," *ICDAR 2011*, pp. 864–868, 2011.

[16] R. Yamamoto, S. Sako, T. Nishimoto, and S. Sagayama, "On-line recognition of handwritten mathematical expressions based on stroke-based stochastic context-free grammar," in *Tenth International Workshop on Frontiers in Handwriting Recognition*. La Baule (France) : Suvisoft, Oct. 2006.

[17] H. Bunke, "String grammars for syntactic pattern recognition," *Syntactic and Structural Pattern Recognition. Theory and Applications*, pp. 29–54, 1990.

[18] H. Fahmy and D. Blostein, "A survey of graph grammars : theory and applications," in *Pattern Recognition, 1992. Vol.II. Conference B : Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on*, aug-3 sep 1992, pp. 294 –298.

[19] H. Bunke, "Attributed programmed graph grammars and their application to schematic diagram interpretation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-4, no. 6, pp. 574 –582, nov. 1982.

[20] S. Macé and E. Anquetil, "Eager interpretation of online hand-drawn structured documents : The dali methodology," *Pattern Recognition*, vol. 42, no. 12, pp. 3202–3214, 2009.