Reconnaissance de documents manuscrits structurés : Des équations manuscrites aux documents anciens

Harold Mouchère

Ecole Doctorale STIM (503)
Université de Nantes
IRCCyN

UNIVERSITÉ DE NANTES

Reconnaissance de documents manuscrits structurés

Des équations manuscrites aux documents anciens

Harold Mouchère

Mémoire d'Habilitation à Diriger des Recherches

Section CNU : 27/61

Soutenue le 2 décembre 2016

Jury :

Président :	Jean-Marc Ogier, Professeur des Universités à l'Université de La Rochelle
Rapporteurs :	Thierry Paquet, Professeur des Universités à l'Université de Rouen Jean-Yves Ramel, Professeur des Universités à Polytech Tours Antoine Tabbone, Professeur des Universités à l'Université de Lorraine
Examinateurs :	Eric Anquetil, Professeur des Universités à l'INSA de Rennes Christian Viard-Gaudin, Professeur des Universités à l'Université de Nantes

Remerciements

Je tiens à remercier tous ceux qui ont contribué de près ou de loin au travail présenté ici et à la rédaction de ce mémoire.

Je remercie pour son soutien Christian Viard-Gaudin qui a largement contribué à mon intégration dans l'équipe pédagogique du département GEII à l'IUT et dans mon équipe d'accueil IVC à l'IRCCyN. Notre collaboration au travers de nos nombreux projets communs a façonné mon projet de recherche en profondeur. Christian a accepté d'être mon garant HDR et de suivre la rédaction de ce document.

Un grand merci aussi à tous les collègues, co-auteurs, étudiants en thèse ou étudiants en stage qui ont participé à ces travaux.

Enfin, tout ceci ne serait pas possible sans le soutien de ma famille, particulièrement Estelle, qui m'a supporté pendant toutes ces heures de travail à la maison et qui a relu plusieurs fois ce manuscrit.

Table des matières

1 Curriculum Vitæ **1**

1.1 Activités pédagogiques 2

1.2 Activités en recherche 4

1.2.1 Projets de recherche 4

1.2.2 Encadrement doctoral 6

1.2.3 Collaborations scientifiques 8

1.2.4 Collaborations avec l'environnement socio-économique . . . 9

1.2.5 Présentation en Workshop interdisciplinaire 9

1.2.6 Rayonnement international 10

1.2.7 Publications et production scientifique 10

2 Introduction **23**

2.1 De la reconnaissance des documents structurés manuscrits 24

2.2 Principales contributions et plan du manuscrit 31

2.2.1 La reconnaissance d'expressions mathématiques manuscrites 31

2.2.2 Les compétitions CROHME 32

2.2.3 L'analyse de structures par les graphes 33

2.2.4 Perspectives et Projets de recherche 34

3 La reconnaissance d'expressions mathématiques manuscrites **35**

3.1 Projet CIEL - Thèse de M. Awal 35

3.1.1 Les données 35

3.1.2 Le graphe d'hypothèses, le classifieur et le rejet 36

3.1.3 L'analyse syntaxique 38
3.1.4 L'évaluation 40
3.2 Projet DEPART - Thèse de S. Medjkoune 41
3.2.1 Une base bimodale 42
3.2.2 La fusion de modalités 44
3.3 Grammaires de graphes - Thèse F. D. Julca-Aguilar 48
3.3.1 Utilisation du contexte 48
3.3.2 Grammaire de graphes 50
3.4 Utilisation des BLSTM - Thèse T. Zhang 52
3.4.1 Restriction au cas des expressions 1D 52
3.4.2 Reconnaissance d'expressions 2D 53

4 Les Compétitions CROHME 55
4.1 Organisation des compétitions 55
4.2 Les bases et corpus créés 56
4.3 Métriques d'évaluation 58

5 L'analyse de structures par les graphes 63
5.1 Reconnaissance de diagrammes - MRF 63
5.2 Extraction de connaissances symboliques - Thèse de J. Li 65
5.3 Reconnaissance de gestes multipoints - Thèse de Z. Chen 67

6 Perspectives et projets de recherche 71
6.1 Analyse de documents anciens 72
6.2 Vision industrielle 73
6.3 Apprentissage et réseaux de neurones 74
6.4 Prise en compte du contexte dans la reconnaissance 75
6.5 Positionnement national et international 76
6.6 Projets pédagogiques 77

7 Sélection d'articles **79**

7.1 A global learning approach for an online handwritten mathematical expression recognition system 80

7.2 Text Alignment from Bimodal Mathematical Expression Sources . . 90

7.3 Advancing the state of the art for handwritten math recognition : the CROHME competitions 2011–2014 95

7.4 An annotation assistance system using an unsupervised codebook composed of handwritten graphical multi-stroke symbols 112

Table des figures **125**

Chapitre 1

Curriculum Vitæ

Nom Mouchère

Prénom Harold

Date de naissance 22 avril 1981

Situation Marié, 3 enfants

Grade Maître de Conférences, Classe normale, 5^{e} échelon

Établissement Université de Nantes (IUT, département Génie Électrique et Informatique Industrielle)

Section CNU 61

Unité de recherche d'appartenance Laboratoire IRCCyN (UMR 6597) dans l'équipe Image et Vidéo Communication (IVC)

Thèse de doctorat "Étude des mécanismes d'adaptation et de rejet pour l'optimisation de classifieurs : Application à la reconnaissance de l'écriture manuscrite en-ligne."
INSA de Rennes, le 5 décembre 2007 à l'Irisa. Jury composé de :

- Heutte Laurent, Professeur à l'Université de Rouen (Rapporteur)
- Schomaker Lambert, Professeur à l'Université de Groningen, Pays-Bas (Rapporteur)
- Cardot Hubert, Professeur à l'Université de Tours (Examinateur)
- Miclet Laurent, Professeur à l'ENSSAT (Université de Rennes 1, Lannion) (Examinateur)
- Milgram Maurice, Professeur à l'Université Pierre et Marie Curie (Examinateur)
- Lorette Guy, Professeur à Université de Rennes I (Directeur)
- Anquetil Éric, Maître de Conférences à l'INSA de Rennes (Encadrant)

1.1 Activités pédagogiques

Cette section présente brièvement les différentes activités pédagogiques réalisées depuis mon intégration à l'Université de Nantes. Les différents enseignements et responsabilités liées sont triés par niveau.

DUT GEII

Les groupes TD de DUT GEII sont constitués de 24 à 28 étudiants.

Algorithmique et Langage C : Semestre 1, 30h TD, 21 TP

Compétence Projet : Semestre 2, 9h TD

Apprendre Autrement : Semestre 2, 18h TP en projet

Programmation Orientée Objet : Semestre 3, 6.7h C, 6.7h TD, 19h TP

Réseau : Semestre 3, 18h TP

Programmation Web : Semestre 4, 4h C, 12h TD, 10.7h TP

Bureau d'étude : Semestre 4, 20h à 30h TP en mode projet, en fonction des années

...

Environ 120h eqTD au total.

Responsabilités :

Bureau d'étude Dès mon arrivée en 2008 et jusqu'en 2013 j'ai pris en charge l'organisation d'une partie des BEs (collecte et affectation des sujets, organisation des soutenances et des évaluations),

Emploi du temps Depuis 2012 je m'occupe des emplois du temps des étudiants de première année du département (105 à 120 étudiants, 4 groupes). Développement d'un outil collaboratif pour la réalisation des progressions de toutes les formations du département,

Projets Tuteurés Depuis 2015 j'organise les projets transversaux, les étudiants mènent en autonomie leur projet en petit groupe.

Modules Responsable des modules : Algorithmique et Langage C, Programmation Orientée Objet et Programmation Web

Licence Pro SEICOM

Le groupe de Licence Pro SEICOM est constitué de 24 étudiants dont en moyenne une douzaine d'alternants.

Algorithmique et Langage C : 20h TD

Programmation Orientée Objet : 5.3h C, 8h TD, 12h TP

Systèmes Microprogrammés : 8h TP

Projets Tuteurés : Suivi des étudiants en semi autonomie (projet de 150h étudiant)

Environ 64h eqTD au total.

Responsabilités :

Responsable des stages et contrats : Validation des sujets, suivi des recherches, organisation des soutenances, ...

Suivi des alternants : tuteur de 2 à 3 étudiants en alternance tous les ans

Modules Responsable des modules : Algorithmique et Langage C, Programmation Orientée Objet

Évolutions Participation à l'élaboration du nouveau programme pédagogique pour la prochaine accréditation

Masters MDM et ATAL

Chacun des groupes comprend une douzaine d'étudiants. Le groupe MDM est anglophone.

Pattern recognition & Neural Networks : Présentation des MLP et SVM, leur apprentissage et utilisation, projet de reconnaissance des chiffres.

École Doctorale STIM

Module de formation doctorale STIM12 : *Intelligence artificielle : images, apprentissage et reconnaissances de formes*, sur ce module de 15h partagé avec N. Normand et C. Viard-Gaudin, j'interviens 5h C pour présenter des réseaux de neurones et l'utilisation du rejet

Autres activités pédagogiques

Je me suis investi dans la vie pédagogique de l'IUT à travers plusieurs activités :

Commission numérique (anciennement "outils numériques pour la pédagogie") : Composée de représentants des différents départements et du service SI, cette commission a pour rôle de favoriser les échanges sur les nouvelles pratiques pédagogiques utilisant les outils numériques, autant d'un point de vue pédagogique (organisation, formation) que technique (achat de matériel, choix technologique)

Organisation de journées pédagogiques : Deux journées pédagogiques ont été organisées en juillet 2015 et avril 2016 par un petit comité (C. Raillard, F. Bastianelli et moi-même) : formation aux outils numériques (Moodle), présentation de ressources partagées (IUT En Ligne), pédagogie inversée, word cafés autour des "jeux sérieux", tables rondes, ... (lien)

1.2 Activités en recherche

Mes activités de recherche se situent dans le domaine de la reconnaissance des formes appliquée aux documents structurés manuscrits. J'entends par documents structurés un document où la reconnaissance de l'organisation des éléments est aussi importante que la reconnaissance de ces éléments (équations mathématiques, diagrammes type "Flowcharts", ...)

1.2.1 Projets de recherche

Intégration dans le projet de recherche ANR CIEL

"Conversion Indexation de l'Ecriture en Ligne"
Budget du projet : 1 000 000€, fin en 2010
Trois partenaires sont impliqués dans ce projet :

- équipe IVC de l'IRCCyN,
- équipe Traitement Automatique du Langage Naturel du LINA
- l'entreprise Vision Objects (Nantes, depuis renommée MyScript).

Encadrement à 50% d'un doctorant (A.-M. Awal de 2008 à 2010)

Participation au projet Région Pays de la Loire DEPART

"Document Ecrits et Paroles – Reconnaissance et Traduction"
Volet "Développement des thématiques structurées et des spécialités scientifiques"
Budget du projet : 500 000€, de fin 2009 à janvier 2014
Trois partenaires institutionnels sont impliqués dans ce projet :

- équipe IVC de l'IRCCyN
- équipe Traduction et Reconnaissance de la Parole du LIUM
- équipe Traitement Automatique du Langage Naturel du LINA

Participation à l'écriture du projet et des livrables. Responsabilités dans la constitution de la base de données d'écriture manuscrite : protocole, collecte, extraction, étiquetage, mise à disposition.

Encadrement à 30% d'un doctorant (S. Medjkoune depuis 2010)

Montage et participation au Projet SCENIC

"SCénarios pour l'ExploitatioN et l'enrichissement Interactif de Collections de documents anciens"
15 000€, fin 2013 - début 2014
Projet de soutien de Atlanstic pour le montage de nouvelles collaborations.
Porteur du projet : C. Viard-Gaudin

Trois partenaires institutionnels sont impliqués dans ce projet :

— équipe IVC de l'IRCCyN
— équipe Reconnaissance de Formes et Analyse d'Images du Laboratoire d'Informatique de Tours (EA 6300)
— Centre d'Études des Théâtres de la Foire et de la Comédie Italienne du L'AMO (EA 4276)

Encadrement d'un ingénieur de recherche pendant 3 mois.

Cette première collaboration inter-disciplinaire avec une équipe SHS a permis le montage du projet ANR CIRESFI accepté fin 2014.

Projet ANR CIRESFI

"Contrainte et Intégration : pour une RéÉvaluation des Spectacles Forains et Italiens sous l'Ancien Régime"
Budget du projet : 400 000€, de Octobre 2014 à 2019
Défi sociétal numéro 8 : Sociétés Innovantes, Intégrantes Et Adaptatives
Axe : Création, cultures et patrimoines
Coordinateur : Françoise Rubellin, Université de Nantes – Laboratoire L'AMO

Neuf partenaires institutionnels sont impliqués dans le projet dont :

— Centre d'Études des Théâtres de la Foire et de la Comédie Italienne du L'AMO
— l'équipe IVC de l'IRCCyN
— l'équipe DUKE du LINA

Participation à l'écriture du projet. Il s'agit d'un projet transdisciplinaire pour la valorisation de 30000 pages de registres numérisées par la BNF pour le LAMO en 2013. Notre contribution consiste à proposer des solutions d'analyse de documents anciens pour automatiser l'annotation des documents et y faciliter la recherche d'information avec des requêtes de type image ou textuelle.

Encadrement à 40% d'un doctorant (A. Granet depuis 2015) et de stagiaires M2 à 50% (Weishen PAN en 2015, Hansil RYU en 2016).

1.2.2 Encadrement doctoral

Thèse de J. Langlois : (début en mai 2016, fin prévue en 2019) thèse CIFRE avec l'entreprise MultitudeTechnologie, co-encadrement 40% avec C. Viard-Gaudin (Prof. Univ. Nantes, directeur de thèse, 40%), N. Normand (Mcf Univ. Nantes, 20%) "**Vision industrielle par réseaux de neurones profonds**"

Thèse de A. Granet : (début en oct. 2015, fin prévue 2018) thèse en co-encadrement 40% avec l'équipe TALN du LINA : E. Morin (Prof. Univ. Nantes, directeur de thèse, 40%) et S. Quiniou (Mcf. Univ. Nantes, 20%) "**Extraction d'informations dans des collections historiques**"

Thèse de T. Zhang : (début en dec. 2014, fin prévue 2017), 60% avec C. Viard-Gaudin (Prof. Univ. Nantes,directeur de thèse, 40%) "**New Architecture for Handwritten Mathematical Expressions Recognition**", Publications liées : conférence internationale : [13], conférence nationale : [52]

Thèse de Z. Chen : (début en oct 2013, fin prévue 2016), thèse co-financée par les régions Bretagne et Pays de la Loire, co-encadrement à 33% avec E. Anquetil (Prof. INSA de Rennes, directeur de thèse, 33%) et C. Viard-Gaudin (Prof. Univ. Nantes, 33%) "**Reconnaissance et interprétation des interactions tactiles multipoints**" Publications liées : conférences internationales : [15, 18, 19]

Thèse de F. D. Julca-Aguilar : (début en oct. 2013, thèse soutenue en avril 2016) thèse en co-tutelle avec l'Université de Sao Paulo (avec Nina Hirata, Ass. Prof.), co-encadrement à 50% avec C. Viard-Gaudin (Prof. Univ. Nantes, directeur de thèse, 50%) "**Structural Analysis of Handwritten Mathematical Expressions using Contextual Information**" sur HAL Publications liées : conférences internationales : [12, 8, 20] conférence nationale : [53] Actuellement ingénieur-recherche à l'Université de Sao Paulo.

Thèse de S. Medjkoune : (début en oct 2010, fin 13 Nov 2013) dans le cadre du projet DEPART, co-encadrement à 30% avec C. Viard-Gaudin (Prof. Univ. Nantes, directeur de thèse, 40%) et Simon Petitrenaud de l'Université du Mans (co-encadrant, 30%) "**Stratégies de fusion pour des signaux écrits et sonores – Application à la reconnaissance d'expressions mathématiques**" sur HAL Publications liées : chapitre de livre : [10] Revue Fr : [4] conférences internationales : [20, 21, 23, 28, 34, 35] conférences nationales : [53, 55, 56] Actuellement en Post-doctorat à l'Ecole des Mines de Douai.

Thèse de J. Li : (début en oct. 2009, thèse soutenue le 23/10/2012), 60% avec C. Viard-Gaudin (Prof. Univ. Nantes,directeur de thèse, 40%) "**Extraction**

de connaissances symboliques et relationnelles appliquée aux tracés manuscrits structurés en-ligne" sur HAL Publications liées : revue : [3, 5] conférences internationales : [26, 27, 30, 33] conférences nationales : [54] Post-doctorat au CEA (Saclay) puis ingénieur recherche au CEA.

Thèse de M. Awal : (encadrement à partir de 2008, thèse soutenue le 12/11/2010), 50% avec C. Viard-Gaudin (Prof. Univ. Nantes) "**Reconnaissance de structures 2D dans les documents manuscrits : application aux équations mathématiques**" sur HAL Publications liées : conférences internationales [43, 41, 40, 39, 31] ; conférences nationales[38, 57] Actuellement Ingénieur R&D chez Ariadnext (vérification de documents d'identité)

Encadrement de masters de recherche

Hansil RYU, 2016 (Master MDM) "Low Quality Video-Surveillance Analysis" (encadrement à 50% avec C. Viard-Gaudin à 50%)

Jingwen HUANG, 2016 (Master MDM) "Semi-automatic image segmentation of old documents " (encadrement à 50% avec C. Viard-Gaudin à 50%)

Weishen PAN, 2015 (Master MDM) "Distance Metric Learning from Multilayer Perceptron to Convolutional Neural NetworkApplication for Word Spotting in Historical Documents" (encadrement à 50% avec C. Viard-Gaudin à 50%)

Chengcheng WANG, 2015 (Master SEGE) "Understanding Handwritten 2d Structural Language Using Conditional Random Field in the Example of Flowchart" (encadrement à 50% avec C. Viard-Gaudin à 50%) Une conférence internationale [16] et une revue internationale en soumission

Ram Hari, 2014 (Master MDM) "Spatial Relationship Identification for Handwritten Mathematical Expressions Recognition" (encadrement à 50% avec C. Viard-Gaudin à 50%)

Junbei Shang, 2014 (Master MDM) "Bidirectional Long Short Term Memory Classifier for Handwritten Mathematical Expressions Recognition " (encadrement à 50% avec C. Viard-Gaudin à 50%)

Bingwei WU, 2013 (Master MDM) "Two-Dimensional (2d) Languages and Application to Handwritten Graphical Parsing" (encadrement à 50% avec C. Viard-Gaudin à 50%)

Xiaoxin WEI, 2013 (Master MDM) "From Recurrent Neural Network to Long Short Term Memory Architecture" (encadrement à 50% avec C. Viard-Gaudin à 50%)

Boussad GHEDAMSI, 2013 (Master MRI de l'ISTIC, Univ. Rennes) "Mutitouch Gestures Recognition Using the Strategy of Graph Embedding" (encadrement à 50% avec E. Anquetil, INSA de Rennes)

Zhaoxin CHEN, 2012 (Master SEGE) "A Dynamic Time Warping-A* Handwriting Recognition System", (encadrement à 60% avec C. Viard-Gaudin à 40%), une publication liée [5] (revue), étudiant en actuellement en thèse

S. Rahmoun, 2012 (Master COLQ) "Composition Manuscrite De Documents Structurés Sur Des Interfaces Tactiles Multipoints", (encadrement à 40% avec E. Anquetil, Prof. INSA de Rennes, 60%)

J. Li, 2009 (Master SEGE) "Writer Identification Based on Grapheme Distribution" Résultats publiés dans un papier "jeune chercheur" à la conférence CIFED 2010 [0]

A. Kanj, 2009 (Master SEGE) "Handwritten Mathematical Expression Recognition : Symbol Recognition Optimization"

Hanyu Yan, 2009 (Master SCUT) accueil de cet étudiant de South China University of Technology (SCUT) qui s'est concrétisé par un papier en conférence international [42]

1.2.3 Collaborations scientifiques

Collaborations internationales

— en 2011 et 2012 avec le **Pr. Kim J. H. du KAIST (Corée du Sud)** : création de la compétition CROHME [36, 29]

— depuis 2011 avec **Associate Pr. Utpal Garain du ISICAL (Inde)** : création puis organisation annuelle de la compétition CROHME [36, 29, 24, 22, 0, 1]

— depuis 2011 avec **Associate Pr. Richard Zanibbi du Rochester Institute of Technology (NY, USA)** : une publication en conférence internationale [37] sur l'évaluation de la reconnaissance des expressions mathématiques puis une visite d'un mois en juin 2012 dans l'équipe IVC qui donne lieu à une nouvelle publication en conférence internationale sur le même thème [25]. Depuis 2013, R. Zanibbi a pris part à l'organisation de la compétition CROHME [24, 22, 0] et nous avons pu valoriser tout ce travail à travers un article de revue [1]

— en 2013 avec **Guihuan FENG de Software Institute (Nanjing University)** : invitée un mois dans l'équipe IVC, cette collaboration sur la reconnaissance de diagramme fait suite à une collaboration antérieure depuis 2010 [31]

— en octobre 2013 accueil de Frank Julca Aguilar, doctorant de l'Université de Sao Paulo (Brésil) (encadrement de **Nina Hirata, Associate Prof.**

à l'USP) puis en co-tutelle avec l'Université de Nantes depuis oct 2013 (encadrement Prof. C. Viard-Gaudin et H. Mouchère). En novembre 2014 accueil de Nina Hirata dans l'équipe IVC dans le cadre de la thèse en co-tutelle de Frank Julca Aguilar

— depuis 2009 collaboration avec **Pr. Lianwen Jin du SCUT (Chine, Guanzu)** qui se traduit par l'accueil régulier d'étudiants de SCUT en master (Hanyu Yan, Chengcheng Pan) et deux publications [16, 42]

Collaborations nationales

Collaborations régulières avec l'équipe IntuiDoc de Irisa (à Rennes, équipe anciennement nommée IMADOC) :

— sur la reconnaissance de diagrammes par une approche grammaticale, d'abord avec A. Delaye [58] en conférence nationale puis avec A. Lemaitre avec des publications internationales régulières [32, 9]

— sur la reconnaissance de gestes manuscrits multipoints avec la Thèse de Z. Chen (directeur E. Anquetil)

1.2.4 Collaborations avec l'environnement socio-économique

Projet ANR CIEL avec la société MyScript (Nantes) leader mondial des solutions de reconnaissance de l'écriture en-ligne. Intégration dans la compétition CROHME.

Transfert industriel avec la société Excence (Rennes) Valorisation des travaux de la Thèse de Z. Chen, expertise dans les interactions homme-machine multipoints, réalisation d'un prototype pour la reconnaissance de gestes dans le cadre d'une application multipoint avec plusieurs utilisateurs, en collaboration avec l'équipe Intuidoc (Irisa, Rennes).

Thèse CIFRE avec la société Multitude-Technologie Expertise dans la vision par ordinateur pour la détection et manipulation de pièces en plastique.

1.2.5 Présentation en Workshop interdisciplinaire

L'écriture, un micro-geste de la main Lors du séminaire *Les journées du geste "Main mouvement et émotion"* organisé par Anne Dubos à Nantes les 6-7 novembre 2014. Ce séminaire réunissait différents spécialistes (ethnologues, danseurs, linguistes, historiens, ...) autour de la thématique de la conservation et la numérisation du patrimoine gestuel.

1.2.6 Rayonnement international

Invitation à l'étranger

Professeur invité 3 semaines en juin 2013 au Rochester Institute of Technology (RIT), Rochester N.Y. dans la continuité de la collaboration avec Richard Zanibbi.

Participations à des jurys de thèse à l'étranger

F. Àlvaro Muñoz "Mathematical Expression Recognition based on Probabilistic Grammars", à l'Universitat Politècnica de València (Espagne), sous la direction de Dr. Joan Andreu Sánchez and Prof. José Miguel Benedí, soutenue en 2015

L. Hu "Features and Algorithms for Visual Parsing of Handwritten Mathematical Expressions", à Rochester Institute of Technology (USA), sous la direction de Richard Zanibbi, soutenue en 2016

Animation de la recherche

Relecture pour des conférences internationales ICDAR, ICPR, ICFHR

Relecture pour des revues internationales International Journal on Document Analysis and Recognition (IJDAR), International Journal of Computer Mathematics (IJCM), Pattern Recognition Letter (PRL), Pattern Recognition (PR)

Organisation de la compétition internationale CROHME

Organisateur de la compétition depuis sa création en 2011 puis en 2012, 2013, 2014 et 2016, cette compétition rassemble à chaque occurrence entre 5 et 9 laboratoires et entreprises du domaine de la reconnaissance d'écriture. Véritable catalyseur de collaborations et de productions, cette compétition sera détaillée dans le chapitre 4.

1.2.7 Publications et production scientifique

Ces publications viennent concrétiser mon travail d'encadrement des doctorants et des étudiants de master mais aussi mes collaborations nationales et internationales. Le calcul des index et du nombre de citations est basé sur les statistiques de GoogleScholar depuis 2008.

H-index	14
i10-index	20
Publications	62
Articles en revues internationales	4 de rang A + 1
Articles en revues nationales	2
Chapitres de livres	4
Publications en conférences internationales	39
Publications en conférences nationales	11

Mes travaux en cours ont conduit à la soumission de 3 autres publications en cours de relecture :

— Article pour la revue Journal of Human-Machine Systems en révision mineure : **Combining Speech and Handwriting Modalities for Mathematical Expression Recognition**, Sofiane Medjkoune, Harold Mouchère, Simon Petitrenaud, et Christian Viard-Gaudin.

— Article pour la revue International Journal on Document Analysis and Recognition : **Online Flowchart Understanding by Combining Max-margin Markov Random Field with Grammatical Analysis**, Chengcheng Wang, Harold Mouchère, Aurélie Lemaitre, Christian Viard-Gaudin

— Chapitre de livre : **Handwritten Mathematical Expressions**, Harold Mouchère, Christian Viard-Gaudin, Richard Zanibi, Utpal Garain, pour Document Analysis and Text Recognition : Benchmarking State-of-the-Art Systems, Series in Machine Perception and Artificial Intelligence (SMPAI), World Scientific Publishing Co.

Liste des publications

Articles de revues avec comité de relecture

[1] Harold Mouchère, Richard Zanibbi, Utpal Garain, Christian Viard-Gaudin. "Advancing the state of the art for handwritten math recognition : the CROHME competitions, 2011–2014". In : *International Journal on Document Analysis and Recognition (IJDAR)* (2016), p. 173-189. ISSN : 1433-2825. DOI : 10.1007/s10032-016-0263-5.

[2] Ahmad-Montaser Awal, **Harold Mouchère**, Christian Viard-Gaudin. "A global learning approach for an online handwritten mathematical expression recognition system". In : *Pattern Recognition Letters* 35 (jan. 2014), p. 68-77. ISSN : 0167-8655. DOI : 10.1016/j.patrec.2012.10.024.

[3] Jinpeng Li, Harold Mouchère, Christian Viard-Gaudin. "An annotation assistance system using an unsupervised codebook composed of handwritten graphical multi-stroke symbols". In : *Pattern Recognition Letters* 35.1 (jan. 2014), pp. 46-57. DOI : 10.1016/j.patrec.2012.11.018.

[4] Sofiane Medjkoune, **Harold Mouchère**, Simon Petitrenaud, Christian Viard-Gaudin. "Approches multimodales pour la reconnaissance d'expressions mathématiques". In : *Revue des Sciences et Technologies de l'Information - Série Document Numérique* 17.3 (2014), p. 97-122. URL : www.cairn.info/revue-document-numerique-2014-3-page-97.htm.

[5] Harold Mouchère, Jinpeng Li, Christian Viard-Gaudin, Zhaoxin Chen. "A dynamic Time Warping Algorithm for Recognition of Multi-Stroke On-Line Handwriten Characters". In : *Natural Science Edition, Journal of South China University of Technology* 41.7 (juin 2013), p. 107-113. DOI : 10.3969/j.issn.1000-565X.2013.07.000.

[6] Abdullah Almaksour, Harold Mouchère, Eric Anquetil. "Apprentissage incrémental avec peu de données pour la reconnaissance de caractères manuscrits en-ligne". In : *Traitement du Signal* (déc. 2009). URL : https://hal.archives-ouvertes.fr/hal-00491329.

[7] **Harold Mouchère**, Eric Anquetil, Nicolas Ragot. "Writer Style Adaptation in On-line Handwriting Recognizers by a Fuzzy Mechanism Approach : The ADAPT Method". In : *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)* 21.1 (2007), p. 99-116. DOI : 10.1142/S0218001407005326.

Chapitres de livre

[8] F. Aguilar, **Harold Mouchère**, C. Viard-Gaudin, N. Hirata. "Top-Down Online Handwritten Mathematical Expression Parsing with Graph Grammar". In : *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications 9423. Montevideo, Uruguay : Springer International Publishing, 2015, p. 444-451. DOI : 10.1007/978-3-319-25751-8_53.

[9] Aurélie Lemaitre, Harold Mouchère, Jean Camillerapp, Bertrand B. Couasnon. "Interest of syntactic knowledge for on-line flowchart recognition". In : *Graphics Recognition New Trends and Challenges. 9th International Workshop, GREC 2011, Seoul, Korea, September 15-16, 2011, Revised Selected Papers*. Sous la dir. d'Young-Bin KWON et Jean-Marc OGIER. T. 7423. Lecture Notes in Computer Science. Springer Berlin Heidelberg, déc. 2013, p. 89-98. DOI : 10.1007/978-3-642-36824-0_9.

[10] Sofiane Medjkoune, Harold Mouchère, Simon Petitrenaud, Christian Viard-Gaudin. "Multimodal Mathematical Expressions Recognition : Case of Speech and Handwriting". In : *Human-Computer Interaction. Interaction Modalities and Techniques. 15th International Conference, HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part IV*. Sous la dir. de Masaaki KUROSU. T. 8007. Lecture Notes in Computer Science. Berlin, Heidelberg : Springer Berlin Heidelberg, juil. 2013, p. 77-86. DOI : 10.1007/978-3-642-39330-3_9.

[11] Patrizio Campisi, Emanuele Maiorana, Harold Mouchère, Christian Viard-Gaudin, Alessandro Neri. "Encyclopedia of Cryptography and Security". In : sous la dir. d'Henk C. A. van TILBORG et Sushil JAJODIA. Boston, MA : Springer US, 2011. Chap. Handwriting Analysis, p. 531-534. ISBN : 978-1-4419-5906-5. DOI : 10.1007/978-1-4419-5906-5_886.

Conférences internationales avec comité de relecture

[12] Frank Julca-Aguilar, Nina S. T. Hirata, Harold Mouchère, Christian Viard-Gaudin. "Subexpression and Dominant Symbol Histograms for Spatial Relation Classification in Mathematical Expressions". In : *23rd International Conference on Pattern Recognition (ICPR2016)*. Cancun, Mexique, 2016. URL : https://hal.archives-ouvertes.fr/hal-01374383.

[13] Ting Zhang, **Harold Mouchère**, Viard-Gaudin Christian. "Online Handwritten Mathematical Expressions Recognition by Merging Multiple 1D Interpretations". In : *Int. Conf. on Frontier in Handwriting Recognition (ICFHR)*. Shenzhen, Chine, 2016. URL : https://hal.archives-ouvertes.fr/hal-01374392.

[14] Zhong Zhuoyao, Weishen Pan, **Harold Mouchère**, Viard-Gaudin Christian. "SpottingNet : Learning the Similarity of Word Images with Convolutional Neural Network for Word Spotting in Handwritten Historical Documents". In : *Int. Conf. on Frontier in Handwriting Recognition (ICFHR)*. Shenzhen, Chine, 2016. URL : https://hal.archives-ouvertes.fr/hal-01374401.

[15] Zhaoxin Chen, Eric Anquetil, **Harold Mouchère**, Viard-Gaudin Christian. "The MUMTDB dataset for evaluating simultaneous composition of structured documents in a multi-user and multi-touch environment". In : *Int. Conf. on Frontier in Handwriting Recognition (ICFHR)*. Shenzhen, Chine, 2016. URL : https://hal-uag.archives-ouvertes.fr/hal-01374376.

[16] Chengcheng Wang, **Harold Mouchère**, Viard-Gaudin Christian, Lianwen Jin. "Combined Segmentation and Recognition of Online Handwritten Diagrams with High Order Markov Random Field". In : *Int. Conf. on Frontier in Handwriting Recognition (ICFHR)*. Shenzhen, Chine, 2016. URL : https://hal.archives-ouvertes.fr/hal-01374389.

[17] Ting Zhang, **Harold Mouchère**, Viard-Gaudin Christian. "On-line Handwritten Isolated Symbol Recognition using Bidirectional Long Short-term Memory (BLSTM) Networks". In : *Third Sino-French Workshop on Education and Research collaborations in Information and Communication Technologies SIFWICT 2015*. 2015, p. 217-232.

[18] Zhaoxin Chen, Eric Anquetil, **Harold Mouchère**, Christian Viard-Gaudin. "Recognize multi-touch gestures by graph modeling and matching". In : *17th Biennial Conference of the International Graphonomics Society. Drawing,*

Handwriting Processing Analysis : New Advances and Challenges. Pointe-à-Pitre, Guadeloupe, 2015.

[19] Zhaoxin Chen, Eric Anquetil, Harold Mouchère, Christian Viard-Gaudin. "A graph modeling strategy for multi-touch gesture recognition". In : *14th International Conference on Frontiers in Handwriting Recognition (ICFHR-2014)*. Crete island, Greece, sept. 2014. URL : https://hal.inria.fr/hal-01088774.

[20] Frank Julca-Aguilar, Nina S. T. Hirata, Christian Viard-Gaudin, Harold Mouchère, Sofiane Medjkoune. "Mathematical symbol hypothesis recognition with rejection option". In : *14th International Conference on Frontiers in Handwriting Recognition*. Crete, Greece, sept. 2014, p. 500-504. DOI : 10.1109/ICFHR.2014.90.

[21] Sofiane Medjkoune, Harold Mouchère, Christian Viard-Gaudin, Simon Petitrenaud. "Text Alignment from Bimodal Mathematical Expression Sources". In : *14th International Conference on Frontiers in Handwriting Recognition*. Crete, Greece, sept. 2014, p. 205-209. DOI : 10.1109/ICFHR.2014.42.

[22] **Harold Mouchère**, Christian Viard-Gaudin, Richard Zanibbi, Garain Utpal. "ICFHR 2014 - Competition on Recognition of On-line Mathematical Expressions (CROHME 2014)". In : *Int. Conf. on Frontier in Handwriting Recognition (ICFHR)*. Crete, Greece, 2014.

[23] Sofiane Medjkoune, Harold Mouchère, Simon Petitrenaud, Christian Viard-Gaudin. "Using Online Handwriting and Audio Streams for Mathematical Expressions Recognition : a Bimodal Approach". In : *Document Recognition and Retrieval XX*. T. 8658. Burlingame, United States, fév. 2013, p. 865810-865810-11. DOI : 10.1007/978-3-642-39330-3_9.

[24] **Harold Mouchère**, Christian Viard-Gaudin, Richard Zanibbi, Utpal Garain, D. H. Kim, J. H. Kim. "ICDAR 2013 CROHME : Third International Competition on Recognition of Online Handwritten Mathematical Expressions". In : *International Conference on Document Analysis and Recognition (ICDAR)*. Washington, DC, USA, août 2013.

[25] Richard Zanibbi, Harold Mouchère, Christian Viard-Gaudin. "Evaluating Structural Pattern Recognition for Handwritten Math via Primitive Label Graphs". In : *Proc. SPIE, Document Recognition and Retrieval XX*. T. 8658. Burlingame, États-Unis, 2013, p. 865817-865817-11. DOI : 10.1117/12.2008409.

[26] Jinpeng Li, Harold Mouchère, Christian Viard-Gaudin. "Reducing AnnotationWorkload Using a Codebook Mapping and its Evaluation in On-Line Handwriting". In : *2012 International Conference on Frontiers in Handwriting Recognition*. Bari, Italy, sept. 2012, p. 1-6. URL : https://hal.archives-ouvertes.fr/hal-00717851.

[27] Jinpeng Li, Harold Mouchère, Christian Viard-Gaudin. "Quantifying spatial relations to discover handwritten graphical symbols". In : *Document Recognition and Retrieval XIX, Part of the IS&T/SPIE 24th Annual Symposium on Electronic Imaging*. San Francisco, United States, jan. 2012, p. –. URL : https://hal.archives-ouvertes.fr/hal-00672002.

[28] Sofiane Medjkoune, Harold Mouchère, Simon Petitrenaud, Christian Viard-Gaudin. "Using Speech for Handwritten Mathematical Expression Recognition Disambiguation". In : *2012 International Conference on Frontiers in Handwriting Recognition*. Bari, Italy, sept. 2012, p. 1-6. URL : https://hal.archives-ouvertes.fr/hal-00717855.

[29] **Harold Mouchère**, Christian Viard-Gaudin, Daehwan H. Kim, J. H. Kim, Garain Utpal. "ICFHR 2012 - Competition on Recognition of On-line Mathematical Expressions (CROHME 2012)". In : *Int. Conf. on Frontier in Handwriting Recognition (ICFHR)*. Bari, Italy, 2012.

[30] Jinpeng Li, Harold Mouchère, Christian Viard-Gaudin. "UNSUPERVISED HANDWRITTEN GRAPHICAL SYMBOL LEARNING Using Minimum Description Length Principle on Relational Graph". In : *International Conference on Knowledge Discovery and Information Retrieval, KDIR 2011*. Paris, France, oct. 2011. URL : https://hal.archives-ouvertes.fr/hal-00615217.

[31] Ahmad-Montaser Awal, Guihuan Feng, Harold Mouchère, Christian Viard-Gaudin. "First Experiments on a new Online Handwritten Flowchart Database". In : *Document Recognition and Retrieval XVIII*. San Fransisco, United States, jan. 2011, 7874-78740A. DOI : 10.1117/12.876624.

[32] Aurélie Lemaitre, Harold Mouchère, Jean Camillerapp, Bertrand B. Coüasnon. "Interest of Syntactic Knowledge for On-line Flowchart Recognition". In : *Graphics Recognition, GREC 2011*. North Korea, sept. 2011. URL : https://hal.archives-ouvertes.fr/hal-00635457.

[33] Jinpeng Li, Harold Mouchère, Christian Viard-Gaudin. "Symbol Knowledge Extraction from a Simple Graphical Language". In : *11th International Conference on Document Analysis and Recognition, ICDAR 2011*. Beijing, China, sept. 2011. URL : https://hal.archives-ouvertes.fr/hal-00615208.

[34] Solen Quiniou, Harold Mouchère, Sébastian Saldarriaga, Christian Viard-Gaudin, E. Morin, Simon Petitrenaud, Sofiane Medjkoune. "HAMEX - A Handwritten and Audio Dataset of Mathematical Expressions". In : *International Conference on Document Analysis and Recognition (ICDAR)*. 2011, p. 452-456.

[35] Sofiane Medjkoune, Harold Mouchère, Simon Petitrenaud, Christian Viard-Gaudin. "Handwritten and Audio Information Fusion for Mathematical Symbol Recognition". In : *11th International Conference on Document Analysis and Recognition, ICDAR 2011*. Beijing, China, sept. 2011. URL : https://hal.archives-ouvertes.fr/hal-00615206.

[36] **Harold Mouchère**, Christian Viard-Gaudin, Daehwan H. Kim, J. H. Kim, Garain Utpal. "CROHME2011 : Competition on Recognition of Online Handwritten Mathematical Expressions". In : *Proc. Int. Conference on Document Analysis and Recognition*. Beijing, China, 2011.

[37] Richard Zanibbi, Amit Pillay, **Harold Mouchère**, Christian Viard-Gaudin, Dorothea Blostein. "Stroke-Based Performance Metrics for Handwritten Mathematical Expressions". In : *Proc. Int. Conference on Document Analysis and Recognition*. Beijing, China, 2011.

[38] Ahmad-Montaser A.M. Awal, Harold Mouchère, Christian Viard-Gaudin. "Un classifieur hybride pour la reconnaissance d'expressions mathématiques manuscrites en-ligne". In : *17ème Congrès Reconnaissance des Formes et Intelligence Artificielle (RFIA)*. Caen, France, jan. 2010, p. 487-494. URL : https://hal.archives-ouvertes.fr/hal-00470592.

[39] Ahmad-Montaser A.M. Awal, Harold Mouchère, Christian Viard-Gaudin. "A Hybrid Classifier for Handwritten Mathematical Expression Recognition". In : *Electronic Imaging : Document Recognition and Retrieval XVI*. T. 7534. San José, United States, jan. 2010, p. 753410-753410. DOI : 10.1117/12.840023.

[40] Ahmad-Montaser Awal, Harold Mouchère, Christian Viard-Gaudin. "Improving online handwritten mathematical expressions recognition with contextual modeling". In : *International Conference on Frontiers in Handwriting Recognition*. India, nov. 2010, p. 427-432. DOI : 10.1109/ICFHR.2010.73.

[41] Ahmad-Montaser Awal, **Harold Mouchère**, Christian Viard-Gaudin. "The Problem of Handwritten Mathematical Expression Recognition Evaluation." In : *International Conference on Frontiers in Handwriting Recognition (ICFHR)*. Kolkata, India : IEEE Computer Society, 2010, p. 646-651.

[42] Hanyu Yan, Jin Lianwen, Christian Viard-Gaudin, **Harold Mouchère**. "SCUT-COUCH2009-TL : An Unconstrained Online Handwritten Chinese Text Lines Dataset". In : *Proceedings of the International Conference on Frontiers in Handwriting Recognition (ICFHR 2010)*. Inde, 2010.

[43] Ahmad-Montaser Awal, Harold Mouchère, Christian Viard-Gaudin. "Towards Handwritten Mathematical Expression Recognition". In : *ICDAR*. 2009, p. 1046-1050.

[44] Eric Anquetil, Abdullah Almaksour, Harold Mouchère. "Fast Online Incremental Learning with Few Examples For Online Handwritten Character Recognition". In : *Proceedings of the Eleventh International Conference on Frontiers in Handwriting Recognition (ICFHR'08)*. Montréal, Canada, août 2008, p. 623-628. URL : https://hal.archives-ouvertes.fr/hal-00463233.

[45] Fabien Lotte, **Harold Mouchère**, Anatole Lécuyer. "Pattern Rejection Strategies for the Design of Self-Paced EEG-based Brain-Computer Interfaces". In : *Proceedings of the 19th International Conference on Pattern Recognition*. 2008.

[46] Sabri Bayoudh, Laurent Miclet, **Harold Mouchère**, Eric Anquetil. "Learning a classifier with very few examples : knowledg based and analogy generation of new exemples for character recognition." In : *ECML*. 2007, p. 527-534.

[47] **Harold Mouchère**, Sabri Bayoudh, Eric Anquetil, Laurent Miclet. "Synthetic On-line Handwriting Generation by Distortions and Analogy". In : *13th Conference of the International Graphonomics Society (IGS)*. 2007, p. 10-13.

[48] **Harold Mouchère**, Éric Anquetil. "A Unified Strategy to Deal with Different Natures of Reject". In : *18th International Conference on Pattern Recognition (ICPR'06)*. 2006, p. 792-795.

[49] **Harold Mouchère**, Eric Anquetil. "Generalization Capacity of Handwritten Outlier Symbols Rejection with Neural Network". In : *Proceedings of the 10th International Workshop on Frontier in Handwriting Recognition (IWFHR'06), to be published*. La Baule, France, oct. 2006, p. 187-192. URL : http://hal.inria.fr/inria-00104310.

[50] **Harold Mouchère**, Éric Anquetil, Nicolas Ragot. "Writer Style Adaptation of On-line Handwriting Recognizers : A Fuzzy Mechanism Approach". In : *Proceedings of the 12th Conference of the International Graphonomics Society (IGS)*. Sous la dir. d'A. Marcelli et C. De Stefano. Salerno, Italy, juin 2005, p. 193-197.

[51] **Harold Mouchère**, Éric Anquetil, Nicolas Ragot. "On-line Writer Adaptation for Handwriting Recognition using Fuzzy Inference Systems". In : *Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR)*. Sous la dir. de Bob Werner. T. 2. Seoul, Korea : IEEE Computer Society, août 2005, p. 1075-1079.

Conférences francophones avec comité de relecture

[52] Ting Zhang, **Harold Mouchère**, Viard-Gaudin Christian. "Using BLSTM for Interpretation of 2D Languages - Case of Handwritten Mathematical Expressions." In : *CORIA-CIFED*. 2016, p. 217-232.

[53] Frank Julca-Aguilar, Christian Viard-Gaudin, Harold Mouchère, Sofiane Medjkoune, Nina S. T. Hirata. "Integration of Shape Context and Neural Networks for Symbol Recognition". In : *Colloque International Francophone sur l'Écrit et le Document (CIFED2014)*. Nancy, France, mar. 2014. URL : https://hal.archives-ouvertes.fr/hal-01150830.

[54] Jinpeng Li, Harold Mouchère, Christian Viard-Gaudin. "Une distance entre deux ensembles de séquences avec la contrainte de continuité". In : *Colloque International Francophone sur l'Écrit et le Document (CIFED2012)*. Bordeaux, France, mar. 2012, p. –. URL : https://hal.archives-ouvertes.fr/hal-00671998.

[55] Sofiane Medjkoune, Harold Mouchère, Simon Petitrenaud, Christian Viard-Gaudin. "Vers l'Alignement des Signaux Écrit et Sonore". In : *Colloque International Francophone sur l'Écrit et le Document (CIFED2014)*. Nancy, France, mar. 2014, p. 341-356. URL : https://hal.archives-ouvertes.fr/hal-01151106.

[56] Sofiane Medjkoune, Harold Mouchère, Simon Petitrenaud, Christian Viard-Gaudin. "Fusion d'Informations Bi-modales pour la Reconnaissance d'Expressions Mathématiques Cas des modalités audio et écriture manuscrite en ligne". In : *Colloque International Francophone sur l'Écrit et le Document (CIFED2012)*. Bordeaux, France, mar. 2012. URL : https://hal.archives-ouvertes.fr/hal-00672003.

[57] Ahmad-Montaser A.M. Awal, Harold Mouchère, Christian Viard-Gaudin. "Apprentissage de relations spatiales pour la reconnaissance d'expressions mathématiques manuscrites en-ligne". In : *Colloque International Francophone sur l'Écrit et le Document (CIFED2010)*. France, mar. 2010. URL : https://hal.archives-ouvertes.fr/hal-00490999.

[58] Adrien Delaye, Harold Mouchère. "Vers une approche générique pour la reconnaissance de formes manuscrites structurées : Application aux équations mathématiques et aux caractères chinois". In : *Colloque International Francophone sur l'Ecrit et le Document (CIFED2010)*. Sousse, Tunisia, mar. 2010, p. -. URL : https://hal.archives-ouvertes.fr/hal-00491000.

[59] Abdullah Almaksour, Harold Mouchère, Eric Anquetil. "Apprentissage incrémental et synthèse de données pour la reconnaissance de caractères manuscrits en-ligne". In : *Colloque International Francophone sur l'Écrit et le Document (CIFED2008)*. Sous la dir. d'Antoine Tabbone et Thierry Paquet. France : Groupe de Recherche en Communication Ecrite, oct. 2008, p. 55-60. URL : https://hal.archives-ouvertes.fr/hal-00335040.

[60] **Harold Mouchère**, Eric Anquetil. "Synthèse de caractères manuscrits en-ligne pour la reconnaissance de l'écriture". In : *Colloque International Francophone sur l'Écrit et le Document (CIFED2006)*. 2006, p. 187-192. URL : https://hal.archives-ouvertes.fr/hal-00113590.

[61] **Harold Mouchère**, Éric Anquetil, Nicolas Ragot. "Etude et gestion des types de rejet pour l'optimisation de classifieurs". In : *RFIA*. 2006.

[62] **Harold Mouchère**, Éric Anquetil, Nicolas Ragot. "Étude des mécanismes d'adaptation pour l'optimisation de classifieurs flous dans le cadre de la reconnaissance d'écriture manuscrite". In : *12es rencontres francophones sur la Logique Floue et ses Applications (LFA'04)*. accepted, Nantes, France, nov. 2004.

Chapitre 2

Introduction

Analyser, reconnaître, comprendre des documents structurés, essentiellement manuscrits, produits par des humains et à destination d'humains, constituent une tâche fort complexe lorsque l'on cherche à la résoudre par des moyens informatiques. Cette préoccupation constitue le centre de mes activités de recherche. Comme nous le verrons dans ce document, elle mobilise des connaissances et des compétences à la croisée de nombreux domaines scientifiques : du traitement d'images à la reconnaissance des formes, des algorithmes d'apprentissage automatique aux modèles de langages, des manipulations de graphes aux algorithmes d'optimisation, de l'évaluation de performances à l'interaction homme-machine. Les documents traités sont toujours "visibles" même si ce ne sont pas toujours des images et ces documents sont toujours porteurs d'un sens codé par les Hommes dans un langage graphique spécifique [63]. Ce langage graphique peut être composé de symboles (lettres, chiffres, ...), de mots, de phrases mais aussi de figures, géométriques ou non, mais ce qui rend ce langage "graphique" c'est que la reconnaissance de l'organisation en deux dimensions de ces éléments est aussi importante que la détection de ces éléments. Prenons un simple courrier comme il en est maintenant traité par millions dans les centres de numérisation et de traitement de documents. L'expéditeur, le destinataire, l'objet, le corps du message et la signature sont organisés dans la page en suivant une grammaire simple plus ou moins normalisée. Si les machines sont maintenant capables de traiter les documents imprimés standards (factures, devis, contrats, bulletins de paie...) le défi reste encore ouvert pour ce qui est des documents manuscrits. L'écriture manuscrite est par nature variable, dépendante du scripteur, du contexte et donc difficile à reconnaître même sur ses éléments de base (caractères, symboles, formes). Une fois ces symboles reconnus il faut encore les structurer pour les interpréter. Historiquement, le domaine s'est largement inspiré des systèmes de reconnaissance de la parole. En effet la reconnaissance d'une phrase, qu'elle soit écrite ou parlée peut être modélisée de la même

manière : un signal 1D temporel utilisant un vocabulaire et une langue spécifique. Par exemple, les systèmes à base de Chaine de Markov Cachée (HMM) couplés à un modèle de langage type n-gram sont très utilisés dans les deux domaines [64, 65]. La reconnaissance du texte manuscrit est maintenant une réalité après près de 30 ans de recherche [66]. D'abord sur des documents très normalisés et avec un vocabulaire limité (adresses postales, chèques) et maintenant sur du texte libre. Par exemple, il est aujourd'hui possible de saisir de façon assez fiable du texte manuscrit directement sur son téléphone ou sur les ordinateurs portables munis d'un stylet. Par contre reconnaître un courrier manuscrit complet, dessiner un diagramme ou une équation, traiter des millions de pages manuscrites de documents anciens, ... restent des tâches difficiles à cause de la seconde dimension des documents graphiques.

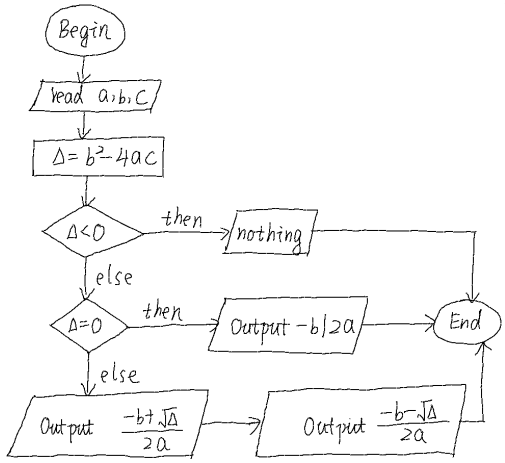
Concevoir et développer un système de reconnaissance de documents manuscrits nécessite donc une expertise large et profonde. Les grandes lignes d'un système de reconnaissance de documents manuscrits structurés vont être présentées dans cette introduction afin de positionner mes contributions au fil des nombreuses étapes nécessaires.

Les références citées ici ne cherchent pas à faire un état de l'art de la reconnaissance de documents manuscrits mais à situer mes travaux parmi les travaux existants.

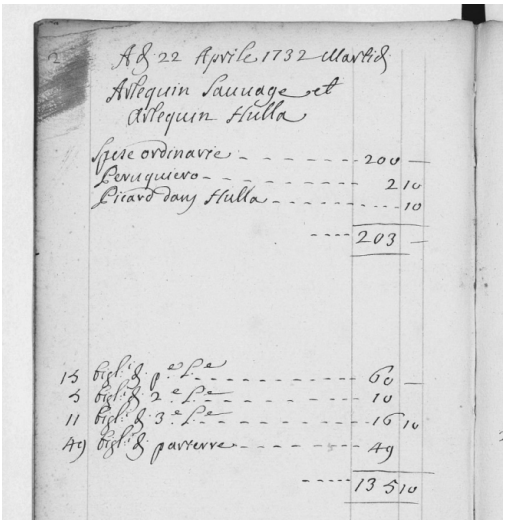
2.1 De la reconnaissance des documents structurés manuscrits

Il existe un grand nombre de types de documents structurés manuscrits. La figure 2.1 montre les principaux types de documents sur lesquels j'ai travaillé. Les diagrammes comme dans la figure (a) ont une grammaire très contrainte avec peu de symboles mais la disposition des éléments et du texte est complètement libre. Au contraire, les équations mathématiques (c) ont une grammaire beaucoup plus complexe, un ensemble de symboles très grand mais une disposition assez stable. Les documents anciens de la collection CIRESFI (b) ont une mise en page assez régulière avec un vocabulaire limité mais ils sont très difficiles à lire même par un humain. Enfin les gestes multipoints d'interaction (d) sont courts et simples mais les formes et dispositions ne sont pas très stables et la synchronisation des sous-parties doit être prise en compte.

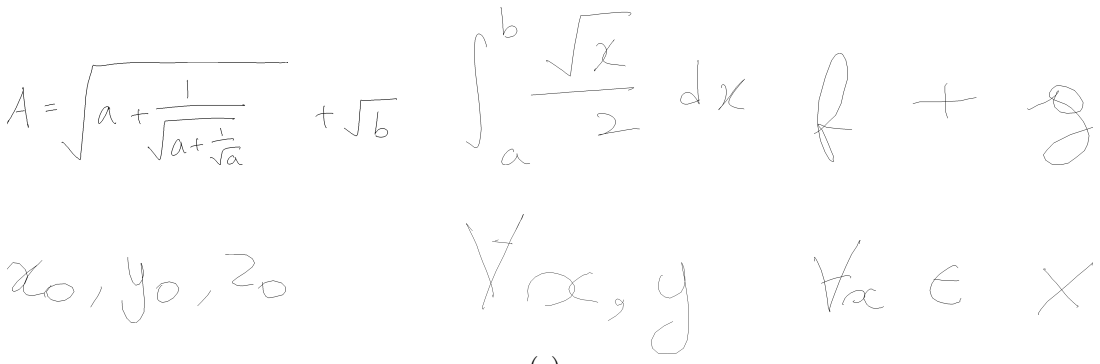
Pour illustrer mes propos je m'appuierai sur l'exemple de la reconnaissance d'expressions mathématiques qui est le domaine où j'ai le plus contribué.



(a)



(b)



(c)

(d)

FIGURE 2.1 – Exemples de documents structurés traités dans mes travaux. (a) un diagramme, extrait de la base FC [31], (b) un document ancien, extrait du projet CIREFI, (c) des équations, extraites de la base CROHME [1], (d) un geste multipoint, extrait de [19].

Nous pouvons distinguer deux grandes familles de documents en fonction du type de signal utilisé : les documents dits *en-ligne* et ceux dits *hors-ligne*. Ainsi que le montre la figure 2.2 le signal en-ligne est acquis grâce à un périphérique spécifique capturant la dynamique du tracé. Nous obtenons une séquence de points organisés en traces qui sont délimitées par un poser et un lever de crayon (ou de doigt). Les gestes multipoints présentés dans la figure 2.1(d) sont *en-ligne* mais plusieurs traces peuvent être dessinées en même temps. Nous pouvons distinguer deux types d'interactions avec les documents en-ligne. La reconnaissance *a posteriori* est effectuée après la saisie complète du document et peut donc utiliser le contexte de l'ensemble de celui-ci pour guider la reconnaissance, comme nous le faisons pour les expressions mathématiques ou les diagrammes. L'autre modalité consiste à reconnaître les symboles *à la volée* c'est-à-dire au fur et à mesure de leur saisie sur l'interface, comme c'est le cas des gestes multipoints. De façon différente, un document *hors-ligne* est une image constituée de pixels. Une difficulté supplémentaire est ajoutée car il faut séparer le fond de la forme à reconnaître (l'écriture). Les documents en-ligne peuvent être convertis en images et traités ensuite par les mêmes processus, mais l'inverse a aussi été expérimenté [67].

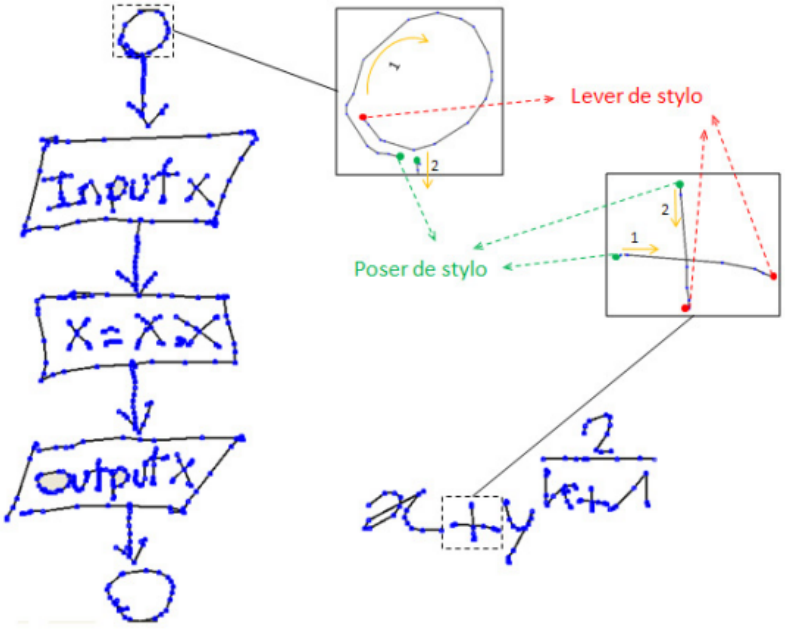


FIGURE 2.2 – L'écriture en ligne est une séquence de traces constituées de points. Extrait de la Thèse de M. Awal(Fig 4).

La figure 2.3 présente les grandes étapes d'un système de reconnaissance de documents structurés. Il s'agit d'un schéma de principe ayant pour but de situer

les différentes contributions de mes travaux et thèses encadrées. En fonction du type de documents (en-ligne, hors-ligne, multipoints), des applications visées et des choix de solutions certaines étapes peuvent être supprimées ou ajoutées.

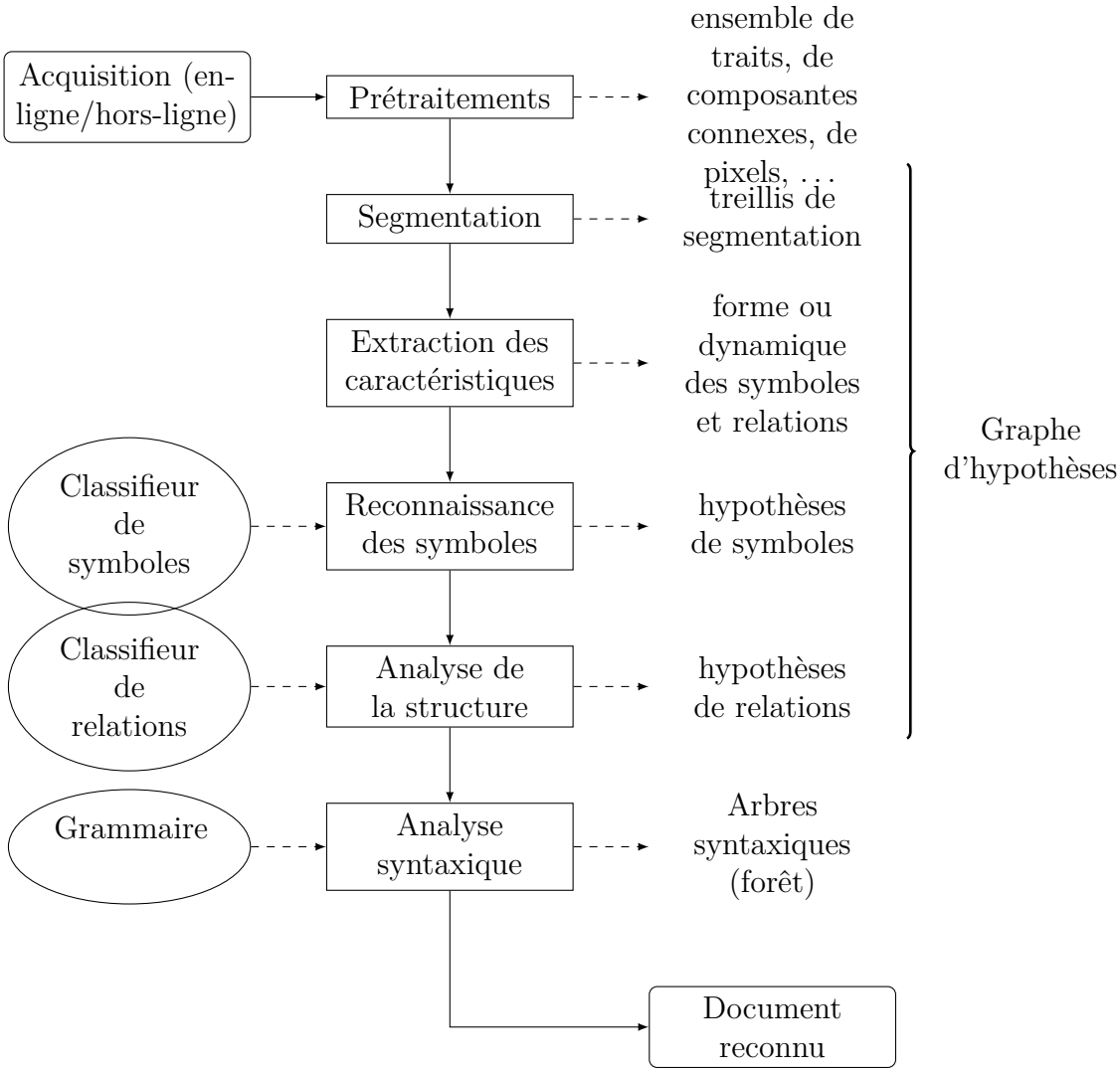


FIGURE 2.3 – Schéma général de la reconnaissance de documents structurés.

Les pré-traitements permettent de réduire le bruit d'acquisition mais surtout la variabilité du signal d'entrée. Si les pré-traitements des images sont bien connus (filtrage, binarisation, ouverture/fermeture, ...) ceux sur le signal en-ligne sont plus spécifiques. Par exemple la plupart du temps il faut normaliser la dimension du tracé mais il est souvent difficile de connaître l'échelle d'un document : le périphérique d'acquisition peut changer, le scripteur peut écrire gros ou petit... Le signal

est aussi ré-échantillonné pour diminuer l'impact de la vitesse du crayon, mais cette étape nécessite aussi de connaitre l'échelle du document. Ces pré-traitements cherchent à réduire la variabilité mais en conservant les points singuliers (points isolés, points de rebroussement...). Ces différentes possibilités ont été comparées dans la Thèse de M. Awal, mais chaque application a ses adaptations.

La segmentation consiste à découper le document en éléments atomiques qui ne peuvent appartenir qu'à un seul symbole à la fois, puis d'énumérer tous les regroupements possibles pour former les symboles du document. Il s'agit de l'étape avec la plus forte complexité dans un système de reconnaissance complet, même pour des expressions mathématiques imprimées [68] où les dispositions et formes sont plus régulières qu'en manuscrit. Plus formellement, si S est l'ensemble de taille n de ces éléments de base, une segmentation en est une partition. Le nombre de partitions d'un ensemble est donné par le nombre de Bell :

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k \text{ avec } B_0 = B_1 = 1. \tag{2.1}$$

Généralement, toutes les segmentations ne sont pas énumérées mais un treillis de segmentation est construit en énumérant les hypothèses de segments (les futurs symboles). Le nombre d'hypothèses est encore de $2^n - 1$. C'est pourquoi des critères de segmentation sont définis pour limiter la combinatoire. Pour reconnaître la parole ou une ligne d'écriture les éléments sont organisés en une seule dimension et les hypothèses de symboles sont construites en agrégeant les éléments selon une relation d'ordre (temporelle ou spatiale) [69]. Si la taille maximum d'un symbole est m, le nombre maximum d'hypothèses de symboles est $n \times m$. En deux dimensions, les éléments peuvent théoriquement se combiner avec tous les autres, mais pour la plupart des applications, des contraintes spatiales et/ou temporelles sont ajoutées pour agréger les éléments proches. Pour pallier ce problème certains travaux sur les expressions en-ligne [70, 71] utilisent le temps comme dimension de segmentation, mais ils doivent gérer la combinatoire de l'ordre variable des traits et la présence de symboles complétés après d'autres symboles. Une autre solution consiste à entraîner un classifieur de segmentation pour décider à chaque point de segmentation s'il faut ou non créer un nouveau symbole [72].

Les segments générés sont ensuite reconnus par des classifieurs. Le plus simple est de considérer ce problème comme un problème de reconnaissance de symboles isolés. Il s'agit donc de choisir les caractéristiques et le type de classifieur. Pour les symboles mathématiques en ligne, nous comparons dans [1] les différentes approches proposées par les participants à la compétition CROHME. Dans la Thèse de M. Awal nous avons aussi étudié plusieurs combinaisons possibles. Cette étape permet d'étiqueter le graphe d'hypothèses de segments avec une ou plusieurs classes

possibles par segment. Nous obtenons alors un graphe d'hypothèses de symboles. Dans la continuité de mes travaux de thèse, nous avons proposé dans la Thèse de M. Awal de permettre au classifieur de rejeter les hypothèses de mauvaises segmentations ce qui permet de supprimer une partie des hypothèses et de réduire la complexité. Pour réduire la complexité certains systèmes choisissent de ne conserver que la meilleure réponse du classifieur.

Une fois les hypothèses de symboles faites, il faut évaluer leur potentielles relations spatiales. Les premières approches consistaient à évaluer un ensemble de règles basées sur quelques caractéristiques géométriques mesurées entre les boîtes englobantes des symboles. Que ce soit pour les documents en-ligne ou hors-ligne, les relations entre éléments manuscrits (symboles, lettres, mots ou lignes) ne sont pas toujours stables. Les systèmes les plus récents prennent en compte le contexte des symboles, leurs classes respectives voir leur formes pour alimenter des classifieurs de relations spatiales. Dans la Thèse de M. Awal nous avons été les premiers à proposer un apprentissage statistique des relations spatiales. Nous avons ensuite amélioré cette approche dans la Thèse de F. D. Julca-Aguilar en prenant en compte le contexte de l'expression pour aider à la reconnaissance. Comme pour les symboles, une ou plusieurs relations spatiales peuvent être conservées.

Une fois les symboles et leurs relations reconnus, le graphe d'hypothèses contient toutes les hypothèses de symboles et toutes les hypothèses de relations. Ce graphe n'est pas toujours construit de façon explicite, il peut être parcouru au fur et à mesure de l'étape suivante d'analyse syntaxique [73]. Gérer la complexité de ce graphe d'hypothèses est une des problématiques qu'abordent les thèses de M. Awal, de S. Medjkoune, de J. Li et de F. Aguilar. La figure 2.4 présente un exemple de graphe d'hypothèses de symboles et de relations. On peut voir que toutes les hypothèses de symboles et de relations n'ont pas été conservées mais qu'il reste plusieurs interprétations possibles.

La dernière étape consiste en l'analyse syntaxique, c'est-à-dire à choisir parmi les hypothèses construites précédemment la meilleure combinaison par rapport au langage utilisé. L'analyse peut être très simple, par exemple en vérifiant localement la cohérence entre les étiquettes de symboles et les relations. La plupart des approches sont basées sur des grammaires qui permettent de prendre en compte le contexte global du document, par exemple en vérifiant dans une expression mathématique la cohérence des parenthèses ouvrantes et fermantes. Certaines approches (gloutonnes) ne construisent qu'un seul arbre syntaxique en commençant par les symboles structurants (e.g. les barres de fractions, les grands opérateurs ou simplement le symbole le plus à gauche). Les approches les plus complètes évaluent un ensemble de solutions en construisant une forêt d'arbres d'analyse. D'abord il faut définir une grammaire (hors contexte, floue [74] ou probabiliste [73]) puis un algorithme d'analyse (CYK descendant pour [73] ou l'algorithme ascendant d'Unger

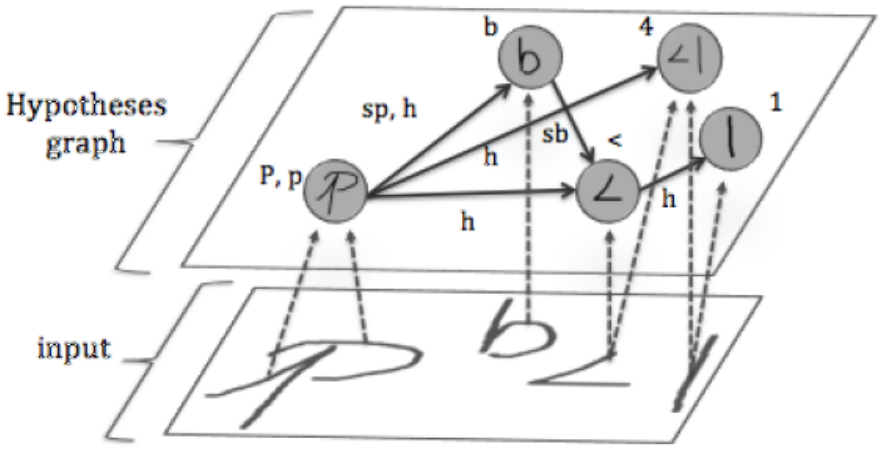


FIGURE 2.4 – Exemple de graphe d'hypothèses. Extrait de Thèse de F. D. Julca-Aguilar (Fig 3.9).

pour [74]). La solution finale est l'arbre expliquant complètement le document et avec la plus forte probabilité. Nous pouvons remarquer que dans ces deux exemples [74, 73], les relations spatiales ne sont pas évaluées avant l'analyse syntaxique mais pendant l'application de chacune des règles.

La présentation séquentielle du processus montre bien une des difficultés du problème : une erreur commise en début de processus ne peut pas être rattrapée en fin d'analyse. Certains systèmes subissent vraiment cette difficulté en n'analysant qu'une seule segmentation ou en ne conservant qu'une seule étiquette de symbole pour chaque segment. A l'inverse, nous avons privilégié les approches permettant d'optimiser une fonction de coût globale. Dans ces systèmes les décisions sont prises le plus tard possible et le critère de sélection prend en compte l'évaluation de tous les paramètres (segmentation, probabilité des étiquettes de symboles et de relations, règles de grammaire utilisées).

Il existe de nombreux types de documents structurés et suivant les objectifs les approches peuvent être très différentes de celles présentées ici dans le cadre des expressions mathématiques mais il y a souvent des points communs. Citons par exemple la reconnaissance de tables dans les documents hors-ligne [75], bien que le contenu des tables ne soit pas forcément utilisé pour la reconnaissance, on retrouve les mêmes étapes : pré-traitements, détection des éléments de base (lignes, alignements, mots,...) puis une étape structurelle voire grammaticale.

Deux aspects du problème ne sont pas discutés ici mais seront développés dans le reste du manuscrit. Tout d'abord l'apprentissage des différents classifieurs et paramètres des systèmes : les différents classifieurs sont interdépendants, comment

optimiser globalement le système ? Ensuite l'ambiguïté naturelle de l'écriture manuscrite : faut-il toujours faire confiance au classifieur ? Quand peut-on prendre la meilleure décision ? Enfin l'évaluation de ces systèmes de reconnaissance : il y a de multiples sources d'erreurs dans ces systèmes complexes, comment évaluer finement les différentes parties ?

2.2 Principales contributions et plan du manuscrit

Cette section présente mes principales contributions au sein de l'équipe IVC depuis ma soutenance de thèse en 2007. Ce travail a été fait en constante collaboration avec mon collègue Christian Viard-Gaudin.

2.2.1 La reconnaissance d'expressions mathématiques manuscrites

Une grande partie de mes travaux ont porté sur ce thème présenté dans le Chapitre 3.

Lors de mon arrivée en 2008 j'ai été intégré directement dans le projet CIEL et plus particulièrement dans l'encadrement de la Thèse de M. Awal portant sur la reconnaissance des expressions mathématiques en-ligne présentée dans la section 3.1. Cette thèse a permis la construction d'un des premiers systèmes holistiques prenant en compte toutes les informations de la reconnaissance [2]. Ce système est basé sur plusieurs contributions originales :

— dans la continuité de mes travaux de thèse nous avons doté le classifieur d'hypothèses de symboles de la capacité de rejeter les mauvaises segmentations,

— nous avons introduit l'apprentissage global permettant de combiner segmentation et reconnaissance grâce à un apprentissage embarqué dans le contexte de l'expression,

— l'apprentissage automatique du classifieur de relations spatiales.

La section 3.2 présente le projet régional DEPART issu de notre collaboration avec les équipes Traitement Automatique du Langage Naturel (TALN) du LINA et l'équipe Traduction et Parole du LIUM. Notre participation portait sur la reconnaissance d'expressions mathématiques multi-modales (écrit-parole). La Thèse de S. Medjkoune a montré la complémentarité des deux modalités et les différentes solutions de fusion de l'information. Dans ce cadre nous avons constitué la base d'expressions multimodales HAMEX. Il s'agit là d'un travail très novateur sur de

l'interaction multimodale. Gageons que la voie explorée ici ouvre d'intéressantes perspectives.

La section 3.3 présente la collaboration avec l'université de Sao Paulo qui s'est concrétisée par la thèse en co-tutuelle de F. Aguilar. Deux contributions intéressantes ont été proposées. La première porte sur l'utilisation des *shape contexts* comme caractéristiques pour la classification des symboles isolés et des relations spatiales. Ils permettent notamment de prendre en compte le contexte du symbole pour sa reconnaissance. La seconde contribution porte sur l'utilisation d'une grammaire de graphes pour l'analyse syntaxique. Comparée aux approches à grammaire de graphes existantes [76, 77], notre solution est entièrement basée sur des classifieurs entrainables (symboles et relations) ce qui permet de bien séparer la grammaire de l'analyse de la structure. La complexité de l'algorithme est réduite grâce aux options de rejet qui simplifient le graphe d'hypothèses de symboles et de relations.

La section 3.4 présente les travaux de Thèse de T. Zhang (toujours en cours) portant sur l'utilisation des BLSTM pour la reconnaissance des expressions mathématiques. Ces réseaux de neurones sont réputés pour leur capacité à reconnaître les séquences [78](parole, texte, ...) notamment en intégrant implicitement le modèle de langage mais ils ne sont pas naturellement adaptés à la reconnaissance de structures. Nous avons proposé de décomposer le problème en un ensemble de classifications de séquences qui sont ensuite recombinées pour former un graphe relationnel. C'est la première fois que les BLSTM sont utilisés pour reconnaître en même temps les symboles de l'expression mais aussi leurs relations spatiales. Le système final obtient des résultats comparables à l'état de l'art mais sans utiliser d'étape d'analyse syntaxique.

2.2.2 Les compétitions CROHME

Il s'agit là d'une contribution majeure à un domaine de recherche pour lequel j'ai réussi à créer une dynamique en cristallisant des énergies autour de cette thématique. Le rayonnement est international et s'inscrit dans la durée.

Avant l'organisation de ces compétitions, le domaine souffrait de l'utilisation d'une diversité de bases d'évaluation et de différentes métriques de mesure des performances. Le chapitre 4 présente mes contributions qui couvrent tout aussi bien des aspects scientifiques (les nouvelles métriques), organisationnels, logiciels et de promotion.

La section 4.1 rappelle un bref historique de l'organisation de la compétition : les différentes tâches, les participants...

La section 4.2 explique comment nous avons construit les bases d'expressions mathématiques manuscrites utilisées. Jusque là les bases existantes utilisaient des expressions choisies manuellement ou générées aléatoirement. Nous avons choisi une solution plus proche de ce qui se fait en reconnaissance de textes ou de la parole, c'est-à-dire prendre des données qui correspondent à un usage avéré.

La section 4.3 présente une contribution essentielle à l'organisation d'une compétition : les métriques utilisées. Nous avons choisi de modéliser les expressions par des graphes de primitives et défini des métriques basées sur la comparaison du graphe de vérité avec le graphe obtenu en sortie de la reconnaissance. Nos métriques permettent d'évaluer les différents niveaux d'erreurs possibles agrégés sur une base de test complète : étiquettes et relations des primitives, segmentation et reconnaissance des symboles et de leurs relations, reconnaissance petites structures locales (type bigram), reconnaissance des expressions complètes.

2.2.3 L'analyse de structures par les graphes

Les graphes sont des outils puissants et complémentaires aux méthodes purement statistiques pour analyser des documents structurés. Nous avons proposé des solutions originales mixant ces approches dans des travaux de deux doctorants que nous présentons dans le chapitre 5.

Nos travaux sur la reconnaissance de diagrammes manuscrits en-ligne (plus précisément les *flowcharts* ou *organigrammes de programmation*), présentés dans la section 5.1, ont d'abord illustré la généricité de nos approches de reconnaissance d'expressions manuscrites. Une différence majeure dans le langage graphique fait que nos approches ont vite été limitées : ce type de diagramme ne peut pas être représenté par un arbre à cause des boucles possibles. Après des pré-études sur les symboles isolés, cette base a été le support de plusieurs collaborations avec l'équipe IntuiDoc sur l'utilisation de leur outil d'analyse DMOS. Durant un stage de Master, nous avons proposé d'utiliser les MRF (Markov Random Field) pour étiqueter les traits et leurs relations, puis d'alimenter l'analyse syntaxique de DMOS avec ce pré-étiquetage. C'est la première fois que les MRF ont été utilisés pour l'étiquetage de relation. Ces travaux ont été soumis en 2016 à la revue IJDAR (relecture en cours).

Une des difficultés avec l'analyse de documents structurés consiste à re-définir pour chaque nouveau type de documents l'ensemble des symboles, des relations spatiales et la grammaire associée. Serait-il possible d'extraire automatiquement ces connaissances à partir d'exemples non annotés ? Sans résoudre complètement ce problème ambitieux, la section 5.2 présente les travaux de la Thèse de J. Li portant sur l'extraction semi-automatique de connaissances dans les langages structurés.

En modélisant les documents comme un graphe d'hypothèses de symboles, un algorithme non supervisé extrait les structures fréquentes qui correspondent à des propositions de symboles ou de structures locales.

La section 5.3 décrit une collaboration en cours avec l'équipe IntuitDoc (Irisa) au travers de la Thèse de Z. Chen (toujours en cours) : la reconnaissance de gestes tactiles multipoints. Ces gestes sont une production manuelle où l'analyse de la structure spatiale des éléments ainsi que de leur synchronisation est indispensable à leur reconnaissance. Les gestes sont décomposés en primitives et leur relations spatiales et temporelles sont analysées pour construire un graphe représentatif. Pour la reconnaissance nous utilisons une technique originale de *graph embedding* utilisant les distances d'édition entre un graphe à reconnaître avec différents graphes prototypes pour alimenter un classifieur SVM. Au delà de la reconnaissance de ces gestes isolés, nous avons proposé des solutions pour la reconnaissance précoce des gestes de commande directe (avant la fin du geste) et pour la gestion de plusieurs utilisateurs simultanément en interaction. Ces approches sont rendues possible grâce à l'utilisation d'options de rejet sur les classifieurs de gestes. Une somme de problèmes nouveaux est étudiée à travers ce travail. Ainsi, pour la première fois, de façon croisée, les interactions multistrokes, multipoints et multiutilisateurs sont considérées. Ces recherches ouvrent des perspectives d'applications pour des mises en œuvre sur les surfaces tactiles de grandes tailles qui commencent à se démocratiser.

2.2.4 Perspectives et Projets de recherche

Dans le chapitre 6 je présente les perspectives à court et moyen termes de mes travaux de recherche. Dans la continuité de mes travaux sur l'analyse de documents manuscrits, je compte renforcer les collaborations avec les domaines connexes soit parce qu'ils sont complémentaires (TALN, IHM) soit parce que les outils utilisés sont proches (vision d'images industrielles ou de scènes naturelles). Je compte aussi développer deux axes transversaux. Le premier concerne l'utilisation des nouvelles avancées sur les réseaux de neurones (profonds ou récurrents) et leurs applications à l'analyse de documents. Le second concerne l'utilisation des différents niveaux de contexte pour guider la reconnaissance d'un document complexe.

Chapitre 3

La reconnaissance d'expressions mathématiques manuscrites

Durant mon doctorat, je me suis focalisé sur la reconnaissance de caractères manuscrits (en-ligne) grâce à des réseaux de neurones (MLP, RBFN) et système d'inférence floue suivant deux axes : la capacité de rejet des classifieurs [48, 61, 49] et l'adaptation au scripteur [6, 44]. Ces deux compétences sur l'apprentissage artificiel ont largement facilité mon intégration dans le projet CIEL puis ont influencé le reste de mes travaux.

3.1 Projet CIEL - Thèse de M. Awal

Le projet CIEL (pour "Conversion Indexation de l'Écriture en Ligne") avait pour objectif d'étudier différents aspects de l'analyse de l'écriture en-ligne qui étaient étudiés depuis longtemps dans le domaine hors-ligne : recherche d'information, indexation, identification du scripteur, adaptation au scripteur, prise en compte de la composante spatiale dans l'analyse des documents. C'est dans ce dernier axe que s'inscrit la reconnaissance des équations mathématiques manuscrites et la thèse de M. Awal. Ces travaux ont été réalisés en collaboration étroite avec l'entreprise partenaire du projet *Vision Objects* (maintenant renommée *MyScript*).

3.1.1 Les données

Cette première thèse s'est trouvée confrontée au problème du manque de données disponibles pour réaliser un apprentissage. Une collecte importante de symboles isolés (plus de 200 classes) a été réalisée en début de projet. Un système de

génération d'expressions manuscrites artificielles a ensuite été utilisé pour créer à partir des symboles isolés et d'une chaine LaTeX une expression manuscrite en 2 dimensions [79]. Cette approche a permis de démarrer la recherche dans le domaine mais nous avons vite vu l'intérêt d'utiliser des expressions réelles : bien qu'un bruit Gaussien soit utilisé pour introduire de la variabilité dans les données, notre modèle de relations spatiales apprenait des relations synthétiques et non des relations réelles. Nous avons donc ensuite fait les premières campagnes de collecte d'expressions manuscrites d'abord en utilisant des corpus d'expressions existants [80] (Garain et Aster) sur des petits nombres de scripteurs puis en sélectionnant des expressions tirées de Wikipedia et saisies par des centaines de scripteurs. Il en résultera une base de 5820 expressions manuscrites triées en différents niveaux de difficultés (du mode 'calculette' à des expressions complexes). Cette base ayant été collectée dans le cadre du projet CIEL avec l'entreprise Vision Objects il n'était pas possible de rendre publique ces expressions, mais c'est ce protocole qui a inspiré les collectes réalisées dans le cadre de la compétition CROHME (Chapitre 4).

L'approche présentée ci dessous a été publiée dans la revue Partern Recognition Letter (PRL) en 2014 [2], l'article est disponible dans la sélection d'articles page 80.

3.1.2 Le graphe d'hypothèses, le classifieur et le rejet

Le système proposé se base globalement sur l'architecture présentée en introduction. La première étape consiste à générer l'ensemble des hypothèses de symboles. Pour limiter la complexité, deux contraintes sur la séquence de traits sont ajoutées : le nombre maximum de traits par symbole (4 traits suffisent à couvrir quasiment tous les exemples d'apprentissage) et le nombre de sauts temporels dans la séquence. Généralement les symboles mathématiques sont écrits avec des traits consécutifs dans le temps, mais parfois le scripteur revient en arrière pour compléter un symbole, c'est le "saut temporel". L'avantage de cette approche est qu'elle couvre bien les symboles possibles et n'a pas besoin d'analyser les propriétés géométriques de l'encre.

Avant d'alimenter l'analyse syntaxique avec ces hypothèses de symboles, nous nous sommes intéressés à trouver la meilleure segmentation du tracé en symboles valides. Il s'agit d'un problème classique de couverture exacte : tous les traits doivent appartenir à un et un seul symbole valide. Pour vérifier la validité des hypothèses de symboles, nous utilisons un classifieur de symboles mais le classifieur est confronté à un grand nombre d'hypothèses correspondant à une mauvaise segmentation. L'idée est de permettre au classifieur de reconnaître ces mauvaises segmentations pour ne pas les conserver ensuite. Comme détaillé dans [38, 39] nous ajoutons une classe "junk" pour rejeter les mauvaises segmentations. Cette classe peut être apprise à partir d'un grand nombre d'exemples générés aléatoirement

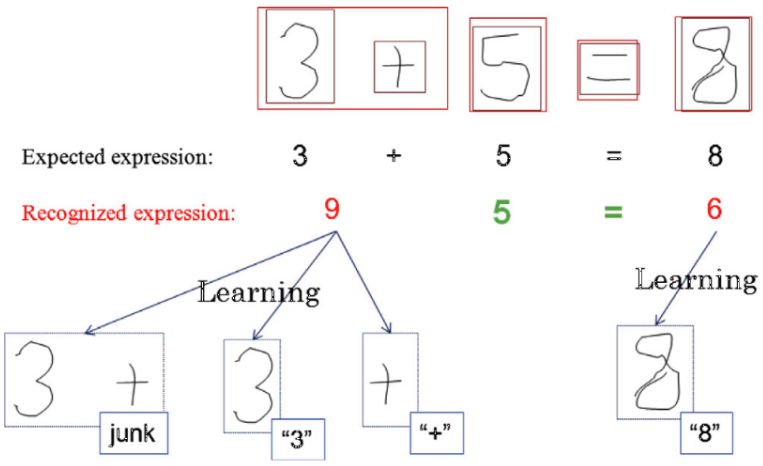


FIGURE 3.1 – Exemple d'apprentissage global pour l'expression $3+5=8$. Extrait de [2] (Fig. 4).

(comme proposé par la suite dans CROHME) mais nous avons préféré une approche permettant d'utiliser seulement les exemples provenant véritablement des échantillons de segments produits par le système en cours d'apprentissage. Nous appelons cela l'**apprentissage global** [2] car il faut une expression complète pour générer les exemples de mauvaises segmentations.

L'idée de l'apprentissage global est de focaliser l'apprentissage sur les exemples les plus utiles à apprendre. Alors que l'ensemble des symboles à reconnaître est bien défini, l'ensemble des mauvaises segmentations est plus vaste et par définition varié. Nous utilisons un classifieur qui peut être appris de façon itérative (comme un réseau de neurones par descente de gradient). Le problème de segmentation est résolu en utilisant l'état courant de ce réseau et les erreurs (symboles ou mauvaises segmentations) sont utilisées pour continuer l'apprentissage. La figure 3.1 donne un exemple où deux erreurs sont commises par le classifieur, une mauvaise segmentation (fusion de 2 symboles) et une mauvaise reconnaissance ; il y a donc 4 exemples à réapprendre. Cette approche peut être considérée comme du boosting car les exemples mal reconnus ont finalement un poids plus fort dans l'apprentissage.

Plusieurs configurations de réseaux de neurones et ensembles de caractéristiques ont été testés en se basant sur les travaux précédents de l'équipe [81]. Le classifieur principalement utilisé est un TDNN (Time Delay Neural Network) avec 7 caractéristiques extraites pour chaque point du signal, deux couches de convolution, puis un perceptron multicouche (MLP en anglais) composé d'une couche cachée et une couche de sortie. Nous avons régulièrement comparé les résultats avec d'autres classifieurs comme les MLP ou les SVM.

3.1.3 L'analyse syntaxique

La seconde étape est alimentée avec les hypothèses de symboles (plusieurs classes possibles pour chaque symbole) et son but est d'extraire l'arbre syntaxique de l'expression. Trois tâches sont nécessaires : l'analyse des relations spatiales, l'analyse grammaticale, le choix de la meilleure interprétation. Ces trois tâches sont réalisées simultanément.

Le système est basé sur une grammaire CFG (Context Free Grammar) organisée en deux dimensions. Pour réduire la complexité, la grammaire est construite avec un ensemble de règles 1D verticales et horizontales. L'alternance des règles permet de couvrir les structures 2D efficacement. À l'application de chaque règle est associée une relation spatiale entre les sous-expressions qui sont combinées. Un coût est calculé en fonction du positionnement des éléments par rapport au modèle. L'expression choisie est celle avec le coût le plus faible.

Le coût global d'une expression candidate est celui de la racine de l'arbre relationnel C_{root} retourné par l'analyseur syntaxique :

$$C(\mathrm{SE}_j) = \begin{cases} C_{\mathrm{reco}}(\mathrm{sh}_j) & \text{si } \mathrm{SE}_j \text{ est le terminal } \mathrm{sh}_j \\ \alpha C_{\mathrm{struct}}(\mathrm{SE}_j) + \sum_i C(\mathrm{SE}_i) & \text{si } \mathrm{SE}_j = \cup \mathrm{SE}_i \end{cases} \tag{3.1}$$

Cette expression du coût global prend en compte le coût de reconnaissance de chaque symbole $C_{\mathrm{reco}}(\mathrm{sh}_j)$ et le coût de chaque relation spatiale $C_{\mathrm{struct}}(\mathrm{SE}_j)$. Le coût des relations spatiales est calculé à partir des alignements des lignes de base des sous-expressions ainsi que de la différence de leurs hauteurs respectives. Nous avons proposé deux formulations. La première est dite géométrique car elle est basée sur l'erreur quadratique de positionnement et de taille des fils par rapport au nœud parent. L'inconvénient de cette approche est qu'il y a beaucoup de paramètres à fixer empiriquement. C'est pourquoi nous avons proposé une approche probabiliste. Nous définissons les différences de positionnement et de taille d'une sous-expression SE_i par rapport à son parent SE comme suit :

$$dh_i = \frac{h_{\mathrm{SE}} - h_{\mathrm{SE}_i}}{h_{\mathrm{SE}}} \qquad dy_i = \frac{y_{\mathrm{SE}} - y_{\mathrm{SE}_i}}{h_{\mathrm{SE}}} \tag{3.2}$$

Les distributions des valeurs de dh et dy pour chaque élément d'une relation (sous-relation) sont modélisées par les histogrammes des occurrences, ou par des modèles gaussiens. La figure 3.2 illustre ces modèles pour la hauteur des sous-expressions dans la relation "superscript" : la base doit avoir à peu près la même hauteur et l'exposant doit avoir une hauteur en moyenne moitié plus petite. Après leur évaluation, le logarithme de ces probabilités est intégré dans l'équation 3.1 comme coût structurel.

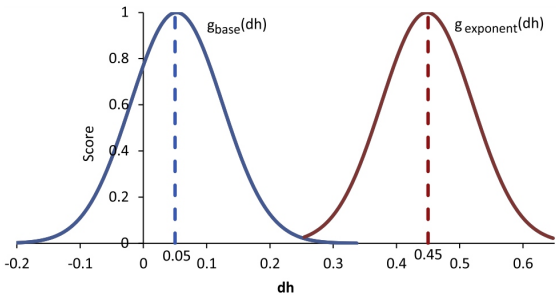


FIGURE 3.2 – Modèles gaussiens de la différence de taille de la relation "superscript". Extrait de [2] (Fig. 5).

Le problème de cette approche réside dans l'apprentissage des relations spatiales. En effet, pour obtenir les données d'apprentissage des modèles gaussiens, il faut appliquer l'analyse syntaxique aux expressions pour extraire les occurrences de chaque relation correspondant à notre grammaire. Pour cela, nous avons utilisé le modèle géométrique pour reconnaître les expressions en forçant la segmentation et la reconnaissance des symboles à partir de la vérité terrain. Même dans ce contexte toutes les expressions ne sont pas analysées correctement et les cas non reconnus ne sont pas utilisés dans cet apprentissage. C'est pourquoi dans la Thèse de F. D. Julca-Aguilar nous avons choisi une modélisation des relations spatiales qui peut être extraite directement des données sans passer par la grammaire.

Une description plus complète du système ainsi que les résultats sont présentés dans [2] disponible en annexe page 80.

Notre approche peut être comparée à celles de MacLean et al [74, 82] et Alvaro et al [73]. Ces systèmes sont très similaires même si certains choix sont différents. Par exemple les deux approches [73, 82] sont probabilistes car au lieu de minimiser une fonction de coût, ils maximisent le produit des probabilités de chaque étape, alors que [74] utilise une approche floue. Dans toutes ces approches, les relations spatiales sont calculées sur les boîtes englobantes des sous-expressions considérées dans la règle, mais Alvaro extrait plusieurs caractéristiques qui alimentent un SVM alors que MacLean définit empiriquement des règles floues à partir de l'angle et de la distance séparant les boîtes. Comme notre solution, ces approches définissent des classes de symboles pour adapter les relations aux types de terminaux (ascendant, descendant, normal, ...). Concernant la reconnaissance des symboles, différents classifieurs sont utilisés, notons que [74, 73] n'utilisent pas de classe de rejet mais que [82] définit un cas `NIL` pour les groupes de traits qui ne sont pas des symboles.

Ainsi que le souligne MacLean [82], il serait possible d'améliorer notre système en utilisant un cadre probabiliste encore plus rigoureux. Néanmoins, nous proposons un système entièrement entrainable (sauf la grammaire) qui permet de

s'adapter facilement à de nouvelles données comme le montrent les expérimentations réalisées sur les diagrammes dans la Thèse de M. Awal et dans l'article [31] dédié à l'analyse de ces diagrammes.

3.1.4 L'évaluation

Dès 2010 [41] nous avons mis en valeur le fait que l'évaluation de la reconnaissance des expressions mathématiques manuscrites ne pouvait se faire avec les outils classiques. Ce constat est venu concrètement d'un problème de normalisation de chaînes LaTeX lors de l'évaluation de notre système : lors d'une reconnaissance la sortie du système était une simple chaîne LaTeX représentant l'expression, mais cette chaîne était rarement identique à la vérité terrain même si la reconnaissance était correcte !

Une expression mathématique peut être représentée de différentes manières : une image, une chaîne LaTeX, une chaine MathML de présentation ou fonctionnelle, un arbre d'organisation de symboles (Symbol Relation Tree ou Symbol Layout Tree définit dans [83]), ... Le choix de la modélisation influencera le choix de la métrique. Il faut que l'évaluation permette de comprendre les points forts et points faibles d'un système. Connaître simplement le nombre d'expressions bien reconnues n'est pas très informatif de ce point de vue. Compte-tenu de la complexité du processus, il sera intéressant d'avoir des informations sur les tâches intermédiaires : segmentation des symboles, reconnaissance des symboles, reconnaissance des relations entre symboles. En prenant l'exemple de la figure 3.3 présentant deux segmentations d'une expression, dans les deux cas, l'interprétation (reconnaissance et analyse) peut être bonne ou mauvaise. Déjà Rhee et al [83] présentaient leurs résultats avec un taux de bonne segmentation, de bonne reconnaissance des symboles et un bon positionnement des symboles.

Plutôt qu'une simple évaluation binaire correct/incorrect il faut aussi une métrique permettant de savoir dans quelle proportion une expression est mal reconnue. Par exemple, dans [84] les expressions sont représentées par une chaîne de texte et la métrique est basée sur une distance d'édition ; dans [74] les expressions sont représentées par des arbres et les erreurs sont comptées en nombre d'opérations nécessaires pour faire correspondre les graphes ; dans [85] une expression est représentée par une image et la métrique est proportionnelle au nombre de pixels corrects.

Ces réflexions nous ont amené à présenter nos résultats en considérant la segmentation des symboles, leur reconnaissance et la reconnaissance globale de structure de l'expression en plus de la simple bonne reconnaissance globale de l'expression. Pour obtenir de façon fiable ces métriques, nous avons choisi une forme

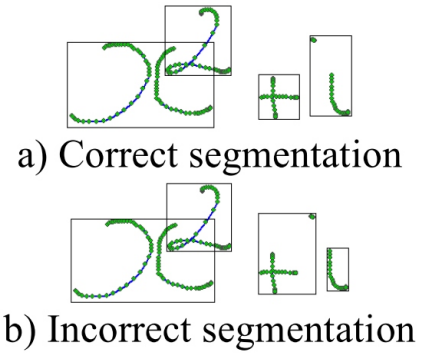


FIGURE 3.3 – Deux segmentations d'une expression manuscrite. Extrait de [41] (Fig. 6).

```
<msup>
  <mrow>
    <mo>(</mo>
    <mrow>
      <mi>x</mi>
      <mo>+</mo>
      <mn>2</mn>
    </mrow>
    <mo>)</mo>
  </mrow>
  <mn>3</mn>
</msup>
```

(a) Presentation markup

```
<apply>
  <power/>
  <apply>
    <plus/>
    <ci>x</ci>
    <cn>2</cn>
  </apply>
  <cn>3</cn>
</apply>
```

(b) Content markup

FIGURE 3.4 – MathML *Content* et *Presentation* de l'expresion $(x+2)^3$. Extrait de [41] (Fig. 2).

canonique du *MathML Content* (voir figure 3.4) ce qui permet une comparaison aisée des arbres.

Ces métriques ont été utilisées dans la première compétition CROHME et ont inspiré la métrique LG à base de graphes étiquetés présentée dans la section 4.3.

3.2 Projet DEPART - Thèse de S. Medjkoune

Le projet DEPART (Documents Écrits et Paroles – Reconnaissance et Traduction) s'inscrit dans la problématique de l'accès à l'information multimodale et multilingue. L'objectif était de rapprocher des compétences régionales de trois équipes des laboratoires des Pays de la Loire : l'équipe TALN du LINA à Nantes, l'équipe Traduction et Parole du LIUM au Mans et l'équipe IVC de l'IRCCyN. Il s'agit donc d'un projet structurant associant l'analyse du signal audio et manus-

crit au traitement automatique des langues. Le projet s'est concrétisé par 3 thèses encadrées à chaque fois par deux partenaires. Je vais décrire dans cette section la Thèse de S. Medjkoune co-encadrée avec C. Viard-Gaudin (directeur) et S. Petitrenaud du LIUM. Le projet s'est aussi concrétisé à plus long terme par la création du parcours ATAL (Apprentissage et Traitement Automatique de la Langue) du Master Informatique où les trois laboratoires continuent de collaborer.

Le travail réalisé porte sur l'étude, la conception et la validation d'un système de reconnaissance d'expressions mathématiques dans un cadre bimodal où l'on considère de façon complémentaire l'écriture manuscrite et la parole. Un article résumant ces travaux a été soumis à la revue IEEE Transactions on Human-Machine Systems (l'article est maintenant en révision mineure). L'article [21] disponible en annexe page 90 se focalise sur un seul aspect de la fusion mais résume les différents résultats obtenus.

3.2.1 Une base bimodale

La mise en place de ce système bimodal et sa validation requérant la disponibilité de données bimodales (chaque expression disponible à la fois sous format manuscrit en-ligne et sous format audio), nous avons collecté, complètement annoté et mis à disposition une base, nommée HAMEX [34], contenant 4 350 expressions bimodales couvrant différents domaines. Fort de l'expérience du projet CIEL, nous avons choisi des expressions toutes différentes réparties en 3 catégories correspondant à 3 tailles de vocabulaires présentés dans le tableau 3.1. Les expressions ont été soit générées pour le corpus Calculette soit extraites de Wikipedia (version française) pour les corpus WikiEM et WikiEM-Ext.

La collecte a été réalisée auprès de 58 scripteurs et 58 locuteurs. Les deux collectes, manuscrites et audio, ont été réalisées séparément. Pour la collecte de l'écriture, nous avons utilisé la technologie papier/crayon électronique, la figure 3.5 présente un exemple de formulaire utilisé. Les encres ont ensuite été extraites et entièrement annotées en InkML [1] incluant l'expression au format MathML [2]. Ces formats permettent de conserver l'encre, la segmentation des symboles et leur disposition dans l'expression. Pour l'audio, un formulaire équivalent était présenté au locuteur enregistré. Les symboles isolés ont aussi été enregistrés par chaque locuteur. Un seul fichier wav étant obtenu par personne, les expressions ont été segmentées et transcrites manuellement. Notons que chaque segment audio contient une expression complète et qu'il n'y a aucune synchronisation entre les deux modalités.

1. http ://www.w3.org/2003/InkML
2. http ://www.w3.org/1998/Math/MathML

Classes	CALCULETTE	WIKIEM	WIKIEM-EXT
Caractères		$abcdefgiknrsxyz$	$a \ldots z$
Lettres Grecques		$\alpha\beta\gamma\phi\pi\theta$	$\alpha\beta\gamma\phi\pi\theta$
Majuscules		XY	XY
Chiffres	$0 \ldots 9$	$0 \ldots 9$	$0 \ldots 9$
Opérateurs	$+ - \pm \times / \div$	$+ - \pm \times / \div$	$+ - \pm \times / \div$
Op. comparaison	$= \neq < \leq > \geq$	$= \neq < \leq > \geq$	$= \neq < \leq > \geq$
Op. élastiques		$\sum \int \sqrt{}$	$\sum \int \sqrt{}$
Op. ensemblistes			$\in \forall \exists$
Fonctions		cos sin log	cos sin log
Parenthèse	( )	( )	( )
Autre	.	$. \rightarrow$	$. \rightarrow \ldots \infty$
Nombre de Symb.	25	56	74

TABLE 3.1 – Ensemble de symboles dans chaque corpus de HAMEX. Extrait de [34]

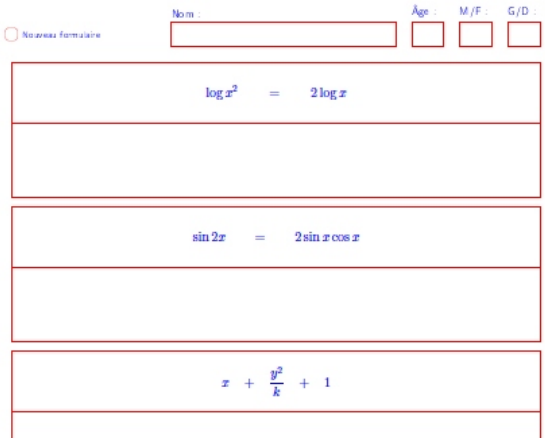


FIGURE 3.5 – Exemple de formulaire (vide) utilisé pour la collecte de l'écriture.

La base obtenue est découpée en deux (apprentissage et test) et a été ensuite utilisée pour les expérimentations sur la fusion présentée ci-dessous. Il s'agit encore à ce jour de la seule base bimodale écrit/parole pour les expressions mathématiques. La base est disponible gratuitement sur demande.

3.2.2 La fusion de modalités

La complémentarité entre plusieurs modalités est connue et exploitée dans différents domaines depuis longtemps (authentification, prise de notes, ...). Citons les travaux de Kaiser [86] dont le domaine applicatif est proche du nôtre : utiliser l'audio et le manuscrit pour reconnaître l'écriture sur un tableau blanc. L'idée de base est de constater que la reconnaissance de chacune des deux modalités est difficile, mais les erreurs de chacune sont différentes. Cette complémentarité a été vérifiée et exploitée à profit dans notre système, d'abord dans un cadre simplifié qui est celui de la reconnaissance des symboles mathématiques isolés, puis dans le cadre plus général et plus réaliste des expressions mathématiques complètes.

Fusion pour les symboles isolés

Pour cette première étape nous avons utilisé les échantillons de symboles isolés dictés dans la base HAMEX ainsi que ceux extraits des expressions manuscrites complétés de symboles isolés provenant de la base CIEL. A partir de ces symboles écrits/dictés isolés, nous avons généré toutes les paires de symboles possibles des deux modalités.

Deux classifieurs ont été créés. Pour les symboles manuscrits le TDNN défini dans la Thèse de M. Awal a été utilisé. Pour le classifieur de symboles audio, il n'était pas possible d'utiliser les systèmes complets comme celui utilisé ensuite car il nécessite des séquences plus longues. Nous avons donc utilisé un classifieur k-ppv plus simple basé sur l'alignement des séquences par DTW (voir [87]) avec l'ensemble des exemples d'apprentissage.

Plusieurs stratégies de fusions ont été testées pour fusionner les scores venant des deux modalités. Tout d'abord citons les méthodes à base de règles classiques : moyenne simple, moyenne pondérée par les performances des classifieurs, moyenne pondérée par les performances de chaque classe et la méthode de Borda. Nous avons aussi utilisé les fonctions de croyance [88] qui permettent entre autre de modéliser l'incertain. La dernière solution consiste à entraîner un classifieur (ici un SVM) pour faire la fusion. Les scores de sortie des classifieurs mono-modaux sont utilisés en entrées de ce classifieur de fusion.

Comme le montrent les résultats publiés dans [35] ce sont ces deux dernières solutions (fonctions de croyance et classifieur de fusion) qui ont donné les meilleurs résultats avec un net avantage pour le classifieur. La grande différence entre ces deux approches est que l'une est basée sur une règle de combinaison, c'est à dire que la règle est la même quelque soient les classifieurs fusionnés, et l'autre est une approche statistique et nécessite donc un apprentissage. C'est sûrement cet apprentissage qui permet une spécialisation de la fusion au contexte, c'est-à-dire aux données (symboles manuscrits) et au type de classifieur. Un exemple concret est la dynamique des scores : les scores du k-ppv ont une dynamique qui n'est pas comparable aux sorties d'un TDNN, même si tous les scores sont normalisés dans $[0, 1]$. Le classifieur est entraîné pour prendre en compte ces dynamiques. Un autre avantage de cet entrainement est que le score final d'une classe ne dépend pas seulement des scores de cette classe dans les deux sous-classifieurs mais de l'ensemble des scores.

Fusion pour les expressions complètes

Le système proposé dans la figure 3.6 est de type modulaire. Le module chargé du traitement du tracé manuscrit est le module principal. C'est au niveau de ce module que sont accomplies les diverses étapes d'interprétation. Le second module se charge de la modalité audio pour fournir une information supplémentaire pour guider la reconnaissance.

Avant de décrire les stratégies mises en œuvre pour la fusion, je vais donner une brève description de ce module de reconnaissance de la parole. Le système de transcription audio est basé sur le noyau CMU Sphinx [89] (à base de Modèles de Markov à états cachés), qui a été adapté à la reconnaissance du Français avec des modèles acoustiques développés au LIUM [90]. Nous avons aussi créé des modèles de langage spécifiques aux expressions mathématiques : des n-gram appris à partir des transcriptions de HAMEX et de phrases synthétiques générées à partir de chaines LaTeX d'équations extraites de wikipedia.

La figure 3.6 présente l'architecture globale du système développé. On retrouve dans la partie droite les modules présentés dans l'architecture générale d'un système de reconnaissance de documents manuscrits (figure 2.3). La stratégie de fusion se fait en remettant en cause les propositions faites par la partie en charge du signal manuscrit en trois points : la reconnaissance des symboles, la reconnaissance des relations et après l'interprétation.

La première étape de la fusion consiste à extraire les mots clés de la séquence transcrite par la reconnaissance audio. À chaque hypothèse de mot clé sont associés la ou les classes de symboles et/ou de relations ainsi que leur score de

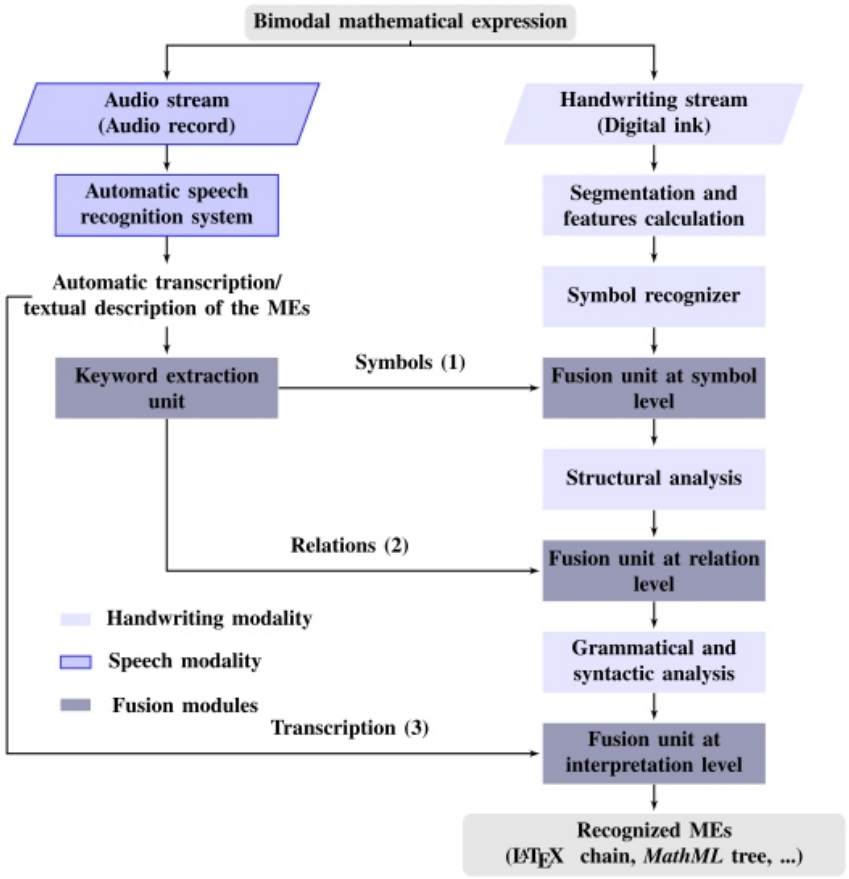


FIGURE 3.6 – Architecture pour la reconnaissance d'expressions mathématiques bimodales, extrait de la soumission à HMS.

reconnaissance *audio*. Certains mots peuvent être muets et n'avoir aucune classe correspondante, la plupart des mots sont associés à une seule classe de symbole ou de relation et certains peuvent avoir plusieurs classes ou relations associés (comme *diviser* ou *carré*). Les hypothèses de symboles et de relations audio sont respectivement utilisées pour ajuster les scores des hypothèses de symboles manuscrits et de relations spatiales. Les fusions se font par fonctions de croyance pour les symboles, nous discuterons ce choix par la suite. Pour les relations spatiales nous utilisons une solution plus simple de sac de mots : les relations présentes à l'audio sont avantagées par rapport aux absentes.

Ces deux premières stratégies ont pour but de réordonner les hypothèses de symboles et de relations pour que la fonction de coût de l'équation 3.1 ordonne différemment les interprétations possibles de l'expression manuscrite. Bien que les expérimentations ont montré une amélioration des résultats (voir [23]) la bonne réponse est encore souvent dans les n premières réponses.

Il reste une information présente dans le signal audio qui n'est pas utilisée : la séquence de symboles. La troisième étape de fusion utilise chaque interprétation possible d'une expression et génère plusieurs transcriptions possibles (il y a souvent plusieurs façons de dicter une expression). En confrontant la transcription audio avec des dictées possibles, l'hypothèse qui nécessite le moins d'opérations possibles est conservée.

Comme le montrent les résultats de [21] (disponible en annexe page 90) cette dernière fusion utilisée seule fait aussi bien que les deux premières combinées. Mais nos derniers résultats (présentés dans un article soumis à THM) améliorent encore le taux de reconnaissance en combinant les trois méthodes de fusion.

Il reste deux pistes à explorer pour compléter cette étude : l'utilisation d'un classifieur de fusion (qui était le plus performant avec les symboles isolés) et l'intégration de la fusion dans l'analyse syntaxique. Ces deux idées sont en fait liées. En effet, nous n'avons pas pu utiliser le classifieur de fusion par manque de données audio/manuscrit alignées (les transcriptions audio ne sont pas synchronisées avec le signal). Une grammaire étant capable de consommer en même temps les symboles audio et manuscrits permettrait de faire un alignement forcé pour l'apprentissage de ce classifieur mais surtout d'explorer plusieurs hypothèses d'alignements pendant la reconnaissance. Une solution pourrait être de construire un graphe d'hypothèses combinant les hypothèses manuscrites avec les hypothèses de mots audio. Cette idée mériterait d'être poussée plus loin, il faudrait sûrement également modifier l'implémentation de notre analyseur syntaxique.

3.3 Grammaires de graphes - Thèse F. D. Julca-Aguilar

La thèse de Frank D. Julca-Aguilar est issue d'une collaboration avec Nina Hirata, Associate Professor à l'Université de Sao Paulo. La thèse en co-tutelle s'est déroulée en partie à Nantes (1 année) et au Brésil. Cet échange a été l'occasion de partager nos connaissances respectives. Cette thèse a eu deux contributions majeures, l'utilisation du contexte pour la reconnaissance des symboles et des relations et surtout l'utilisation d'une grammaire de graphes pour l'analyse syntaxique.

Le système proposé suit le modèle présenté dans la figure 3.3 qui est assez similaire au modèle général présenté en introduction (figure 2.3).

Les trois premières étapes permettent la construction d'un graphe d'hypothèses étiquetées (voir figure 2.4) où les nœuds sont des hypothèses de symboles et les arcs des hypothèses de relations. La prochaine section présente comment les contextes de formes (*shape context*) ont été utilisés dans ces étapes. Les étapes suivantes réalisent l'analyse syntaxique par la construction d'une forêt d'arbres syntaxiques, leur évaluation et le choix du meilleur arbre.

3.3.1 Utilisation du contexte

Le contexte des formes a été utilisé suivant plusieurs approches. D'abord nous avons adapté le classique *shape context* [91] qui est utilisé dans le domaine hors-ligne pour la reconnaissance de formes isolées au signal en ligne [20]. La version proposée utilise des histogrammes polaires flous pour représenter la densité de points du signal suivant l'angle et la distance par rapport au centre. Pour conserver l'information en-ligne ces histogrammes sont répartis sur le signal de façon ordonnée. Les valeurs des histogrammes sont ensuite utilisées comme caractéristiques en entrée d'un MLP. Cette solution permet de nettement diminuer la complexité par rapport à la solution standard de [91] qui répartit les *shapes contexts* sur toute la forme à reconnaître et tente de les faire correspondre avec ceux des modèles.

Ces nouvelles caractéristiques ont montré de bons résultats sur les symboles isolés (même avec une classe de rejet). Lorsqu'il faut reconnaître les symboles dans une expression, il est aussi possible de capturer le contexte du symbole à reconnaître par rapport au reste de l'expression. La figure 3.8 illustre bien comment les traits qui n'appartiennent pas au symbole mais qui sont dans son contexte proche peuvent aider à sa reconnaissance. Nous avons notamment pu constater une nette amélioration de la détection des mauvaises segmentations.

Pour la reconnaissance des relations entre symboles ou entre sous-expressions, la plupart des systèmes utilisent des caractéristiques géométriques reliant les opé-

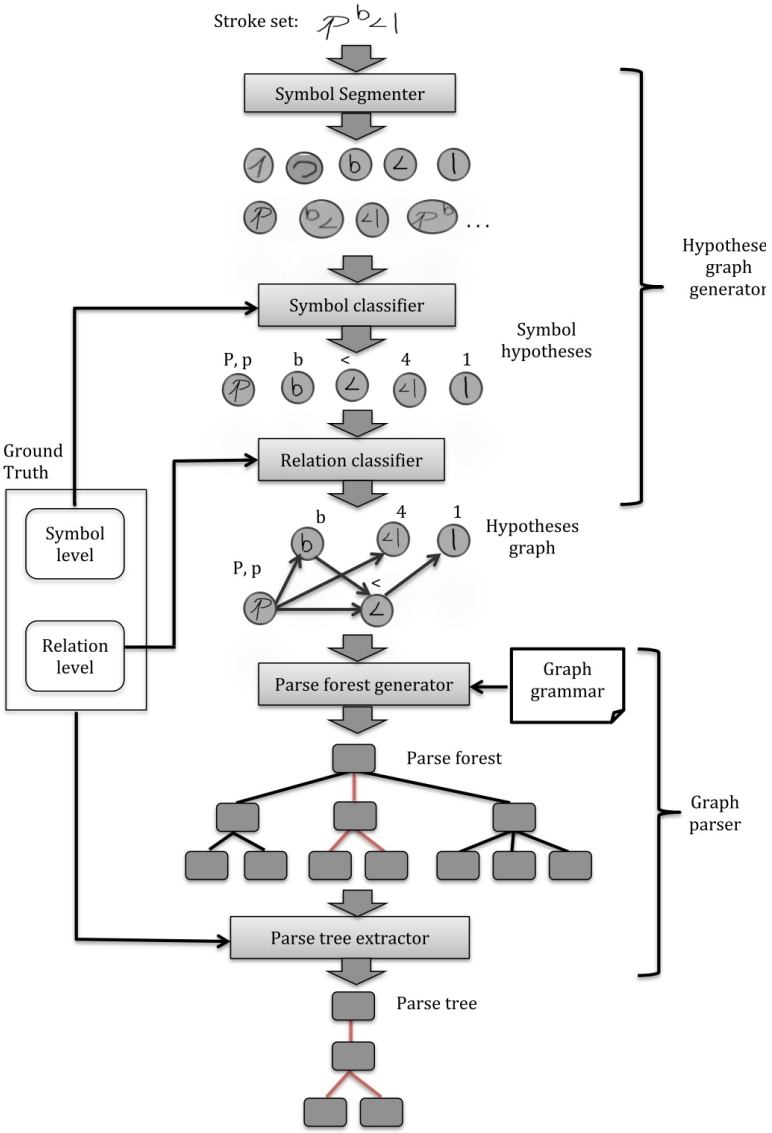


FIGURE 3.7 – Structure du système de reconnaissance utilisant une grammaire de graphes. Extrait de Thèse de F. D. Julca-Aguilar (Fig. 3.8).

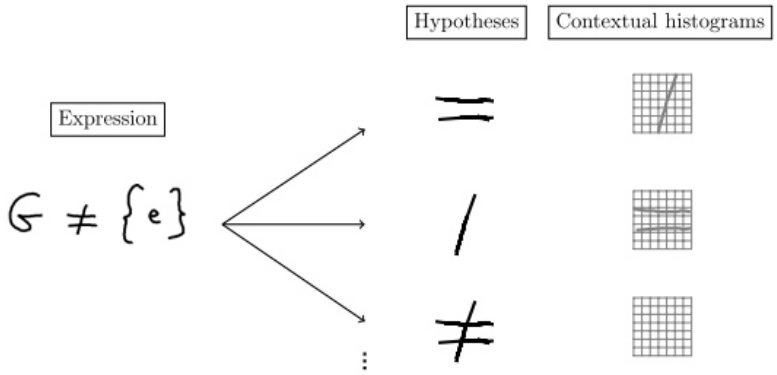


FIGURE 3.8 – Utilisation du contexte du symbole comme caractéristique. Extrait de Thèse de F. D. Julca-Aguilar (Fig. 4.7).

randes. Dans ces travaux nous avons proposé de représenter la position relative des opérandes par une image en niveaux de gris des rectangles englobants de ces opérandes. Comme pour la Thèse de M. Awal ces caractéristiques sont combinées à une information décrivant le type des opérandes : la position de la ligne de base et du corps de ligne, savoir s'il s'agit d'un symbole isolé ou d'une expression... Les performances de cette approche sont équivalentes à l'utilisation des caractéristiques géométriques, mais ces caractéristiques géométriques ont été choisies et optimisées manuellement au fil des publications. L'avantage de notre approche est qu'elle est simple à mettre en œuvre quel que soit le contexte d'utilisation.

Ces travaux sur l'utilisation du contexte sont à rapprocher de ceux présentés tout récemment dans la Thèse de L. Hu [72] du RIT qui est allé plus loin dans la modélisation du contexte des relations spatiales. Pour améliorer les performances, il faudrait utiliser aussi la forme des opérandes et pas seulement leur type. En effet, pour un même symbole, ses différents allographes peuvent beaucoup faire varier les valeurs des caractéristiques.

3.3.2 Grammaire de graphes

Très peu de systèmes de reconnaissance des expressions mathématiques manuscrites utilisent les grammaires de graphes. Nous pouvons citer [77] qui propose une première solution mais avec des limitations : une seule segmentation dans le graphe d'hypothèses, des structures de règles limitées, une analyse seulement ascendante basée sur CYK.

Dans notre approche, les parties droites des règles de la grammaire sont des graphes dont les nœuds sont des non-terminaux ou des terminaux (dans ce cas avec

une étiquette de symboles) et les arcs sont étiquetés avec une relation spatiale (dans la figure 3.9 : a : above, b : below, h : horizontal). Il n'y a pas d'autres contraintes, il peut s'agir d'un nœud isolé ou d'une structure complexe. La figure 3.9 montre un exemple de règle complexe.

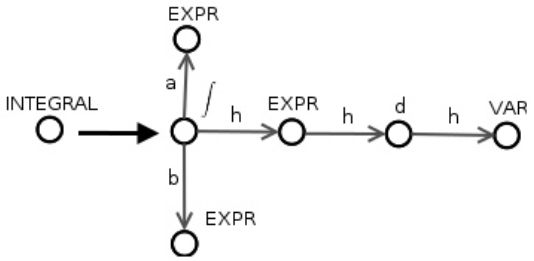


FIGURE 3.9 – Exemple de règle de grammaire de graphes (définition du non terminal `INTEGRAL`). Extrait de la Thèse de F. D. Julca-Aguilar (Fig 3.4(b)).

Pour l'algorithme d'analyse nous avons choisi une approche descendante. Ce choix a pour conséquence qu'à l'application de chaque règle, il faut trouver une partition de l'ensembles des traits qui corresponde au graphe de la partie droite de la règle (il faut qu'au moins un trait soit associé à chaque nœud). Cette approche est reconnue comme NP complet et c'est pourquoi les travaux existants ajoutent des contraintes sur l'ordre des traits dans les symboles : les symboles ne peuvent être composés que de traits suivant l'ordre temporel dans [77] ou suivant l'axe x dans [74] (ou y pour les règles verticales).

Nous avons choisi d'utiliser les connaissances extraites du graphe d'hypothèses pour optimiser la reconnaissance. En effet, pour qu'une partition de l'ensemble de traits corresponde à la partie droite d'une règle, il faut qu'au moins une hypothèse de symbole existe pour chaque nœud de la partie droite. De plus il faut que les hypothèses de relations entre les hypothèses de symboles retenues soient compatibles avec celle de la règle. Enfin, à chaque fois qu'un terminal est utilisé dans une partie droite (comme $\int$ et d dans l'exemple de la figure 3.9), il faut que son étiquette soit celle de l'hypothèse de symbole retenue pour ce nœud.

Le résultat de cette analyse syntaxique est une forêt d'arbres syntaxiques (les branches communes sont partagées) qui contient toutes les interprétations possibles. Il reste à évaluer le score de chaque arbre pour sélectionner le meilleur. Cette évaluation se fait de façon ascendante en sommant les coûts de reconnaissance des symboles et des relations.

Cette proposition se base entièrement sur les capacités des classifieurs de symboles et de relations à identifier les bonnes hypothèses et à rejeter les mauvaises. Dans le pire des cas, la complexité maximum théorique peut être atteinte, mais les expérimentations ont montré l'efficacité de notre solution.

3.4 Utilisation des BLSTM - Thèse T. Zhang

Les BLSTM (Bidirectionnal Long Short Term Memory) [78] sont des réseaux de neurones récurrents basés sur l'utilisation de blocs mémoires remplaçant les traditionnels neurones non-linéaires. Ces blocs mémorisent à court ou moyen terme chacun une information qui est contrôlée par 3 portes pour l'effacer, la combiner avec de nouvelles entrées ou l'utiliser en sortie. La récurrence se fait en parallèle de façon bidirectionnelle : dans le sens du temps et dans le sens inverse. Ces deux activations sont ensuite combinées par une couche perceptron classique pour obtenir une sortie pour chaque instant du signal. Si ces réseaux sont très performants c'est aussi grâce à l'utilisation d'une étiquette spécifique appelée *blank* qui permet au réseau de ne pas répondre la plupart du temps mais seulement de façon ponctuelle. L'utilisation de cette sortie spécifique impose d'utiliser un étage de CTC (Connectionist Temporal Classification), qui pallie la disponibilité de l'alignement temporel lors de l'apprentissage. Ces réseaux ont obtenu de meilleures performances que l'état de l'art dans plusieurs thématiques de reconnaissance de séquences (la parole, l'imprimé ou le manuscrit, ...) notamment grâce à leur capacité d'intégrer une partie du modèle de langage. Ils ont aussi été utilisés en analyse d'image par une application de la récurrence en 2D créant un maillage régulier d'étiquettes.

Comme nous l'avons vu dans les sections précédentes, les expressions mathématiques ne sont pas des séquences 1D ni des maillages réguliers en 2D. La thèse de Ting Zhang tente de répondre à la question : "Comment est-il possible d'utiliser les BLSTM pour la reconnaissance d'expressions mathématiques manuscrites". Deux obstacles se présentent. D'abord le langage mathématique est plus ouvert qu'une langue naturelle. En mathématique, il est possible d'utiliser (presque) n'importe quelle séquence de symboles alors qu'un langage naturel est davantage contraint. Le BLSTM peut-il nous dispenser de l'utilisation d'une analyse syntaxique coûteuse en 2D ? Le second obstacle est la topologie des expressions mathématiques, il faut réussir à sortir de la séquence d'étiquettes pour construire un graphe de symboles et de relations.

3.4.1 Restriction au cas des expressions 1D

Après une première étude sur les symboles isolés [17], nous avons commencé par reconnaître les expressions qui peuvent être modélisées par une chaîne, comme $a + b$, $a + b^2$ ou a_{+b}. En effet en utilisant un SRT (Symbol Relation Tree), ces expressions sont une alternance de symboles et de relations. Il s'agit donc d'associer aux traits une étiquette de symbole et entre les traits une étiquette de relation ou de segmentation. La figure 3.10 montre un exemple d'une séquence de 4 traits

reconnus dans l'ordre temporel. On peut remarquer que comme dans les graphes LG introduits dans la section 4.3, les étiquettes de segmentation sont les mêmes que les étiquettes de symboles, donc le symbole + en deux traits est reconnu par la séquence $+, +, +$. Pour étiqueter les relations spatiales, nous créons des points intermédiaires lors du ré-échantillonnage du signal.

En filtrant les expressions de la base CROHME pour ne conserver que les expressions linéaires, nous avons pu construire une base d'apprentissage et une de test. Les premiers résultats présentés dans [52] ont montré que les BLSTM peuvent effectivement reconnaître les expressions mathématiques et leurs relations spatiales.

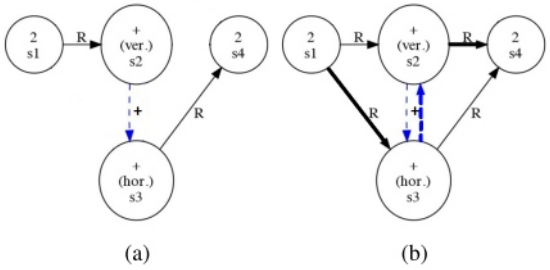


FIGURE 3.10 – L'expression 2+2 est écrite en 4 traits (comme dans la figure 4.1.a). (a) nœuds et arcs reconnus par le BLSMT. (b) arcs ajoutés pour obtenir un graphe valide. Extrait de [52].

Bien sûr cette approche est très restrictive. Toutes les expressions dont le SRT est un arbre ne peuvent pas être reconnues. De plus, même dans les expressions linéaires, les symboles dont les traits ne sont pas écrits en séquence temporelle ne peuvent pas être segmentés correctement.

3.4.2 Reconnaissance d'expressions 2D

La plupart des expressions mathématiques ne peuvent pas être couvertes complètement par une seule séquence linéaire. Nous avons choisi de parcourir plusieurs fois les traits de l'expression dans un ordre différent pour reconstruire les différentes branches du SRT. Dans un premier temps, nous créons un graphe mettant en relation les traits de l'expression avec tous les arcs potentiellement intéressants à parcourir. Il faut couvrir les possibles relations spatiales et les arcs de segmentation. Les critères utilisés pour relier les traits sont principalement géométriques : trouver le trait le plus proche dans une direction donnée, connecter des traits qui se croisent... La figure 3.11 montre un exemple de graphe contenant les arcs générés pour une expression donnée. Il faut un taux de rappel des arcs de la vérité

terrain proche 1 et une précision la plus forte possible. Pour cela, en plus de la séquence temporelle, une dizaine de parcours aléatoires du graphe sont évalués par le BLSTM. Le choix de ces parcours aléatoires est fait de manière à couvrir le plus possible le graphe.

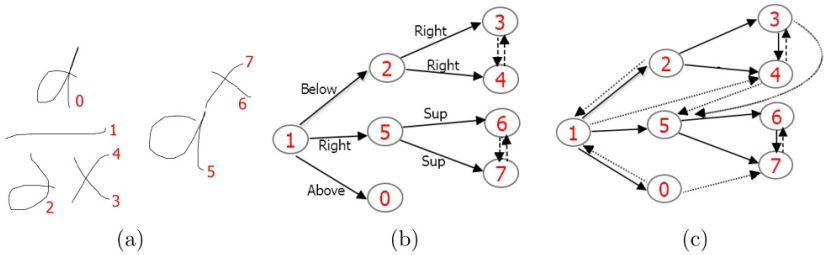


FIGURE 3.11 – (a) L'expression $\frac{d}{dx}a^x$ écrite en 8 traits ; (b) arbre vérité terrain à couvrir (arcs de segmentation en pointillé) ; (c) ensemble d'arcs générés pour créer les chemins de reconnaissance, les traits pleins sont les arcs corrects, les arcs en pointillés sont en surnombre et les deux en tirets sont manqués.

Une fois les chemins étiquetés, il s'agit de reconstruire les symboles et les relations entre symboles. Pour l'instant ces deux étapes sont faites successivement en combinant les scores de reconnaissance provenant de chacun des chemins.

Les premiers résultats [13] nous placent à la troisième ou quatrième place par rapport à la dernière compétition CROHME. Nous avons donc démontré la faisabilité de l'approche. Un avantage notable de notre système est l'absence de modèle de langage ce qui permet de limiter la complexité malgré les multiples parcours à reconnaître.

Par rapport au schéma global proposé en introduction, ces travaux brouillent les frontières des étapes de segmentation, de reconnaissance des symboles et des relations en déléguant ces tâches à un seul classifieur qui inclut aussi une partie du modèle de langage. Un objectif au delà de cette thèse sera de proposer une structure de LSTM qui ne soit pas une chaîne mais un arbre ou un graphe orienté acyclique. Ainsi on se rapprocherait d'une approche globale se passant de la combinatoire des étapes de segmentation et de reconnaissance des symboles et de leurs relations.

Chapitre 4

Les Compétitions CROHME

Ce chapitre présente les compétitions CROHME (*Competition of Recognition of On-line Handwritten Mathematical Expressions*). Il reprend et complète des éléments expliqués en détails dans l'article "Advancing the state of the art for handwritten math recognition : the CROHME competitions, 2011–2014" disponible en sélection page 95. Après 4 éditions [36, 29, 24, 22], en 2016 nous organisons la $5^{ème}$ compétition [0].

4.1 Organisation des compétitions

Avant l'organisation de la première compétition CROHME, les chercheurs développaient leur propre base locale d'apprentissage et de test. Même si certaines étaient de taille conséquente et disponibles gratuitement, elles n'étaient pas effectivement utilisées. Citons les bases MathBrush [92], ExpressMatch [93] et MfrDB [94]. Ces bases utilisent des listes de symboles différents, des complexités d'expressions différentes, des protocoles différentes (e.g. les expressions sont-elles les mêmes pour chaque scripteur ?), des formats d'encodage différents et des types de vérité terrain différents... Dans ces conditions il était difficile de comparer de façon fiable les systèmes.

Un premier apport de CROHME a été la mise en commun de ces ressources en proposant un format commun, différents niveaux de complexité et en réutilisant ces bases existantes pour obtenir une base suffisamment importante pour permettre un apprentissage et une évaluation pertinents mais surtout communs. Un second apport a été dans la mise à disposition d'outils de traitements des expressions, mais surtout pour l'évaluation des systèmes.

Les compétitions sont divisées en différentes tâches de complexités différentes. Les trois premières années (2011-2013) les différentes tâches consistaient en des ni-

veaux de difficulté différents correspondant à des jeux de symboles utilisés de taille croissante (jusqu'à 101 classes) et des grammaires d'expressions plus complexes. En 2015 nous avons introduit les tâches de reconnaissance des symboles isolés (extraits des expressions complètes) et de reconnaissance des matrices. Enfin en 2016, nous avons 4 tâches représentant les différentes difficultés de la reconnaissance d'expressions :

— reconnaissance des symboles isolés,

— reconnaissance des structures (à partir des symboles déjà segmentés et étiquetés),

— reconnaissance des expressions complètes (tâche principale),

— reconnaissance des expressions matricielles.

Nous pouvons voir que les deux premières tâches permettent l'évaluation des étapes principales d'un système complet : la reconnaissance des symboles et la reconnaissance des structures. Comme nous le verrons dans la partie 4.3, le but de la compétition n'est pas seulement de mettre en avant le meilleur système, mais surtout de permette une évaluation et comparaison fines des systèmes.

Le tableau 4.1 montre l'évolution chiffrée de la compétition. Nous pouvons voir qu'un nombre croissant de laboratoires sont intéressés par cette thématique. Parmi les 11 participants de 2016, 2 entreprises sont présentes (*Myscript* et *Wiris*).

	2011	2012	2013	2014	2016
Nombre de grammaire	2	3	2	1	1
Nombre de classes	57	75	101	101	101
Exp. en apprentissage	921	1341	8 836	8 836	8 836
Exp. en test	348	486	671	986	1 100
Mat. en apprentissage				362	362
Mat. en test				175	250
Nombre de participants					
Tâche symboles isolés				8	11
Tâche structures					6
Tâche expressions	4	6	8	7	8
Tâche matrices				2	3

TABLE 4.1 – Evolution de la compétition CROHME. Extrait complété de [22].

4.2 Les bases et corpus créés

Le choix des expressions utilisées en apprentissage et en évaluation est important, comme dans une tâche de TALN. En effet, si la reconnaissance des expressions

mathématiques se résumait à la reconnaissance de ses symboles, une base de symboles isolés où chaque classe serait équitablement représentée aurait suffit. Dans la section précédente, nous avons relevé les biais des bases existantes. Si l'objectif est de faire un système de reconnaissance utilisable, il faut constituer des bases d'apprentissage et de test réalistes. Les expressions mathématiques peuvent couvrir un large domaine (dans le projet CIEL, nous avons recensé jusqu'à 200 symboles). Nous avons donc défini progressivement un sous domaine des expressions dans lequel se placent les compétitions. Quatre ensembles de symboles ont été définis (de taille 37 et 57 en 2011, 75 en 2012 et 101 symboles à partir de 2013) associés à 4 grammaires permettant des structures de plus en plus complexes.

Comme le montre le tableau 4.1, chaque année nous avons réalisé un effort pour mettre à disposition de nouvelles données. Les bases d'apprentissage se cumulent jusqu'à atteindre plus de 8 000 expressions en 2013 et de nouvelles données de tests sont créées spécifiquement pour chaque compétition. Près de 500 scripteurs différents ont participé. Les expressions ajoutées chaque année (en apprentissage ou en test) sont à chaque fois inédites, c'est-à-dire qu'une expression en test ne devrait pas apparaitre en apprentissage.

Pour générer chaque année un corpus d'expressions différentes, il a fallu trouver une source d'expressions réalistes et définir un critère de choix des expressions dans ce corpus. Mis à part les expressions qui proviennent des bases existantes (soit une grosse partie de la base d'apprentissage), toutes les expressions proviennent des bases Wikipedia (version *en*) et pour CROHME 2016 ArXiv (articles des années 2000 et 2001 [1]). Ces expressions ont ensuite été normalisées et filtrées par les grammaires définissant les tâches. Pour faire la sélection des expressions, il nous semblait important que les propriétés des nouvelles bases de tests restent à peu près constantes : fréquence d'apparition des symboles et de certaines structures, complexité des expressions... Nous avons donc mis au point un algorithme itératif de sélection des expressions basé sur le calcul de fréquence des termes dans le corpus en construction par rapport à un corpus de référence. Une description détaillée est donnée dans [1] disponible en sélection page 95.

En 2016 nous avons aussi fourni aux participants un corpus d'expressions sous forme de chaines LaTeX pour leur permettre d'apprendre des modèles de langage. En effet, la plupart des modèles utilisés sont des grammaires stochastiques dont l'apprentissage nécessite beaucoup de données. Nous espérons que ce corpus permettra aux participants de faire de nouvelles propositions de modèles adaptés aux expressions mathématiques (comme des n-gram de symboles par exemple).

1. source du corpus ArXiv : KDD Cup 2003

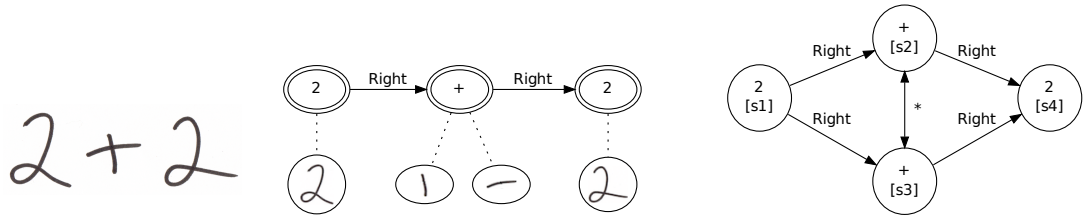
4.3 Métriques d'évaluation

En plus des efforts sur la mise à disposition de bases et de corpus d'expressions, les compétitions CROHME ont été l'occasion d'améliorer grandement les métriques utilisées pour l'évaluation des systèmes de reconnaissance d'expressions mathématiques. Les limites des métriques d'évaluation existantes ont déjà été exposées dans la section 3.1.4. Une métrique utile doit avoir plusieurs propriétés importantes : permettre une mesure absolue pour comparer deux systèmes ou l'évolution d'un système ; permettre de qualifier les performances des différentes parties d'un système pour l'améliorer ; ne pas être trop abstraite pour être compréhensible par le concepteur. Le simple taux de reconnaissance au niveau expression permet une mesure absolue, mais ne donne aucun détail quant aux erreurs commises. Les mesures de taux de bonne segmentation, bonne reconnaissance des symboles et des relations sont intéressantes pour détailler le fonctionnement d'un système mais difficilement utilisables pour comparer deux systèmes de façon absolue. Par exemple, il est difficile de comparer deux mauvaises interprétations d'une même expression, notamment parce que les structures des expressions obtenues sont différentes. Il s'agit typiquement d'un problème de distance d'édition de graphes.

Grâce à une collaboration fructueuse avec R. Zanibi (Associate Prof. au RIT, NY), nous avons développé une métrique pour comparer deux interprétations d'un même document structuré. Cette métrique est décrite en détail dans les articles [1, 25, 37], je vais en donner ici les grands principes pour illustrer ses avantages.

Nous avons choisi de représenter les expressions par un graphe se basant sur la seule chose constante quelque soit l'interprétation : les traits. Les traits sont indivisibles, portent chacun une étiquette de symbole et sont en relation par des arcs représentant la segmentation en symboles ou les relations spatiales entre eux. La figure 4.1 illustre cette modélisation par un graphe nommé LG (pour *Labeled Graph*).

Deux interprétations d'une même expression (e.g. la vérité terrain et le résultat d'une reconnaissance) conduisent donc à deux LG différents mais partageant la même numérotation des nœuds. Il est donc possible de comparer ces deux graphes de façon très efficace puisqu'il n'y a pas d'insertion ou de suppression de nœuds à considérer. Pour une expression de n traits, il y a n étiquettes de nœuds et n^2 étiquettes d'arcs à comparer. Comme la représentation est canonique, si les graphes sont identiques, alors les expressions sont identiques. En triant les étiquettes par type (symboles, segmentation, relations), nous avons créé des métriques dites "niveau traits" mesurant (rappel et précision) combien de traits sont bien étiquetés, combien de relations de segmentation sont détectées et combien de relations spatiales sont correctement reconnues. La fusion de ces métriques, permet de dire quelle proportion de l'expression est bien reconnue.



a. Expression b. Symbol layout tree c. Stroke label graph

FIGURE 4.1 – Une expression manuscrite en 4 traits (a), son interprétation représentée par un *symbol layout tree* qui met en relation des symboles composés de traits (b) et son *stroke label graph* qui est la projection du graphe précédent sur les traits (c). Extrait de [1], Fig. 2.

Un avantage de ces métriques bas niveau est qu'elles pourraient facilement être utilisées comme fonction de coût dans le processus d'apprentissage d'un système prenant ses décisions à ce niveau. Cela pourrait par exemple être une perspective des travaux de la Thèse de T. Zhang. D'un autre coté l'inconvénient de ces métriques est qu'elles sont difficilement interprétables. C'est pourquoi nous avons aussi défini les métriques de niveau objet.

Un symbole composé de plusieurs traits est représenté par un sous-graphe complètement connecté (une clique) où tous les nœuds et tous les arc portent la même étiquette (en fonction de la convention de nommage des arcs de segmentation). À partir de cette définition, il est assez facile d'extraire les taux de rappel et de précision pour la segmentation et la reconnaissance des symboles. Pour qu'une relation entre symboles soit bien reconnue, il faut que ses symboles soient bien segmentés et que tous les arcs entre leurs traits soient bien étiquetés. À partir de ces métriques, nous pouvons aussi extraire des informations statistiques intéressantes à propos des erreurs faites par les systèmes, par exemple les matrices de confusion ou un histogramme du nombre d'erreurs par expression.

Pour l'analyse des erreurs d'un système nous avons récemment proposé de calculer un histogramme des erreurs de structures. Ce concept, présenté dans [1], consiste à générer tous les sous-graphes de taille fixe du SRT issus de la vérité terrain de l'expression. Ces sous-graphes correspondent à des n-gram dans une séquence. Chaque sous-graphe est comparé à la structure correspondante dans le résultat de la reconnaissance. Chaque erreur est mémorisée dans un histogramme à 3 dimensions (sous-graphe de symboles correct, sous-graphe correspondant niveau traits, sous-graphe de traits en erreur). La figure 4.2 montre un exemple de résultat pour un bigram de symboles particulier '$x+$' qui est l'erreur la plus fréquente de

notre système en 2014. Quatre structures de traits correspondent à cette situation (1 ou 2 traits pour le '+', 1, 2 ou 3 traits pour le 'x'), pour chaque situation les erreurs sont colorées en rouge et le compte des erreurs est montré. Cet outil permet une analyse fine des erreurs.

Notre modélisation permet à chaque nœud et arc de porter plusieurs étiquettes éventuellement pondérées par un score. Cette fonctionnalité ne va pas être détaillée ici mais a été utilisée dans plusieurs contextes. En premier lieu elle a permis de s'adapter aux tâches de reconnaissance des matrices [22]. En effet dans une matrice il y a plusieurs niveaux d'étiquetage correspondants aux structures des matrices : nous avons défini les super objets *matrice*, *colonne*, *ligne* et *cellule* qui viennent se superposer aux structures des sous-expressions contenues dans et autour des matrices. Cette capacité a aussi été utilisée dans une expérimentation de fusion de plusieurs interprétations d'une même expression. Ainsi, nous avons pu présenter dans [1] une fusion théorique des décisions de tous les systèmes ayant participé à la dernière compétition.

Je conclus ce chapitre en précisant que tous ces outils sont sous licence libre (Creative Commons CC BY-NC-SA 3.0) et donc disponibles gratuitement. Ils peuvent facilement être utilisés pour l'évaluation de la reconnaissance d'autres types de documents structurés, pour des documents en-ligne (par exemple pour les diagrammes dans [32, 9]) mais aussi pour des documents hors-ligne en redéfinissant la primitive de base représentée par chaque nœud.

Le papier de revue [1] a reçu un très bon accueil de la communauté comme en témoigne le nombre de chercheurs qui m'ont demandé le document dès son acceptation mais surtout les commentaires des relecteurs du papier ; extraits choisis :

Relecteur #1 :

> This paper is a great contribution for the future research on mathematical expression recognition. The structure of the paper is good and it is well written and prepared. [...]

Relecteur #2 :

> First of all, let me thank the authors for their efforts on organizing a great competition, CHROME, and also for their efforts on summarizing their past competition results into this great paper. I am totally sure that this paper can be a "monument" for present and future researchers in this field – doubtlessly, this paper should be accepted (after minor revisions).
>
> I highly appreciate this paper by the following points.
>
> 1) This paper describes the competition protocols (including evaluation schemes) very clearly. This might be useful for the researchers

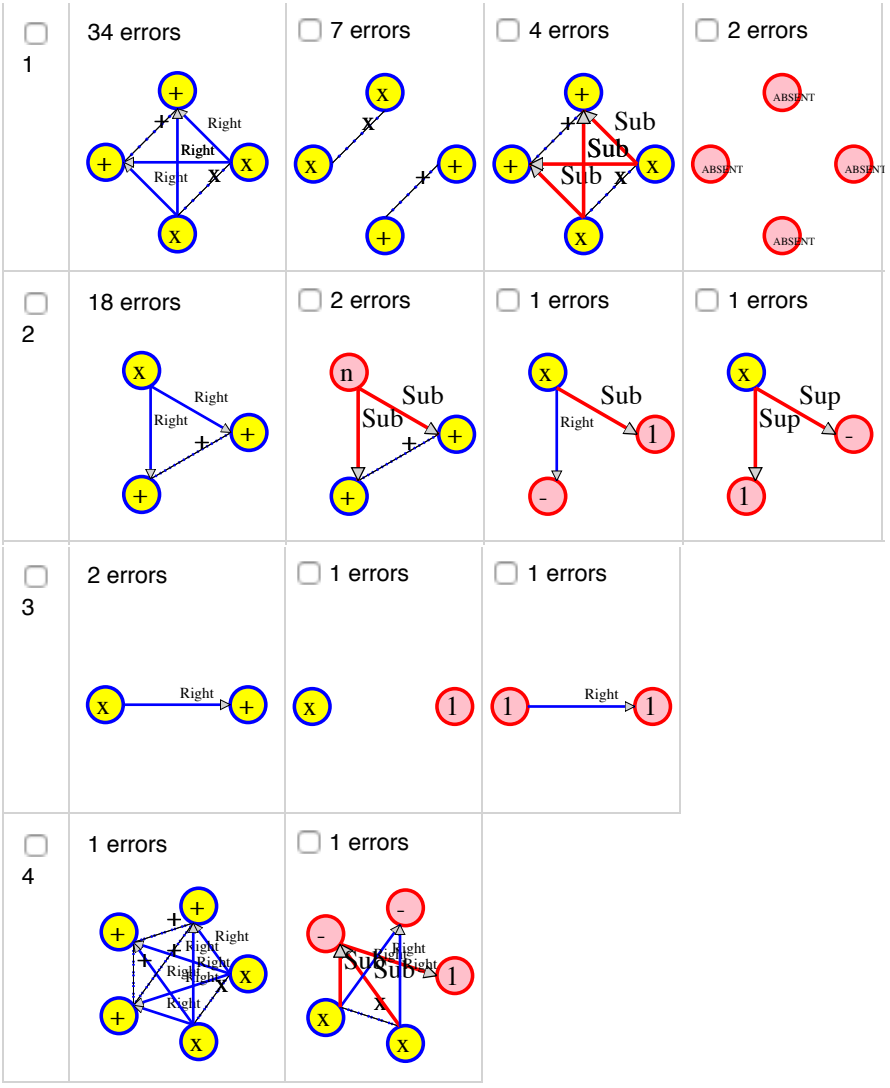


FIGURE 4.2 – Histogramme des erreurs de structures pour la sous expression '$x+$' pour notre système testé sur la base de test CROHME 2014. Extrait de [1] (Fig. 5).

who want to organize any other competitions of structural pattern recognition. It is also interesting to read the revisions of the evaluation schemes ; the organizers never stop to improve their competitions !

2) This paper analyzes their dataset from various perspectives. As far as I know, this has never been done for online math datasets (whereas for offline machine printed datasets, for example, INFTY group has done a similar analysis). [...]

3) The paper analyzes trends in the methodologies of the participated systems. Most competition reports just treat the participated systems as black boxes. This paper paid great efforts for classifying their methodologies in different levels. [...]

4) Instead of just showing a "big accuracy table" which just lists the accuracy of the participated systems, the organizers made further (very interesting) analyses, which can be done only the organizers ! [...]

I am really impressed by the organizers (i.e,. the authors) attitude to fully utilizing their competitions for this research field. I could confirm that "competitions are not just for showing the top accuracies." (Actually, the absolute accuracy does not make any scientific sense — if we use a different dataset, the accuracy might change...) Again, this paper can be a "monument" showing the fact that deeper analyses of competition results can yield many important and immortal information for present and future researchers. [...]

Chapitre 5

L'analyse de structures par les graphes

Nous avons vu dans les chapitres précédents l'importance de la modélisation des problèmes par des graphes. Dans ce chapitre je vais développer trois axes de recherche autour des modèles graphiques mais avec des domaines applicatifs différents des expressions mathématiques.

5.1 Reconnaissance de diagrammes - MRF

Pour montrer la généricité de nos approches, nous avons dès la Thèse de M. Awal cherché un autre langage visuel structuré en-ligne. Nous avons choisi les diagrammes manuscrits. Il s'agit d'un challenge différent car le langage a des propriétés différentes par rapport aux expressions mathématiques : peu de symboles, une grammaire assez simple mais des relations spatiales de nature complètement différente, beaucoup de libertés sur la mise en page, la présence de texte dans et hors des diagrammes et surtout la structure du document qui n'est pas un arbre. La figure 2.1(a) montre un exemple typique de diagramme.

La constitution d'une base ainsi qu'une première étude [31] ont été réalisées en collaboration avec G. Feng (Nanjing University, Chine). Grâce à cela, l'approche globale d'apprentissage et de segmentation proposée dans la Thèse de M. Awal a été évaluée sur ce type de document. Ces premiers résultats ont montré le bon comportement de notre approche basée sur l'apprentissage du rejet mais aussi la limite du type de grammaire que nous avons utilisé qui ne permet pas de représenter les boucles présentes dans les documents.

Nous avons ensuite repris ces travaux avec une approche cette fois-ci purement structurelle en utilisant le système DMOS [95] de l'équipe Intuidoc de l'Irisa

(Rennes). Cette approche présentée dans [9] utilise conjointement des informations hors-ligne (les lignes droites, classiquement utilisées par DMOS) et de l'information en-ligne (les traits de l'écriture). L'information en-ligne permet de mieux segmenter les symboles en désambiguïsant deux cas : deux segments appartenant au même trait appartiennent forcément au même symbole mais deux segments qui sont connectés mais de traits différents n'appartiennent pas forcément au même symbole. DMOS a aussi permis de résoudre en partie le problème de la structure non arborescente des documents en permettant de réutiliser des objets déjà expliqués par la grammaire lors de l'analyse.

Bien que les résultats soient meilleurs avec cette nouvelle approche, il manquait une information statistique provenant de la forme des traits et de leurs relations spatiales. Nous avons donc voulu utiliser un outil permettant d'associer un label à chaque trait et chaque relation spatiale, bien adapté à la structure de graphe des documents et tenant compte du contexte de chaque trait : les MRF (*Markov Random Field*).

Des travaux récents [96, 97] sur l'utilisation des MRF sur les documents en-ligne se contentent d'étiqueter les traits du document puis par un post-traitement de les regrouper pour la segmentation. L'approche que nous proposons permet de prédire les étiquettes des traits mais aussi celles des relations entre ces traits.

D'une façon générale un MRF cherche à maximiser la probabilité de l'équation 5.1 où $\mathbf{X}$ représente les traits d'un document et $\mathbf{Y}$ est l'ensemble des étiquettes de ces traits. Cette probabilité jointe est le produit pour toutes les cliques c de la fonction potentielle ϕ qui lie les observations aux étiquettes par les paramètres w.

$$\tilde{P}(\mathbf{X}, \mathbf{Y}) = \prod_{c}^{\mathcal{C}} \phi(\mathbf{X}_c, \mathbf{Y}_c; w) \tag{5.1}$$

La plupart du temps les cliques utilisées sont les singletons Y_i et les paires d'étiquettes (Y_i, Y_j), donc les fonctions potentielles sont de la forme $\phi(\mathbf{X}_i, \mathbf{Y}_i; w)$ pour les singletons et $\phi(X_i, X_j, Y_i, Y_j; w)$ pour les paires.

Nous avons proposé d'ajouter l'ensemble des étiquettes de relations $\mathbf{W}$ pour obtenir l'équation 5.2 où trois niveaux de cliques sont considérés.

$$\tilde{P}(\mathbf{X}, \mathbf{Y}, \mathbf{W}) = \prod_{c}^{C_e} \phi(\mathbf{X}_c, \mathbf{Y}_c; w) \prod_{c}^{C_r} \phi(\mathbf{X}_c, \mathbf{W}_c; w) \prod_{c}^{C_i} \phi(\mathbf{X}_c, \mathbf{Y}_c, \mathbf{W}_c; w) \tag{5.2}$$

On retrouve dans l'ordre des fonctions potentielles d'association des étiquettes aux traits (singletons), des relations aux paires de traits et une fonction vérifiant la cohérence des triplets (Y_i, W_{ij}, Y_j).

L'apprentissage de ce type de MRF est assez coûteux et pour réduire la dimension des paramètres w nous avons choisi d'utiliser comme caractéristiques des fonctions potentielles les sorties d'un classifieur de traits et de relations.

Nous avons réalisé deux expérimentations avec les MRF. La première n'utilise que la partie étiquetage des traits puis combine ces informations avec l'approche grammaticale DMOS décrite en début de section (en collaboration avec A. Lemaitre de IntuiDoc). La seconde utilise le modèle complet et utilise les étiquettes **W** pour réaliser la segmentation du document et déduire les relations entre les symboles.

Ces travaux sur l'utilisation des MRF ont été réalisés pendant le stage de Master de C. Wang et ont conduit à la soumission d'un papier à IJDAR intitulé "Online Flowchart Understanding by Combining Max-margin Markov Random Field with Grammatical Analysis", (toujours en relecture) et un papier à ICFHR2016 [16].

5.2 Extraction de connaissances symboliques - Thèse de J. Li

Cette section présente les travaux de Thèse de J. Li qui ont été publiés dans deux revues [5, 3]. Cette dernière est disponible en sélection d'articles page 112.

Tout langage graphique comporte d'une part des symboles élémentaires, par exemple un alphabet ou des formes graphiques propres à un langage métier (organigramme, schéma électrique...), et d'autre part des règles de composition permettant de donner globalement un sens au document produit. La connaissance des symboles élémentaires et de leurs relations nous permet d'interpréter ces messages manuscrits (tracés).

La collecte d'échantillons d'écritures et l'étiquetage au niveau de chaque trait sont des tâches difficiles et fastidieuses surtout sur un langage graphique inconnu (i.e. sans système de reconnaissance automatique existant). Ce constat a motivé le développement d'un système de plus haut niveau permettant d'automatiser cette procédure d'étiquetage fastidieuse. Notre approche consiste dans un premier temps à regrouper les symboles en des ensembles homogènes. Ces ensembles de symboles peuvent ensuite être facilement étiquetés ce qui réduit le coût de l'étiquetage symbolique. Sans connaissances symboliques a priori du langage graphique cette tâche nécessite des approches non supervisées permettant de découvrir l'alphabet des symboles.

Cet objectif de réduction du coût d'étiquetage se retrouve souvent dans les documents hors-ligne. On peut citer les travaux récents de L. Guichard [98] qui après

un clustering non supervisé des mots font intervenir l'utilisateur pour étiqueter tout un cluster. Les approches classiques se basent sur une segmentation unique à analyser à l'échelle des mots, des pseudo-lettres ou des graphèmes. L'originalité de notre approche est que nous recherchons des structures fréquentes composées des éléments de base organisés avec les mêmes relations.

Le principe de l'approche proposée est résumé par la figure 5.1. Le cœur de l'algorithme itératif est basé sur SUBDUE (*SUBstructure Discovery Using Examples*) [99] qui applique le principe MDL (*Minimum Description Length*) aux graphes étiquetés. Un graphe de relations similaires à ceux utilisés pour la reconnaissance des expressions mathématiques est donc défini en connectant les traits entre eux par des relations. Par définition, les classes de formes et les classes de relations ne sont pas connues a priori. Nous avons donc fait une quantification des formes et des relations pour générer des hypothèses d'étiquettes. Ensuite SUBDUE extrait les structures les plus fréquentes pour faire émerger les structures de niveau symbole. Une fois ces premières hypothèses de symboles définies, il faut remettre en question les clusteurs de formes et de relations déjà définies pour tenir compte de ces nouveaux terminaux.

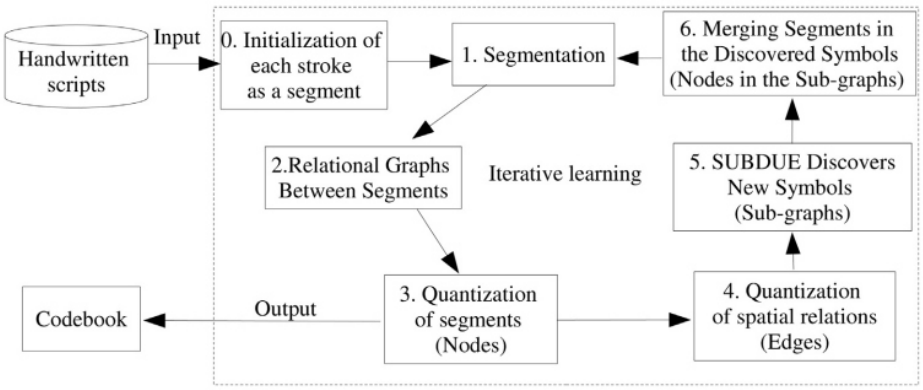


FIGURE 5.1 – Principe de l'extraction de symboles multi-traits. Extrait de [3] Fig. 2.

Nous avons étudié le comportement de cet algorithme sur plusieurs langages graphiques en fonction du nombre d'hypothèses de symboles générés à chaque itération, du nombre d'itérations et de la façon de faire les quantifications des symboles et des relations. Les résultats ont montré que si les gains d'étiquetage peuvent être importants pour des langages simples, le problème est plus complexe pour les langages dont les symboles comprennent beaucoup de traits. Nous avons aussi observé que si le nombre d'itérations appliquées est trop important, l'algorithme extrait des petites structures de symboles fréquents. Cette propriété pourrait être

utilisée dans un autre contexte pour faire une analyse non supervisée des structures grammaticales d'un langage. C'est d'ailleurs cette observation qui m'a donné l'idée de chercher les structures fréquemment en erreur dans les graphes LG pour définir les histogrammes d'erreurs de structures utilisés dans CROHME et illustrés par la figure 4.2.

5.3 Reconnaissance de gestes multipoints - Thèse de Z. Chen

Cette section présente les travaux de thèse en cours de Z. Chen. Il s'agit d'une collaboration avec l'équipe IntuiDoc financée par deux demi-bourses régions Pays de la Loire et Bretagne.

On constate aujourd'hui une démocratisation massive des périphériques tactiles (Smartphones, tablettes et tables tactiles...) qui ouvre un marché important pour accueillir ces technologies de composition en-ligne de documents. Cependant il y a un décalage entre les capacités des matériels (périphériques tactiles) toujours plus étendues et les approches de composition et de reconnaissance de documents structurés en-ligne qui sont restées ancrées sur une modalité d'interaction mono-point et multi-tracé qui exploite très peu le multipoint. Or certains périphériques sensitifs/tactiles autorisent aujourd'hui une interaction multipoint pouvant aller jusqu'à plus de 40 points de contact simultanés.

Il y a très peu de travaux dans l'état de l'art proposant des systèmes de reconnaissance spécifiques aux gestes multipoints (citons quand même [100, 101]). Ils se focalisent plus sur l'impact de ce type de gestes sur les usages dans les interfaces homme-machine et donc les gestes utilisés restent simples et classiques : zoom, clic, déplacement à deux doigts, formes géométriques simples... Comme expliqué en introduction, nous sommes dans le cadre d'un signal en-ligne mais cette fois la reconnaissance doit se faire à la volée.

Nous nous sommes intéressés à deux aspects du problème des gestes tactiles multipoints : la reconnaissance des gestes isolés et la segmentation des gestes dans un contexte multi-utilisateurs.

Pour les gestes multipoints isolés nous avons commencé par créer une base de gestes en considérant les différents types de geste et les différentes situations que permettent les gestes multipoints. Nous pouvons notamment distinguer les gestes de commandes directes qui nécessitent d'être reconnus le plus tôt possible pour permettre un retour visuel à l'utilisateur (e.g. le zoom ou la rotation) et les gestes de commandes indirectes qui sont reconnus à la fin du geste. Nous avons aussi choisi des gestes qui illustrent l'importance de la dynamique du tracé, des gestes

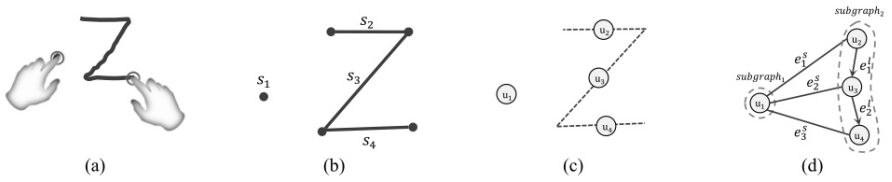


FIGURE 5.2 – Construction du graphe représentant un geste multipoint. (a) le geste original (b) interpolation polynomiale, (c) création des nœuds de segments, (d) ajout des arcs de relations spatiales et temporelles.

avec la même forme finale (par exemple un double trait '=') peuvent avoir des significations différentes s'ils sont faits simultanément, avec des débuts décalés ou l'un après l'autre.

La reconnaissance des gestes doit prendre en compte cette spécificité de la dynamique du tracé dans la modélisation. Pour cela nous avons choisi de représenter les gestes comme des petits documents structurés par des graphes attribués pour représenter les informations de forme des traits, de relations spatiales entre ces traits et leurs relations temporelles. La figure 5.2 illustre la transformation d'un geste en graphe. Pour la classification des gestes, nous avons choisi d'utiliser le principe du *graph embedding* [102] : un ensemble de graphes prototypes est sélectionné, les distances entre le graphe à classer et ces prototypes servent de caractéristiques pour un classifieur classique (un SVM pour nous). Ces travaux ont fait l'objet de deux publications [19, 18] qui ont montré les très bonnes performances de l'approche qui est entièrement entraînable et ne nécessite que très peu d'a priori sur le type de gestes à reconnaître.

Au delà de ce classifieur de gestes nous nous sommes intéressés à la reconnaissance des commandes directes, c'est-à-dire avec une reconnaissance anticipée dès le début du geste. Pour cela nous avons défini trois classifieurs pour reconnaître les débuts de geste, les gestes à mi parcours et les gestes terminés. En dotant ces classifieurs de capacités de rejet et avec un entraînement adapté, nous sommes capable de reconnaître les gestes au plus tôt. Les capacités de rejet entraînées avec des gestes réels sont importantes car le système doit pouvoir détecter les conflits possibles entre certains gestes dont les débuts se ressemblent. Cette approche très originale par rapport à l'existant est en cours de publication (soumission à ICPR2016).

Les travaux en cours sur la fin de la thèse concernent la reconnaissance des gestes dans un contexte multi-utilisateur. Dans les systèmes industriels, ce problème est pour l'instant résolu soit en considérant que les utilisateurs sont forcément éloignés soit en n'autorisant qu'un seul point d'interaction pour chacun. Si l'on veut autoriser les utilisateurs à faire des gestes multi-traits et multipoints en

même temps et sans contraintes spatiales, on retrouve un problème de segmentation comme ceux présentés dans les chapitres précédents mais dynamique. Après avoir proposé un scénario de saisie de document impliquant deux utilisateurs pour collecter des données réelles, nous sommes en train de proposer un système permettant de distinguer les gestes terminés (qui nécessitent un retour visuel) des gestes encore en cours de construction quel que soit l'utilisateur.

Ces travaux en pointe dans le domaine des Interactions Homme-Machine se font avec le soutien de l'entreprise *excence*[1] dans le cadre d'un contrat de transfert industriel. *excence* est spécialisée dans le développement d'interfaces tactiles multipoints sur mesure pour la relation client. Le cadre applicatif de cette collaboration est donc la mise au point d'une interface permettant à un client et un vendeur d'interagir en même temps sur un document visionné sur une table tactile.

1. http://www.excense.fr/fr/

Chapitre 6

Perspectives et projets de recherche

La diversité des domaines de recherche que j'aborde et de leurs applications sont pour moi une source de motivation. Ainsi, le traitement d'images et la reconnaissance des formes couvrent tellement de domaines, dans tant de contextes (interaction homme-machine, écriture en-ligne, analyse de documents anciens, vision industrielle) qu'ils sont une source inépuisable pour toujours avoir des champs nouveaux à défricher et de nouvelles questions à poser. Les outils scientifiques fondamentaux mis en jeu sont également très variés (algorithmes d'apprentissage automatique, modèles de langages, algorithmique des graphes, évaluation de performances). D'une manière générale je souhaite conserver cette richesse qui, j'en suis persuadé, est source de créativité. La recherche par projets est un moyen d'approfondir des objectifs scientifiques bien définis en amont avec des partenaires complémentaires. Par ailleurs, il est important de se laisser du temps et des ressources pour essayer de nouvelles choses au fil des rencontres et des occasions. Par exemple, les compétitions CROHME ont permis la structuration de la communauté, la création de nombreuses collaborations et le développement d'outils maintenant indispensables, mais pourtant ce type de travaux est difficile à financer par des projets.

Localement, mes projets de recherche se placent clairement dans le *RFI numérique Atlanstic2020*. En effet les projets proposés ci dessous correspondent à trois des interfaces interdisciplinaires privilégiées par ce RFI : les sciences humaines et sociales, l'usine du futur et pour la pédagogie les objets connectés. Néanmoins je pense que ces projets peuvent aussi être développés ailleurs en dehors de ces supports.

Les deux premiers projets présentés, "Analyse de documents anciens" et "Vision industrielle", sont dans la continuité directe de mes projets en cours. Les deux

sections suivantes décrivent des axes de recherche plus transversaux et à plus long terme sur l'utilisation de l'apprentissage automatique dans les documents structurés (section 6.3) et sur la prise en compte du contexte d'un document pour sa reconnaissance (section 6.4). Les deux dernières sections concernent mon Positionnement national et international ainsi que quelques pistes de Projets pédagogiques en liens avec mes travaux de recherche.

6.1 Analyse de documents anciens

La BNF, comme d'autres centres d'archives, numérise ses fonds documentaires, à un rythme de plus de 100 000 ouvrages par an. [1] Plus localement, 15 000 pages écrites par Jules Verne sont consultables à la Bibliothèque Municipale de Nantes. [2] La consultation de ces documents se fait par des plateformes comme Gallica et une partie des documents imprimés peut être indexée grâce à des outils d'OCR. Néanmoins l'indexation et la reconnaissance des documents manuscrits restent des tâches de recherche. L'accès à ces fonds culturels est un des verrous de la culture numérique et de la société numérique. La consultation efficace de ces documents sera aussi une opportunité pour nos collègues des Sciences Humaines et Sociales pour consolider leurs hypothèses de travail.

Depuis 2015 nous avons démarré le projet CIRESFI. Il s'agit d'un projet ANR SHS basé sur l'utilisation d'un corpus de documents du XVIII$^{\text{ème}}$ siècle récemment numérisés. Si les partenaires littéraires analysent manuellement le contenu de ces documents pour mieux comprendre la société de l'époque, notre rôle est de proposer un outil d'analyse automatique pour accélérer cette recherche. Notre principale contribution s'est concrétisée en 2015 par le démarrage de la Thèse de A. Granet en collaboration avec l'équipe TALN du LINA. L'objectif de cette thèse est d'utiliser les quelques ressources textuelles disponibles pour aider l'analyse des documents grâce aux outils de traitement automatique du langage naturel. Concrètement il n'y a pas de vérité terrain au niveau ligne ou mot de disponible et les annotations utilisables n'ont pas été saisies pour permettre un apprentissage mais pour être analysées par les chercheurs SHS (transcription de textes, listes de titres, statistiques sur les occurrences de certains mots,...). L'analyse des documents en elle même se fera avec les outils utilisés dans les applications présentées précédemment : approches grammaticales (CFG ou DMOS [95]) et par réseaux de neurones (CNN, BLSTM [103],...). L'utilisation des outils du traitement du langage naturel doit

1. http://www.bnf.fr/fr/collections_et_services/anx_bib_num/a.numerisation_masse_bnf.html
2. http://www.julesverne.nantesmetropole.fr/home/se-documenter/manuscrits-numerises.html

nous permettre d'utiliser au mieux les ressources disponibles pour guider l'analyse et la reconnaissance des images de documents.

Au delà de ce premier projet sur cette thématique, j'aimerais m'intéresser aux problématiques d'apprentissage dans ces grands corpus non annotés. Citons quelques défis parmi les nombreux restants : l'absence de vérité terrain rend nécessaire l'utilisation d'algorithmes d'apprentissage non supervisés ou semi-supervisés ; le vocabulaire utilisé dans ce type de document n'est pas toujours disponible de façon complète, mais il existe souvent d'autres sources de textes qui peuvent aider (lien avec le TAL) ; certains documents ont une structure qui se répète localement de pages en pages et cette régularité peut être utilisée à profit, mais il faut aussi être capable de détecter les modifications de mise en page ainsi que les pages exceptionnellement différentes qui sont souvent les plus intéressantes.

Pour aborder ces problématiques deux aspects sont à prendre en compte. Tout d'abord je pense qu'il faut réunir les ressources des différents laboratoires pour permettre d'accélérer les étapes d'analyse de base comme cela s'est fait dans d'autres communautés (reconnaissance de la parole, TALN...). Cette mutualisation est un objectif ambitieux et nécessite une adhésion forte de la communauté mais permettra de réduire le coût de démarrage de nouveaux projets similaires. Ensuite il faut constituer des équipes-projets interdisciplinaires réunissant des compétences en analyse de documents, en apprentissage (notamment pour les réseaux profonds), en TAL et en SHS.

6.2 Vision industrielle

L'usine du futur et la robotique industrielle passeront forcément par plus de vision industrielle. L'œil est un outil indispensable à la machine pour lui permettre une meilleure adaptabilité pour par exemple le contrôle ou le guidage. Les solutions de vision existantes en milieu industriel sont créées spécifiquement pour chaque nouvelle tâche nécessaire, ce qui est coûteux en temps de mise en place de nouvelles chaînes de production. De récentes publications comme [104] montrent l'intérêt de l'apprentissage automatique pour ces problèmes de vision. Certains aspects de la vision industrielle sont assez proches des problématiques de l'analyse de documents et je pense que ces deux domaines de recherche partageant l'image comme point commun peuvent s'alimenter l'un l'autre en solutions originales.

Nous avons récemment commencé un transfert industriel avec l'entreprise Multitude Technologies qui se concrétise par la thèse CIFRE de J. Langlois commencée en 2016. Cette entreprise de production de pièces en plastique (principalement pour l'automobile) cherche à développer son expertise en vision industrielle pour automatiser certaines tâches complexes. Une tâche typique est le dévracage volumique

(*bin-picking* en anglais), elle consiste à sortir les pièces une à une d'un bac de pièces homogènes. La tâche analogue en analyse de documents serait le *word-spotting* : retrouver toutes les occurrences d'un mot spécifique dans un grand corpus. La différence est qu'une pièce dans un bac a beaucoup plus de degré de liberté qu'un mot sur une page, mais les occurrences d'un mot manuscrit sont plus variables que celles d'une pièce industrielle. Nous proposons une approche à base de réseaux de neurones profonds pour localiser et qualifier l'orientation des pièces préhensibles par un bras robotisé. Le milieu industriel exige une performance importante, en contrepartie les conditions sont mieux contrôlées que dans les documents manuscrits. Par exemple, nous pourrons utiliser à profit l'existence de modèles 3D des pièces à détecter pour accélérer l'apprentissage des réseaux.

Pour développer à long terme cet axe de recherche, il faut que nous nous rapprochions des collègues en robotique pour un continuum cybernétique-informatique comme recherché à travers la constitution des RFI, notamment le RFI Atlanstic 2020, mais aussi avec l'avènement du LS2N (Laboratoire des Sciences du Numérique de Nantes). Cette collaboration pourrait par exemple permettre de piloter directement les robots en prédisant la séquence d'opérations à réaliser par des réseaux récurrents plutôt que de juste définir un objectif.

6.3 Apprentissage et réseaux de neurones

Proposer des systèmes qui peuvent être appris est un gage de généricité. Nous pouvons voir que les réseaux de neurones ont fait leurs preuves dans bien des domaines. Les nouvelles architectures de types réseaux récurrents (BLSTM par exemple) et réseaux profonds sont en train de booster des domaines jusqu'alors traités par une succession de processus : l'analyse de scènes naturelles, la vision industrielle, la reconnaissance de l'écriture ou de la parole, le traitement automatique du langage naturel (voir le projet RAPACE récemment accepté dans le RFI Atlanstic 2020)...

La reconnaissance de l'écriture et des documents structurés ne sont pas une simple application de ces outils de pointe. Ces solutions doivent être adaptées car il y a une imbrication fine du domaine et de l'outil. Les documents structurés ne sont pas de simples images contenant des éléments indépendants, les relations entre ces éléments doivent être reconnues pour construire une interprétation. Nous devons proposer des algorithmes de reconnaissance et d'apprentissage spécifiques mais aussi des topologies de réseaux de neurones adaptées à notre contexte. Par exemple, si les réseaux de neurones sont maintenant capables de répondre avec une carte 2D pour la segmentation de scènes naturelles [105] ou par des séquences en traitement du langage [106], il n'existe par pour l'instant de solution pour générer

un graphe étiqueté représentant un document. La Thèse de T. Zhang est un premier pas dans cette direction puisque nous utilisons un reconnaisseur de séquences pour générer des graphes représentant les expressions. À terme je compte proposer une architecture neuronale dont la topologie s'adapte à la topologie des données (traits ou composantes connexes). L'idée est de générer directement un graphe relationnel étiqueté par un seul réseau et non d'estimer chaque étiquette indépendamment.

La méthodologie de l'apprentissage en lui-même doit aussi évoluer. Nous avons montré dans la Thèse de M. Awal l'intérêt de faire une boucle de rétro-action reconnaissance-apprentissage grâce à l'apprentissage global. Je pense que les techniques d'apprentissage sont maintenant assez mûres pour donner plus d'autonomie aux réseaux de neurones pour une tâche comme la reconnaissance d'écriture. En effet, comme un enfant qui apprend à lire et à écrire en même temps, nous pourrions fusionner un système de génération d'écriture et de reconnaissance d'écriture et proposer un système capable de l'un et de l'autre. Les travaux sur les réseaux de neurones profonds ont déjà montré l'efficacité du partage de poids (*transfert learning*[107]) entre des réseaux dédiés à des tâches connexes mais différentes.

Pour rendre pérenne cette activité de recherche sur l'apprentissage, il faudra investir sur deux axes : tout d'abord sur les plateformes logicielles existantes (Theano/Lasagne, TensorFlow...) et en parallèle mettre en place une plateforme matérielle rendant disponibles des capacités de calcul sur GPU importantes. Ces plateformes permettront de répondre plus rapidement à des besoins interdisciplinaires voir à des transferts industriels par exemple à travers les cellules de valorisation universitaires.

6.4 Prise en compte du contexte dans la reconnaissance

Grâce au cloud et au big data, les systèmes ne sont plus isolés mais ils peuvent avoir accès à l'information étendue. Comme nous l'avons montré dans nos travaux sur la reconnaissance d'expressions multimodales, les compléments d'informations peuvent être une contribution importante à la reconnaissance. En fonction du domaine d'application ces sources peuvent être de différentes natures et avec une intégration différente dans le système de reconnaissance.

Cet axe de recherche est très riche car, par définition, à la frontière de différents domaines de recherche : reconnaissance de la parole, TALN, analyse d'images mais aussi les SHS ou les IHM. De plus il est promis à un bel avenir industriel. En effet à l'heure du big data, les projets industriels tirant parti de ces nouvelles sources

d'informations pour en extraire des services (par exemple l'entreprise *Dictanova*[3]) sont florissants. Si ces services utilisent pour l'instant des sources monomodales (du texte, des musiques, des images...), ils iront bientôt sur des fusions de modalités en enrichissant par exemple des textes avec des informations venant d'images.

Le contexte peut être pris en compte à plusieurs échelles suivant le type de problème et l'architecture du système proposé. Nous avons déjà exploré ces différentes échelles dans le cadre de la reconnaissance des expressions mathématiques (Thèse de S. Medjkoune et Thèse de F. D. Julca-Aguilar) mais je pense que cette idée peut être développée pour d'autres applicatifs. Pour la reconnaissance de textes manuscrits anciens il est possible d'utiliser des transcriptions disponibles de documents connexes au contexte (comme dans le projet CIRESFI où la liste des pièces jouées existe). La difficulté est alors de choisir le bon niveau de prise en compte des informations de contexte (caractère, mot, ligne,...).

Dans le domaine de l'Interaction Homme-Machine comme dans la Thèse de Z. Chen, le document est le contexte dans lequel sont fait les gestes, contexte qui peut guider la reconnaissance des gestes d'interaction. Dans ce type d'applicatif, c'est classiquement un post-traitement qui permet de désambiguïser le résultat en fonction du contexte ; nous pourrions essayer d'ajouter aux entrées du classifieur des informations modélisant le contexte. Les MOOC tels que ceux du projet *CominOpenCourseware*[4] sont des documents multi-modaux (vidéos avec son) mais en plus, ils sont annotés par les utilisateur par une lecture active. La complémentarité des modalités ainsi que les informations supplémentaires des annotations peuvent participer à mieux reconnaitre, interpréter, indexer ou segmenter leur contenu (images des vidéos, parole du présentateur, texte, équations ou schémas des transparents...).

6.5 Positionnement national et international

Il y a en France de nombreux chercheurs sur la thématique de l'analyse des documents hors-ligne, moins sur les documents en-ligne, et nous avons la chance de nous être organisés dans le "Groupe de Recherche en Communication Écrite".[5] Je pense que nous avons de nombreux défis à relever, tant sur le plan scientifique que sur le plan du financement de nos projets. Pour cette raison, je compte participer à cette structuration de la recherche localement, régionalement et nationalement, non seulement en répondant aux appels à projets, mais aussi en collaborant à la

3. http://www.dictanova.com/ Dictanova propose un logiciel d'analyse de l'expérience client se basant sur leurs commentaires en utilisant les outils du TALN.
4. Projet financé par COMINLabs et Région PDL https://comin-ocw.org/
5. Association GRCE : http://grce.labri.fr/

créations de ressources communes : bases annotées, bibliothèques de code, formats de fichiers communs, ...

Je pense, comme cela a déjà été le cas dans le passé avec quelques réussites emblématiques, que nous contribuons à mettre en place un terreau pour une recherche innovante qui permet à nos doctorants et aux start-up régionales, voire aux plus grandes structures, un épanouissement réciproque (comme chez MyScript, Dictanova, CEA, Aridnext... qui accueillent nous anciens doctorants). D'autres opportunités au carrefour du Machine Learning, du Traitement Automatique du Langage Naturel et de la gestion des connaissances vont apparaitre. Je suis résolu à jouer un rôle clé pour favoriser ces émergences. Au niveau régional, les différents RFI mis en place participent à cette structuration. Osons aller encore plus loin, imaginons une meilleure intégration de la recherche académique et industrielle avec des laboratoires communs partageant des projets de recherche en amont et des plateformes expérimentales.

Au niveau international, nous avons réussi à créer une synergie autour de la reconnaissance d'expressions mathématiques grâce aux compétitions CROHME. Il faut que nous conservions ce leadership en poursuivant nos efforts de structuration par des collaborations, des compétitions et du partage de ressources logicielles, de données d'apprentissage et d'évaluation mais aussi par du partage de systèmes complets fonctionnels et ouverts. Concrètement, lors des prochaines compétitions CROHME, je proposerai un système complet opensource dans lequel les participants pourront remplacer une brique générique par leur contribution sur laquelle ils sont spécialistes.

6.6 Projets pédagogiques

Je souhaiterais terminer cette liste de perspectives en rappelant que je suis aussi un enseignant. À travers mes enseignements à l'IUT, mes interventions en master et dans l'école doctorale, j'ai la chance de rencontrer des étudiants de tous les niveaux de formation, chacun ayant ses spécificités. L'encadrement de doctorants est aussi un acte pédagogique puisqu'il faut former le futur chercheur.

Le département GEII de l'IUT où j'interviens est par définition une source de diversité thématique intéressante. Mon thème de recherche se prête bien à la pédagogie par projet car il pique souvent la curiosité des étudiants. Nous organisons donc régulièrement des projets autour du traitement d'images ou de la reconnaissance de formes. Les compétences en informatique et électronique embarquées des étudiants complétées de mes conseils en vision et apprentissage pourraient permettre de réaliser par exemple des prototypes d'objets connectés intelligents. Toujours à l'échelle de l'IUT je voudrais continuer ma participation à la réflexion sur

l'usage des nouveaux outils pédagogiques (Pédagogie par projets, classe inversée, MOOC, TBI).

Sur le principe des RFI, il faut conserver un lien fort entre formation et recherche. Il sera pertinent de profiter des plateformes mises en place autour de l'apprentissage des réseaux de neurones pour intégrer ces outils aux niveaux master et école doctorale. Ces outils semblent souvent inaccessibles aux étudiants des autres spécialités, c'est pourquoi les formations transversales doivent, en plus des parties théoriques, contenir des modules pratiques d'utilisation de l'intelligence artificielle et d'analyse des résultats. Ces modules renforceront les plateformes, permettront de répondre à des besoins industriels croissants en formant des étudiants à ces technologies et créeront des dynamiques interdisciplinaires autour de l'apprentissage. Pour favoriser ces émergences, il faut continuer nos efforts de formation par la recherche en proposant des cursus intégrant des expertises multimodales, comme le parcours ATAL du master informatique de l'Université de Nantes.

Chapitre 7

Sélection d'articles

7.1 A global learning approach for an online handwritten mathematical expression recognition system 80
7.2 Text Alignment from Bimodal Mathematical Expression Sources 90
7.3 Advancing the state of the art for handwritten math recognition : the CROHME competitions 2011–2014 . 95
7.4 An annotation assistance system using an unsupervised codebook composed of handwritten graphical multi-stroke symbols 112

Pattern Recognition Letters 35 (2014) 68–77

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A global learning approach for an online handwritten mathematical expression recognition system

Ahmad-Montaser Awal [a,*], Harold Mouchère [b], Christian Viard-Gaudin [b]

[a] Laboratoire d'étude des Mécanismes Cognitifs, Université Lumière Lyon2, Lyon, France
[b] LUNAM Université, Université de Nantes, IRCCyN/IVC, Nantes, France

ARTICLE INFO

Article history:
Available online 16 November 2012

Keywords:
Handwriting recognition
Bidimensional languages
Math recognition
Structural pattern recognition
Syntactic pattern recognition

ABSTRACT

Despite the recent advances in handwriting recognition, handwritten two-dimensional (2D) languages are still a challenge. Electrical schemas, chemical equations and mathematical expressions (MEs) are examples of such 2D languages. In this case, the recognition problem is particularly difficult due to the two dimensional layout of the language. This paper presents an online handwritten mathematical expression recognition system that handles mathematical expression recognition as a simultaneous optimization of expression segmentation, symbol recognition, and 2D structure recognition under the restriction of a mathematical expression grammar. The originality of the approach is a global strategy allowing learning mathematical symbols and spatial relations directly from complete expressions. A new contextual modeling is proposed for combining syntactic and structural information. Those models are used to find the most likely combination of segmentation/recognition hypotheses proposed by a 2D segmentation scheme. Thus, models are based on structural information concerning the symbol layout. The system is tested with a new public database of mathematical expressions which was used in the CHROME competition. We have also produced a large base of semi-synthetic expressions which are used to train and test the global learning approach. We obtain very promising results on both synthetic and real expressions databases, as well as in the recent CHROME competition.

1. Introduction

Since the emergence of new technologies such as digital pens, tablets, smartphones, etc., digital documents are used increasingly. Nevertheless, scientific documents are full of diagrams and equations. Those notations are indispensable for describing problems and theories using common universal languages. The main property of these languages is their two dimensional nature, where symbols are organized in a two-dimensional (2D) space. Among others, mathematical notation is a very important 2D language because it is used in almost all sciences.

In order to take advantage of pen-based technologies, it is necessary to build systems allowing the transfer of the handwritten traces (physical form) to a digital text (logical form). The stochastic nature of handwriting and the variability in writing styles make achieving this task quite difficult. Handwriting recognition has been an active domain of research since the 60s of the last century. Text recognition systems have known an important development in the last few years (Plamondon and Srihari, 2000).

However, these systems are limited to the recognition of characters organized in a sequence of words belonging to a given vocabulary. The spatial arrangement of these constituents is the horizontal line (or vertical in some Asian languages). This is not the case when the grammar controlling the language is itself of a 2D nature. This is particularly the case for MEs (Chan and Yeung, 2000a,b), schemas (Feng et al., 2009), diagrams (Yuan et al., 2008), tables (Coüasnon, 2001), chemical equations (Wang et al., 2009), musical scores (Szwoch, 2007), characters of some languages such as Chinese (Delaye et al., 2009), etc.

Many tools support the input of MEs into digital documents, but special skills are required to use them properly. For example, LaTeX and MathML require the knowledge of predefined keywords. Other tools, such as MathType or Microsoft equations, depend on a visual environment for adding mathematical symbols using the mouse. Those dependencies on the mouse or the keyboard increase significantly the time required to input an expression vs. drawing an expression.

In this paper, we will focus on the recognition of online handwritten MEs. In the next section we will present some related works. Section 3 describes how MEs are represented. The proposed ME recognition system (MEXREC) is then presented; we will particularly focus on our contribution with the global learning scheme, and the spatial relation modeling. Unlike existing systems,

* Corresponding author.
E-mail addresses: ahmad-montaser.awal@univ-lyon2.fr (A.-M. Awal), harold.mouchere@univ-nantes.fr (H. Mouchère), christian.viard-gaudin@univ-nantes.fr (C. Viard-Gaudin).

0167-8655/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2012.10.024

the proposed system is fully configured and trained directly from complete mathematical expressions, with no constraints related to stroke time order. Finally, we present some results in Section 6.

2. Related works

Math recognition has been an active research area since the late 60s, first works were dedicated to the offline recognition of typed expressions (Anderson, 1968; Chang, 1970) or handwritten ones (Belaid and Haton, 1984). The process of 2D language recognition is mainly based on three sequential steps: segmentation, symbol classification and interpretation (structural and syntactic analysis). However, global approaches have been introduced quite recently to solve this problem by applying jointly these three steps.

Being able to segment the 2D ink-traces into its basic symbols is a very important stake. In an online signal we can impose the condition of lifting the stylus when moving from one symbol to the next one. This condition is readily acceptable only in some languages, like MEs (Rhee and Kim, 2009), chemical equations (Wang et al., 2009) and flowcharts (Yuan et al., 2008). However, this assumption does not solve the segmentation problem, which consists in grouping strokes belonging to the same symbol.

A ***stroke*** is the sequence of points between a pen-down and a pen-up of a stylus. The stroke is our basic unit and we assume that a stroke belongs only to one symbol. However, a symbol can be written with one or more strokes, which are not necessarily sequential. Existing works do not support this latter hypothesis, since interspersed symbols increase considerably the complexity of the segmentation process. Most of existing works consider that all strokes are present before starting the segmentation and recognition, taking into account the global context. Other approaches consider processing the strokes on the fly as they are input and though provide updated results for each new stroke, allowing a direct feedback to the user.

The number of possible segmentations for a set of strokes is defined by the Bell number (Eq. 1). For example, considering the presence of a set of seven strokes, we obtain $B_7 = 877$ distinct segmentations.

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k; \text{ with } B_0 = B_1 = 1 \tag{1}$$

Not all of these segmentations have to be considered, as time order and spatial information can be used to prune the segmentation space. A bottom-up analysis is applied in Dimitriadis et al., 1991 using horizontal/vertical projections. Similarly, bounding boxes can be used to group strokes (Chan and Yeung, 2001). Other spatial constraints have also been used; a new stroke is judged belonging (or not) to the same symbol of the previous one based on a distance measurement (Tapia and Rojas, 2003). Lehmberg et al. (1996) used the strokes' properties in addition to spatial information. The probability decides if a stroke must be grouped with one or more (up to three) of the following strokes. This method is very sensitive to stroke temporal order.

This last approach evolved to what is called "simultaneous segmentation and recognition", where the main criterion to group strokes is the probability of those strokes representing a given symbol. We will briefly explore possible classification approaches usually used in 2D language recognition.

The goal of the classification step is to identify the objects found during the segmentation process. Symbol classification is quite complicated in the case of mathematical symbols because of the large number of classes and their overlapping. In fact, when looking at different corpus of mathematical expressions (Raman, 1994; Garain and Chaudhuri, 2004), we can notice that more than 220 symbol classes are encountered. This number becomes much bigger if the fonts have to be detected. A large variety of classifiers have been used in 2D languages recognition systems: template matching (Rhee and Kim, 2009), structural approaches (Chan and Yeung, 2000a,b), K nearest neighbor (Prusa and Hlavac, 2007), support vector machines (SVM) (Keshari and Watt, 2007), fuzzy logic based classifiers (Macé and Anquetil, 2009) and neural networks (Dimitriadis and Coronado, 1995).

All those classifiers require a prior segmentation, i.e., having the symbols extracted first. Yet, some techniques such as hidden Markov models (HMM) (Kosmala et al., 1999) can perform segmentation and recognition, simultaneously. Another approach consists in proposing all possible segmentations and associating a cost with each candidate segmentation. Finally, the segmentation with the minimal cost is chosen (Fukuda et al., 1999). This last approach becomes more feasible thanks to dynamic programming techniques that allow an efficient exploration of the search space.

After identifying the symbols, the next stage of recognition consists in finding the physical and logical structures of the expression.

Structural analysis consists in finding the spatial relations between the symbols based on symbol structural information. Depending only on bounding boxes as a source of this information (Ha et al., 1995) might cause some ambiguities, but structural information could be based on symbol typographical centers (Zanibbi and Blostein, 2002), or even on bounding boxes with baselines adapted to the type of the symbol (Eto and Suzuki, 2001; Mitra et al., 2003).

When all the symbols are correctly recognized, it is still necessary to analyze the expression's 2D structure to produce the final output, for example to differentiate "2x" from "2^x". The intuitive way of defining a spatial relation is defining regions around each symbol (Lavirotte and Pottier, 1998; Zanibbi and Blostein, 2002). However, spatial relations are of a fuzzy nature. Thus, the use of fuzzy rules is very appropriate for such analysis (Zhang et al., 2005; Fitzgerald et al., 2006, 2007). Aly et al. proposed a method based on the distribution of certain relation features (Aly et al., 2009) using a training database. A normalized distribution map is constructed using relational spatial information (Eto and Suzuki, 2001). Each relation is then associated to a probability obtained from the estimated distribution. We will propose a similar statistical approach to learn spatial relation based on Gaussian distributions.

Syntactic analysis is usually the last step in expression recognition. The objective of this step can be summarized in the following three points:

- Assure the grammatical correctness of the recognized expression.
- Produce the derivation tree of the expression which is then easily transformable to a standard presentation format such as LaTeX or MathML.
- And, more importantly, use the global context in order to resolve the local ambiguities.

A context free grammar (CFG) can be perfectly used to produce a subset of mathematical expressions. CFGs have been efficiently applied to analyze 1D formal languages such as programming languages. However, analyzing 2D languages requires special algorithms and constraints in order to reduce the complexity (Miller and Viola, 1998). Two principal approaches have been investigated in the literature: grammar or graph based analysis.

Since 1D parsers are more efficient than bi-dimensional ones, (Tokuyasuy and Chou, 1999) proposed to apply iteratively syntactic analysis along horizontal and vertical axes respectively using a stochastic context free grammar. (Chan and Yeung, 2000a,b) proposed to transform the expression from its 2D form to a uni-

dimensional one, and to analyze it using a "define clause grammar" (DCG). Based on structural information extracted from the expression, grammatical analysis could be applied using CFG (Garain and Chaudhuri, 2004). Probabilistic grammar is proposed in (Yamamoto et al., 2006), where each rule of the grammar is associated to a structural constraint and probability reflecting the certainty of its logical relation. Similarly, a fuzzy online structural analysis algorithm is proposed in (Fitzgerald et al., 2006) in order to cope with the nature of spatial relations. More recently, (Scott and George, 2010) proposed to associate each production rule of a context free grammar to a fuzzy function. We will propose a similar approach by associating each production rule to a Gaussian model specific to each spatial relation as we will see in Section 5.2.

Graph rewriting can also be used for syntactic analysis. The structure of the expression is represented by a graph. Rewriting consists in replacing a sub-graph by a single node containing the syntax of the sub-expression (Grbavec and Blostein, 1995; Kosmala et al., 1999). However, the calculation time of graph rewriting is long.

Classically, previous steps are applied sequentially. As a result, an error occurring in one step would be inherited by subsequent steps. Furthermore, some local ambiguities require the whole context to be resolved. This led to an approach based on a simultaneous application of all the steps, or what is called a "global approach" introduced by Lecun et al. (1998) which is more and more used recently. Indeed, the main steps are still the same but performed jointly in a global framework to recognize at a time the entire expression.

Yamamoto et al. (2006) proposed to model the whole recognition process by a stochastic context free grammar. The grammar takes into account the writing order and the 2D nature of symbols. The first type of production rules controls the production of symbols from the input strokes. Each of those rules is probabilistic, i.e., when a rule is applied a probability is directly calculated from the symbol classification module. The rest of production rules models the spatial relations and the syntax by calculating a structural probability using the bounding boxes. The CYK algorithm (Cock–Younger–Kasami) is used to find the most probable expression. The main disadvantage of this algorithm is its dependency with respect to the temporal order of strokes. As a result, the user must input strokes in a correct order or pre-processing methods must be applied.

In order to overcome the temporal order problem, Rhee et al. proposed a layer search framework (Rhee and Kim, 2009). The recognition problem is reformulated as a search of the most probable interpretation of a given stroke set. The expression structure is extended by adding symbol hypotheses, representing the different identity of symbols, and at each structural ambiguity a new branch is added, creating a search tree. The search of the most likely solution is carried out by the "best first search" algorithm.

Other examples of global approaches can be found in (Prusa and Hlavac, 2007; Shi et al., 2007). Regardless of the algorithm, a global based approach is generally formulated as an optimization of a global cost function.

Precisely, our system is based on a global approach methodology by simultaneously applying the main recognition steps (symbol segmentation/recognition and structure/syntactic analysis). The proposed architecture tries to break up the limits of existing methods by taking into consideration the following points. First of all, it will allow handling several segmentations, several recognitions, and several logical relationships to select the best possible interpretation of the input strokes. Thus, this will avoid any local ambiguity and take advantage of the global context. Secondly, differently from most proposed methods, strokes will be treated not only temporarily but also spatially. This will easily allow dealing with delayed strokes, such the "*i*" point, or even transforming a '=' symbol in a '≠' after the whole expression is written. Another distinctive feature of the system is its capacity to train the symbol classifier and the spatial relational model directly in the global context of the entire expressions, and that is what we will call "global learning". The classifier is referred to as a "global classifier" which has a reject capacity allowing it to discard wrong segmentations.

3. Representation of mathematical expressions

Before introducing the MEXREC system in Section 4, it is important to understand how a ME is usually represented. Two main families of trees are usually encountered to describe the same structural information. The symbol relational tree (SRT) presents the spatial (or logical) relations between symbols (or sub expressions) of the expression (Geneo et al., 2006; Rhee and Kim, 2009). On the other hand a baseline structure tree (BST) captures the structure of an expression by representing the relation between the symbols' baselines (Tapia and Rojas, 2005; Zanibbi and Blostein, 2002). Operator trees are another kind of description for MEs, but they are more concerned by the semantic of the ME than its layout.

We propose a variation of the SRT as follows. A relational tree is constructed by a 2D context free grammar allowing verifying the syntactic correctness of the expression.

A non-terminal node (NT) contains a sub-expression (SE) (the root contains the proposed solution), it is described by a set of strokes and the corresponding label string, produced by the combination of sub-expressions linked by a spatial relation R. The cost of a non-terminal node (including the root), called structural cost, $C_{struct}(R|SE)$ is the cost that two or more sub-expressions (SE_i) are in relationship by R to build a bigger sub-expression (SE).

Each terminal node (T) contains a set of strokes produced by the hypothesis generator. For each of these nodes, the symbol classifier produces a ranked list of labels with their recognition scores $C_{reco}(sh_i)$; where sh_i is a segmentation hypothesis (see Section 5.1). The rules that produce terminals are not associated with spatial relations. Conversely, a spatial relation is associated with each rule generating non-terminal nodes. As a simple example, the following context free grammar generates candidates similar to those of Fig. 1:

$$
\begin{aligned}
&sym \leftarrow x, y, 1, 2, \ldots\\
&op \leftarrow +, -, \times, \ldots\\
&formule \leftarrow subExp\ op\ sym \quad (operator)[HorizontalRule]\\
&subExp \leftarrow subExp\ op\ sym \quad (operator)[HorizontalRule]\\
&subExp \leftarrow symsym \quad (superscript)[VerticalRule]\\
&subExp \leftarrow sym
\end{aligned}
$$

Finally, the global cost (C_E) of a candidate expression of n symbols connected by r relations has been defined as a weighted sum of the symbol recognition and structural scores associated to each node (ex: Eq. 3):

$$C_E = \sum_{i=1}^{n} C_{reco}(sh_i) + \alpha . \sum_{j=1}^{r} C_{struct}(R_j|SE_j) \tag{2}$$

The α factor is used to adapt the respective ranges of the structural and recognition costs. Alpha has been set experimentally to 0.18 using a validation dataset.

Fig. 1 shows an example of such a relational tree. In the tree of the left candidate expression, the non-terminal node '2' connects the terminals '5' and '6', that contain symbol hypotheses x and 2 respectively, by the relation R_1 ("superscript") producing the sub-expression x^2. The other non-terminal, node '1', is the root of the tree. It connects the nodes '2', '3' and '4' with the relation R_2 ("operator") producing the final result x^2-1. The global cost of this solution is:

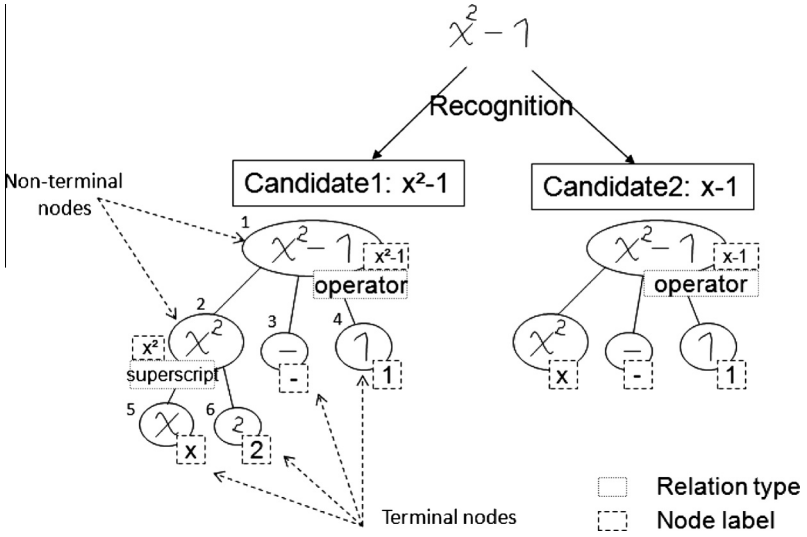


Fig. 1. Relational trees for two expression candidates.

$$C_E = C_{reco}(\text{'x'}) + C_{reco}(\text{'2'}) + C_{reco}(\text{'}-\text{'}) + C_{reco}(\text{'1'}) + \alpha.C_{struct}(R_1|x^2) + \alpha C_{struct}(R_2|x^2-1) \quad (3)$$

Fig. 1 illustrates a reduced parse tree with two solutions. Several relational trees will be built as explained in the next section by the recognition system; the one with the lowest cost is the proposed solution.

4. MEXREC: a mathematical expression recognition system

Considering a given expression as a set of strokes $E = \{s_1, s_2, \ldots, s_n\}$ representing symbols, recognizing an expression consists in finding the best possible grouping of strokes, identifying the symbol corresponding to each group, and finally interpreting the expression according to the language model. Those different steps participate in calculating the global cost function C_E. The general architecture of the MEXREC system is presented in Fig. 2.

4.1. Symbol hypothesis generator

The generator elaborates stroke combinations called a symbol hypothesis $sh \subseteq E$. However, many hypotheses are invalid due to under or over-segmentation. In the latter case, only a sub-part of a multi-stroked symbol is chosen; while in the first case strokes are grouped from two or more symbols.

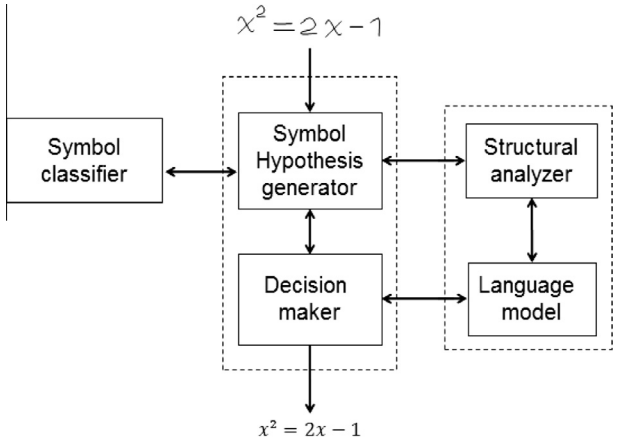


Fig. 2. System architecture.

The generator is based on an extension of a 2D dynamic programming algorithm (2D-DP) in order to allow group strokes which are not consecutive in time. This property is very important in math recognition because it is very frequent to input some delayed strokes to complete a symbol (ex: an extension of a fraction bar or a square root or transforming a symbol in another one by adding an additional stroke, such as '−' transformed in '+'). This increases exponentially the number of hypotheses according to the number of strokes. In order to control the number of hypotheses some constraints are used:

a. Maximum number of hypotheses fixed experimentally to 500
b. Maximum number of strokes per hypothesis: small symbols could be written in one stroke, while some others could reach till seven strokes (ex: *arctan*). The limit was fixed to five strokes after studying the distribution of the number of strokes within the learning symbols database (only 0.4% of symbols have more than five strokes)
c. Maximum distance between strokes: avoid grouping strokes that are far away from one another (fixed experimentally to 70% of the average diagonal length of the stroke bounding boxes).
d. Maximum number of temporal jumps: a temporal jump takes place when a symbol is completed after starting another one (delayed strokes). We allow up to two temporal jumps per symbol.

4.2. Symbol classifier

The symbol classifier associates a recognition score and a label with each symbol hypothesis. We have chosen to use a Time Delay Neural Network (TDNN) as symbol classifier. For a given hypothesis sh_i, $p(C_j|sh_i)$ denotes the probability that this hypothesis being the class C_j; with $\sum_j p(C_j|sh_i) = 1$ as a "softmax" function is applied to the TDNN outputs. Many ambiguities of mathematical symbols could be resolved at the global context level. For this reason, some methods consider the best N candidates of the symbol classifier (Yamamoto et al., 2006). Similarly, other approaches delay the decision of ambiguous symbols identity and trying to resolve it globally (Rhee and Kim, 2009). We keep the *topN* candidates with

a maximum of N ($topN \leqslant N$). The value of $topN$ is chosen to reach the condition: $\sum_{j=1}^{topN} p(C_j|sh_i) \geqslant k$. The goal of the threshold k aims at keeping only candidates with a strong confidence. In order to be easily combined with other costs (e.g. structural), the recognition score is then converted to a cost using a logarithmic function:

$$C_{reco}(sh_i) = -log(P(c = C_j|sh_i)) \tag{4}$$

Classically, a symbol classifier is used in an isolated symbol context. In this case, the input is always a unique symbol, and we expect the classifier to associate the correct class to this input. In our context, the role of the classifier is not only associating the input to a class but also providing a recognition score to each segmentation hypothesis. Thus, the classifier must have the capacity of identifying wrong segmentations and giving them high costs. In other words, we are in front of a rejection problem. In this paper, we will call the reject class the "junk class".

Among many existing methods we have explored the possibility of using a hybrid classifier containing in cascade a reject classifier and a symbol classifier (Awal et al., 2010a; Zhu et al., 2006). Another way of considering the wrong hypotheses is to add a specific class to the classifier to represent the rejection cases (Wilpon et al., 1990).

This class is added to the classification problem and must be considered during the training phase.

In this paper we use this last presented solution: include the junk class as an $N + 1$ output of the classifier.

Thus, the cost of associating the current hypothesis sh_i to the class C_j is calculated from the recognition score by:

$$C_{reco}(sh_i) = \begin{cases} Cost\_max\,; & \text{if } C_j = junk \\ -\log(P(c = C_j|sh_i)); & \text{otherwise} \end{cases} \tag{5}$$

The objective of giving a very high score to the junk hypotheses is to prevent the decision maker from choosing solutions containing theses hypotheses. A less crisp decision is possible by taking the complement of the junk probability $-log(1 - P(c = junk|sh_i))$. This allows more flexibility especially during the training phase.

4.3. Structural analyzer

Generally, structural analysis is based on the alignment and the size of symbols. So, for one symbol or sub-expression we consider its baseline position (y) and its x-height (h). These values are computed from the bounding box (BB) and depend also on the recognized symbol. For instance, for a letter with an ascender, the baseline is taken at the bottom of the BB, while the x-height is defined as one third of the height of the BB. If we consider a symbol like a 'x' without any ascender or descender, then the x-height is defined as the height of the BB.

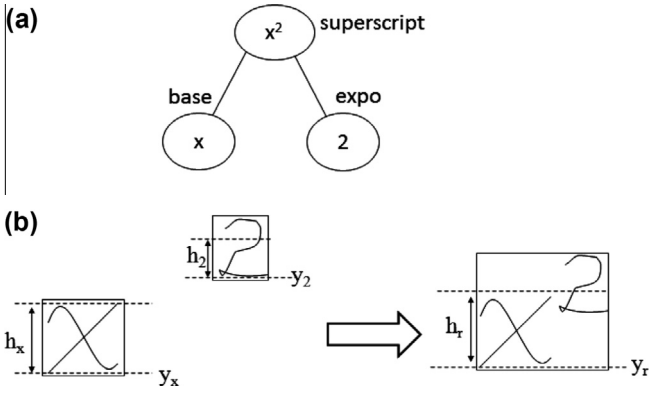


Fig. 3. Superscript relation: (a) tree representation, (b) spatial information.

A sub-expression is built from 2, 3 or 4 children depending on the type of the spatial relation. Its parameters h and y are updated according to the kind of the used relation. In the case of the superscript relation as displayed in Fig. 3, we obtain $y_r = y_x$, and $h_r = f(h_x, h_2)$. In this example, x and 2 are the children of the sub-expression x^2 by the relation "superscript" (and the sub-relations "base" and "exponent").

A structural cost is associated with each non-terminal node according to the type of the relation, using a function of the position and size of its children. An intuitive solution is to calculate a mean square error between the expected (ideal) positions and sizes for a given relation and the observed ones. However, ideal positioning of symbols within mathematical relations are difficult to define because of the fuzzy nature of those relations. In consequence, we define probabilistic costs which correspond to the matching of the observed positions and sizes with Gaussian models computed form a training dataset. The cost that a relation R produces a sub expression SE is given by the equation (see Section 5.2):

$$C_{struct}(R|SE) = -\log(p(R|SE)) \tag{6}$$

4.4. Language model

The language model is defined by a 2D grammar implemented as a combination of two 1D grammars. The first defines rules on the horizontal axis and the second on the vertical one. These rules are applied successively until reaching elementary symbols, and then a bottom-up parse (CYK) is applied to construct the relational tree of the expression. Each production rule of the grammar is associated to a spatial relation that describes the layout of elements of this rule. The application of a rule is penalized by the cost of the corresponding relation. So, each rule of this grammar is activated if its relation is more probable than other rules.

4.5. Decision maker

Finally, the decision maker selects the set of hypotheses E' that minimizes the global cost function and respects the language model using all the strokes of the input expression E.

The global cost of a candidate expression is that of the relational tree root returned by the syntax analyser $C(SE_{root})$ where the cost of a node in the tree is defined by the recursive formula:

$$C(SE_j) = \begin{cases} C_{reco}(sh_j); & \text{if } SE_j \text{ is terminal} \\ \alpha.C_{struct}(R|SE_j) + \sum_i C(SE_i); & \text{otherwise; with } \bigcup_i SE_i = SE_j \end{cases} \tag{7}$$

We will in the coming sections present the global learning schema that it uses to train the symbol classifier as well as the spatial relation learning method.

5. Global learning schema

As explained in (Lecun et al., 1998) the main idea behind the global learning is to model the recognition problem as a sequence of weighted graphs which allow taking a global decision by minimizing a derivable global cost function. This global cost function allows learning the different parts of the system thanks to gradient descent starting from the global context (the whole document). In our context of math recognition, we are close to reach this goal. As described in the previous section, we design a global cost which takes into account the recognition and the segmentation of sym-

bols (thanks to our classifier with a junk class) and the spatial relations between these symbols. The next two sections present the in-context training phases for these two main cost functions: the symbol classifier and the spatial relation analyzer.

5.1. Symbol classifier training

It is not self-evident how to learn the reject class from isolated symbols. There is not an available junk example database, where potentially the junk class represents "everything except a valid symbol". This class should represent sub-parts of symbols, combination of sub-parts of many symbols or even a combination of complete symbols. However, for a simple expression composed of seven strokes, there are 877 possible segmentations and only one is the correct one. As a result the number of junk examples obtained from thousands of expressions could reach several millions. Storing a subset of those millions could be a solution, but it is difficult to choose which examples are more suitable to represent the junk class.

This leads us to the idea of the global learning of the symbol classifier, which is done by directly using the current results of the expression recognition system. So, the classifier is trained with the expression learning database. This training is based on ground truth for the current expression, and the result of the recognition given by the symbol recognizer in its current state. Initially during this training phase the hypothesis generator is used without the language model (the grammar) and a DP algorithm finds the best segmentation and recognition of the expression's symbols. This process is repeated on all the training expressions in order to update the symbol classifier. The classifier must have the capacity for iterative learning (such as a TDNN), where each new example updates the current state of the classifier using a stochastic gradient descent. Fig. 4 illustrates different situations that require updating the classifier. In the case of the symbols ('5', '=', '8'), since the '5' and '=' are correctly recognized, they do not participate in the updating process. In contrast, the symbol '8' is wrongly recognized as a '6', so it is used to train class '8'.

Concerning the segment combining the strokes '3' and '+', it does not have a correspondence in the ground truth and thus represents a wrong segmentation that must be classified as a junk. Furthermore this wrong segmentation was recognized as a '9'. As a result, this segment is used to update the classifier to favor its recognition as a junk. Finally, the segments corresponding to '3' and '+' do not appear in the solution, meaning that probably the local recognition scores of these hypotheses were low. Consequently, they have to participate to the current corrective learning action. As a result, the gradient error which is backpropagated into the network is computed from these selected cases. This process is repeated on the whole expressions database till the convergence of the symbol classifier. This global learning schema allows many learning strategies:

- *Pure global learning*: the classifier is initialized randomly (empty classifier). Then the training expression database is used in a global learning loop to train the classifier.
- *Isolated and then global learning*: before starting the global learning, the classifier is initialized by an isolated symbol database in order to better learn the less frequent classes in the expression database.
- *Global and then isolated learning*: isolated training is done after a global learning for the same reason.
- *Isolated during global learning*: isolated symbols could be used as expressions during the global learning. In this case each symbol is considered as an expression of one symbol.

In this paper, we use only pure global learning.

5.2. Spatial relation modeling

With the *geometric approach* used to evaluate the structural costs, ideal differences of position and size were hypothesized between every components and the corresponding sub-expression obtained when using a specific relation. For instance, in a "Left–Right" (*mrow* in MathML) relationship, the difference in the baseline positions is supposed to be zero. However, for other relationships, assuming ideal values is not trivial. This is why we propose in this section to learn the cost functions from a training set containing samples of the different relations.

The model is based on the differences of the baseline position (y) and x-height (h) of a sub-expression SE_i compared with its parent SE, defined as follows:

$$dh_i = (h_{SE} - h_{SE_i})/h_{SE}; \quad dy_i = (y_{SE} - y_{SE_i})/h_{SE}$$

The differences (dh, dy) are the normalized differences of the position and the size of a child node regarding the sub-expression (independent of the expression scale). The distributions of dh and dy values of each sub-relation of a relation are then modeled by Gaussian models. For example, as displayed in Fig. 5, the relation "superscript" implies two models of size differences, $g_{base}(dh)$ is related to the size difference between the base and the composed expression, $g_{exp}(dh)$ is the difference in size between the exponent and the expression.

We can conclude that the models of the relation "superscript" imply that a base having a size similar to that of the parent is very probable (relative difference being only 0.05). Conversely, the average size of the exponent should be smaller than the global sub-expression. It is worth to note that the size of the parent is not necessary larger than that of the base child; as a result dh can have negative values as shown in Fig. 5.

Structural costs are then computed from these Gaussian models. A sub-expression SE is produced by its sub-expressions

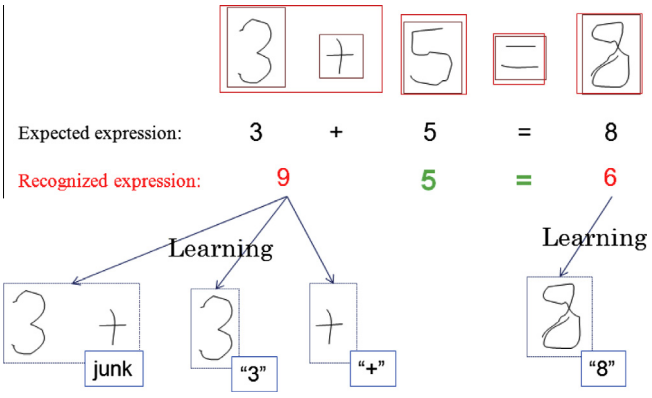


Fig. 4. Updates achieved during the global learning of the expression 3 + 5 = 8.

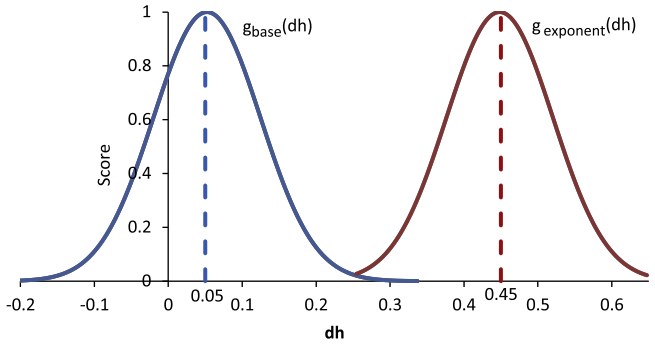


Fig. 5. Gaussian models of the size difference for the "superscript" relation.

$\{SE_1, \ldots SE_i, \ldots, SE_N; N = 2, 3, 4\}$ and linked by the relation R. We denote $p(SE|R)$ the probability that the sub-expressions SE_i forming SE are related by the relation R, where: $p(SE|R) = \prod_{i=1}^{N} p(SE_i|R_i^s); R_i^s$ is the sub-relation that connects the node of relation R and the ith child. For instance, the relation R = *superscript* has $N = 2$ sub-relations, $R_1^s = base$ and $R_2^s = exponent$. By applying the Bayes rule, the probability that a relation produces a sub-expression knowing its model is given by:

$$p(R|SE) = \frac{p(SE|R).p(R)}{p(SE)} \tag{8}$$

The term $p(SE)$ can be ignored because it is constant for all the relations, thus the probability of a relation R is:

$$p(R|SE) \propto \prod_{i=1}^{N} p(SE_i|R_i^s).p(R) \tag{9}$$

where $p(R)$ is the a priori probability of the relation R computed from the distribution of relations in the training database. The probability $p(SE_i|R_i^s)$ that a sub-expressions SE derives from the sub-expression SE_i is given by the Gaussian models g using the corresponding spatial information (dh_i, dy_i):

$$p(SE_i|R_i^s) = g_{dh}^{R_i^s}(dh_i).g_{dy}^{R_i^s}(dy_i) \tag{10}$$

where $g_{dx}^{R_i^s}(x) = e^{-\frac{(x-\mu_{R,i,dx})^2}{2\sigma_{R,i,dx}^2}}$ with $\mu_{R,i,dx}$ and $\sigma_{R,i,dx}^2$ are the Gaussian parameters for the ith child of the spatial relation R considering the feature x.

We have defined 11 relations based on two or three elements (Awal, 2010). For example, the "Left–Right" that is constructed from two elements is the most basic one. Other two element relations are the "Sub-script" and the "Super-script" that represent the index and exponent relations. Binary mathematical operations $(+, -, \rightarrow, *, \ldots)$ are represented by a three element relation called "Operator". Each of the other mathematical operations (such as $\sum, \int, (), \ldots$) is associated to a specific spatial relation. For example, the sum relation is modeled by two rules. A vertical one composed of three elements to produce the sum with its limits (such as $\sum_{i=0}^{N}$), the result of this rule is then associated within a horizontal rule to form the complete expression (such as $\sum_{i=0}^{N} x_i$).

During the relation learning phase, relations have been directly extracted from correctly recognized expressions by forcing a correct symbol classification from the ground truth. We adopt this strategy to avoid a handmade labeling of expressions at the relation level. Then the *dh* and *dy* of each relation are used to estimate the Gaussian parameters.

6. Experiments

6.1. Expression dataset

The proposed architecture requires a large expression database in order to train the symbol classifier and to model the spatial relations directly from expressions. Obtaining such a database from real handwritten expressions is possible, but not easy to achieve. We have proposed a tool (Awal et al., 2009) that allows producing any dataset of handwritten MEs from a given corpus and previously collected isolated symbols. It generates pseudo-synthetic handwritten MEs using a stochastic layout guided by the LaTeX string defining the expression.

Currently, the corpus is extracted from the "Aster" database proposed in (Raman, 1994). A set of 36 expressions is chosen covering a majority of mathematical domains (Awal et al., 2009). We will call it: "RamanReduced" corpus.

Each expression contains in average 11 symbols, representing 34 distinct classes. First, three databases (1st three lines in Table 1) are generated: each expression is artificially synthesized using isolated symbols from the CIEL database collected from 280 writers (Awal et al., 2009) and the IRONOFF database (Viard-Gaudin et al., 1999).

Expressions in the RamanReduced_CIEL DBs are generated using isolated symbols of only one writer at the time. A drawback of this technique is that the very same sample of a given symbol will be repeated in all the expressions of a given writer. For this reason, we have introduced the notion of virtual writers to produce the RamanReduced_IROCIEL DB. In this case, the number of isolated symbols is much bigger than the required samples. A variety of writers' samples is used to produce one expression in order to assure large symbol variability.

In addition we have collected real expressions where each expression is written entirely by one real writer. In the short term, real databases are only used for testing the system in order to evaluate its performances. Isolated symbols and also real expressions have been collected using a pen and paper technology.

We collected the RamanReduced_Real dataset within our research group. Ten writers wrote the expressions, and as a result 70 expressions have been collected. Secondly, we have collected a larger expression database in order to obtain a big database that can eventually be used as a real train database. A total number of 77000 MEs have been extracted from the French Wikipedia within 7,000 web pages. Those expressions have been filtered using two criteria: the symbol set and the length (between 3 and 49 symbols).

A subset of 6144 expressions has been chosen randomly to be collected using forms printed on *Anoto* papers. University students, professors, and researchers have participated in this collection (512 persons). Each writer filled out two forms for a total of 12 expressions. However, only 211 expressions are compatible with the RamanReduced corpus, thus we refer to this sub-base as RamanReduced_Wiki_CIEL (test).

Since there are few available handwritten expression databases, we have decided to share our real expression databases in order to enrich the comparison between different proposed systems and methods, which is actually not possible because each research group has their own expression databases. RamanReduced_Real and RamaReduced_Wiki_CIEL are publicly available[1]. A sub-part of our real expressions has been used to construct the CROHME database used in the online handwritten MEs recognition competition (Mouchère et al., 2011, 2012).

6.2. Evaluation

It is inappropriate to evaluate an expression recognition system only at the expression level, especially when dealing with long and complex expressions. In consequence, we have chosen three measures similar to those used recently in (Mouchère et al., 2011, 2012; Rhee and Kim, 2009; Zanibbi, 2011), we used the following measures:

$$SegRate = (correctly\ segmented\ symbols)/number\ of\ symbols$$
$$RecoRate = (correctly\ recognized\ symbols)/number\ of\ symbols$$
$$ExpRate = (correctly\ recognized\ expressions)/number\ of\ expressions$$

These same measures are also used in order to validate the recognizer during the training stage.

6.3. Results

The objective of these experiments is to measure the performance of the MEXREC system under different conditions. We focus

[1] AWAL_EM database : http://www.irccyn.ec-nantes.fr/spip.php?article638&lang = en.

Table 1
Constitution of the MEs databases.

Database	#Isolated DB	#Writers	Expression database		Symbol database
			#Expressions	#Symbols	#Symbols
RamanReduced_CIELTraining	CIEL	180	180 × 36 = 6480	180 × 412 = 74,160	180 × 34 = 6120
RamanReduced_IROCIELTraining	CIEL IRONOFF	200	200 × 36 = 7200	200 × 412 = 82,400	180 34 + 480 x 15 = 13,320
RamanReduced_CIELTest	CIEL	100	100 × 36 = 3600	100 × 412 = 41,200	100 × 34 = 3400
RamanReduced_RealTest	n.a.	10	70	784	n.a.
RamanReduced_Wiki_CIEL Test	n.a	20	211	1477	n.a

in this section on the results obtained to evaluate (i) the importance of the junk class, (ii) the impact of the training databases.

Note that the impact of the spatial relation modeling has been studied in (Awal et al., 2010b). Finally, we will compare the system with other systems that have participated in the CROHME 2011 competition.

6.3.1. Importance of the junk class

The symbol classifier is a very important stage of the global system. In addition to recognizing the symbols, it guides the segmentation process and participates in calculating the global cost function. As we have explained before, a classifier trained on isolated symbols has difficulties dealing with invalid segmentations (junk). Table 2 shows the performance of the expression recognizer using a global classifier (TDNN); compared to reference results obtained using an isolated classifier without rejection capacity (TDNN). For this experiment, RamanReduced_CIEL training set has been used.

We can observe that the system performance is significantly improved at the expression recognition level when using a classifier with rejection capacity (using an additional junk class trained in a global scheme). In fact, the *ExpRate* is improved from 25.6% to 61.8% on the RamanReduced_CIEL test DB, and from 11.4% to 27.1% on the RamanReduced_Real DB.

6.3.2. Impact of the training databases

The choice of the training database is essential to efficiently train the symbol classifier. It must be representative of the corpus domain, and also covers the variability of writing styles. When the global learning is based on synthetic expressions, the variability of the expressions can be augmented by enriching the isolated DBs used in the generation process.

Two different training databases are used to obtain the results presented in Table 3. The first, RamanReduced_CIEL is produced classically with the drawback of repeated samples. Where on the other hand the second DB, RamanReduced_IROCIEL, is produced using virtual writer technique. In both cases, the same Gaussian structural models, estimated with the RamanReduced_CIEL DB, are used.

We can conclude from the expression recognition rates shown in Table 3 that the introduction of virtual writers in the training database improves significantly the performance of the system. By doing this, it has been possible to use a bigger set of symbols samples (13,320 instead of 6120). The global performance of the system is improved by 6.1% on the synthetic test DB, and 8.2% on the real test DB (the 281 MEs).

6.3.3. CROHME competition

In fact, direct comparison with existing systems is inappropriate because systems are tested with different expression databases and different evaluation measures (Lapointe and Blostein, 2009) (Awal et al., 2010c). For this reason, an international competition has been held in the International Conference on Document Analysis and Recognition (ICDAR 2011) (Mouchère et al., 2011).

The training and test datasets were subdivided into two parts. Part-I contains 296 training expressions and other 181 for the test. The part-II consists of 921 training expressions and other 348 for the test. The part-II expression set includes the part-I expressions. The grammar controlling the part-II expressions is more complex than that of part-I, and thus the expressions are more difficult to recognize. Moreover, the number of distinct symbols is bigger in the part-II set.

Table 4 shows the results of our system, which was considered out of competition, compared to the winner one. At the time of the 2011 competition, our best system was trained on RamanReduced_IROCIEL using a global classifier and a Gaussian structural model. After this competition, we have updated the system by:

- increasing the usage of real expressions during the training phase with data from the HAMEX database (Quiniou et al., 2011)
- solving some problems in grammar definition and mathML generation
- improving the scaling normalization of symbol hypothesis before recognition

We can observe that at the expression level, our system outperforms the winning system. However, its performance is slightly lower than that of the winner system at the levels of symbol segmentation (87.56–88.07 in Part-I and 84.23–87.82 in Part-II) and symbol recognition (91.67–92.22 in Part-I and 87.16–92.56 in Part-II). However, the system was able to recover at the expression level (global level) and achieves a very good performance compared to the winner system (40.88–29.28 in Part-I and 22.41–19.83 in Part-II). This should mean that our system is more efficient in the interpretation stage. However, a deeper analysis of miss-recognized expressions in term of structure errors should be done. The slightly lower segmentation and recognition rates in the 2011 competition have been overcome in the updated version of our sys-

Table 2
System's performance with or without rejection capacity.

Test dataset	Symbol classifier	SegRate%	RecoRate%	ExpRate%
RamanReduced_CIEL	Isolated (wihout reject)	64.2	62.9	25.6
	Global (with reject)	86.9	84.6	61.8
RamanReduced_Real	Isolated (wihout reject)	50	46.6	11.4
	Global (with reject)	78.7	72.5	27.1

Table 3
System's performance using different training databases.

Test dataset	Training database	SegRate	RecoRate	ExpRate
Synthetic MEs (RamanReduced_CIEL)(3600 MEs)	RamanReduced_CIEL(6480 MEs)	91.4	88.7	64.9
	RamanReduced_IROCIEL(7200 MEs)	94.3	92.1	71.0
Real MEs(281 MEs)	RamanReduced_CIEL(6480 MEs)	83.0	77.4	40.9
	RamanReduced_IROCIEL(7200 MEs)	88.3	84.8	49.1

Table 4
Results on CROHME 2011 Test Set.

Dataset	Systems	SegRate	RecoRate	ExpRate
Part-I	Winner	88.07	92.22	29.28
	Our system in 2011	87.56	91.67	40.88
	Our updated system	89.79	95.21	63.54
Part-II	Winner	87.82	92.56	19.83
	Our system in 2011	84.23	87.16	22.41
	Our updated system	87.04	92.47	47.41

tem. We have succeeded not only to slightly outperform the winning system at the symbol segmentation and recognition levels, but also to increase the expRate by almost 50% for PART-I and 100% for PART-II (40.88 → 63.54 and 22.41 → 47.41 respectively).

7. Conclusion and perspective

A complete system of MEs recognition systems has been presented. The classic three steps of 2D language recognitions are applied simultaneously in order to reduce error propagation from one step to another. We approach the recognition problem as a search for the best possible interpretation of a sequence of input strokes. Unlike most existing works, we have considered a symbol classifier with a reject capacity in order to deal with the invalid hypotheses proposed by the hypothesis generator. This global approach is applicable for any 2D languages, like in our recent work on flowchart recognition (Awal et al., 2011).

The originality of our system stems from the global learning schema. This learning allows training the symbol classifier directly from mathematical expressions. The advantage of this global learning is to consider the junk examples and include them in the classifier knowledge. Furthermore, we have proposed a contextual modeling based on structural analysis of the expression. This analysis is based on models learnt directly from the expressions using the global learning scheme. Although in our current implementation the grammar has only a filtering role on the candidate expression outputs, it might also be possible to take advantage of the global learning strategy to update the spatial relationship models as it is done in the backpropagation step with the neural network classifier. With such a system, we have obtained the best results in the CROHME 2011 competition; we have no doubt that this will stimulate this research area and that for CROHME 2012, new comers will present very competitive systems.

References

Aly, W., Uchida, S., Fujiyoshi, A., Suzuki, M., 2009. Statistical classification of spatial relationships among mathematical symbols. In: 10th Internat. Conf. on Document Analysis and Recognition, Barcelone, pp. 1350–1355.

Anderson, R.H., 1968. Syntax-directed recognition of handprinted two-dimensional mathematics in interactive systems for experimental. Appl. Math., 436–459.

Awal, A.-M., 2010. Reconnaissance de structures bidimensionnelles: Application aux expressions mathématiques manuscrites en-ligne, Ph.D. Thesis, University of Nantes.

Awal, A.-M., Mouchère, H., Viard-Gaudin, C., 2009. Towards handwritten mathematical expression recognition. In: 10th Internat. Conf. on Document Analysis and Recognition, Barcelone, pp. 1046–1050.

Awal, A.-M., Mouchère, H., Viard-Gaudin, C., 2010a. A hybrid classifier for handwritten mathematical expression recognition. In: Document Recognition and Retrieval XVII, San Fransisco, pp. 1–10.

Awal, A.-M., Mouchère, H., Viard-Gaudin, C., 2010b. Improving online handwritten mathematical expressions recognition with contextual modelling. In: Internat. Conf. on Frontiers in Handwriting Recognition, Kolkata, pp. 427–432.

Awal, A.-M., Mouchère, H., Viard-Gaudin, C., 2010c. The problem of handwritten mathematical expression recognition evaluation. In: Internat. Conf. on Frontiers in Handwriting Recognition, Kolkata, pp. 646–651.

Awal, A.-M., Feng, G., Mouchère, H., Viard-Gaudin, C., 2011. First experiments on a new online handwritten flowchart database. In: Document Recognition and Retrieval XVIII, San Fransisco.

Belaid, A., Haton, J.P., 1984. A syntactic approach for handwritten mathematical formulae recognition. Pattern Anal. Machine Intelligence 6, 105–111.

Chan, K.-F., Yeung, D.-Y., 2000a. An efficient syntactic approach to structural analysis of on-line handwritten mathematical expressions. Pattern Recognition 33, 375–384.

Chan, K.-F., Yeung, D.-Y., 2000b. Mathematical expression recognition: A survey. Internat. J. Doc. Anal. Recognition 3, 3–15.

Chan, K.-F., Yeung, D.-Y., 2001. PenCalc: A novel application of on-line mathematical expression recognition technology. In: Sixth Internat. Conf. on Document Analysis and Recognition, Seattle, pp. 775–778.

Chang, S.K., 1970. A method for the structural analysis of 2-D mathematical expressions. Infor. Sci. 2 (3), 253–272.

Coüasnon, B., 2001. DMOS: A generic document recognition method, application to an automatic generator of musical scores, mathematical formulae and table structures recognition systems. In: Sixth Internat. Conf. on Document Analysis and Recognition, Seattle, pp. 215–220.

Delaye, A., Anquetil, E., Macé, S., 2009. Explicit fuzzy modeling of shapes and positioning for handwritten Chinese character recognition. In: 10th Internat. Conf. on Document Analysis and Recognition, Barcelone, pp. 1121–1125.

Dimitriadis, Y.A., Coronado, J.L., 1995. Towards an ART based mathematical editor that uses online handwritten symbol recognition. Pattern Recognition 28, 807–822.

Dimitriadis, Y.A., Coronado, J.L., la, C.d., 1991. A new interactive mathematical editor, using on-line handwritten symbol recognition, and error detection-correction with an attribute grammar. In: First Internat. Conf. on Document Analysis and Recognition, St. Malo, pp. 885–893

Eto, Y., Suzuki, M., 2001. Mathematical formula recognition using virtual link network. In: Sixth Internat. Conf. on Document Analysis and Recognition, Seattle, pp. 762–767.

Feng, G., Viard-Gaudin, C., Sun, Z., 2009. On-line hand-drawn electric circuit diagram recognition using 2D dynamic programming. Pattern Recognition 42, 3215–3223.

Fitzgerald, J.A., Geiselbrechtinger, F., Kechadi, T., 2006. Structural analysis of handwritten mathematical expressions through fuzzy parsing. In: The Internat. Conf. on Advances in Computer Science and Technology, Puerto Vallarta, pp. 151–156.

Fitzgerald, J.A, Geiselbrechtinger, F. & Kechadi, T, 2007. Mathpad: A fuzzy logic-based recognition system for handwritten mathematics. In: Ninth Internat. Conf. on Document Analysis and Recognition, Curitiba, pp. 694–698.

Fukuda, R., et al., 1999. A technique of mathematical expression structure analysis for the handwriting input system. In: Fifth Internat. Conf. on Document Analysis and Recognition, Bangalore, pp. 131–134.

Garain, U., Chaudhuri, B., 2004. Recognition of online handwritten mathematical expressions. Trans. Systems, Man Cybernet. 34, 2366–2376.

Geneo, R., Fitzgerald, J.A., Kechadi, T., 2006. A purely online approach to mathematical expression recognition. In: Internat. Workshop on Frontiers in Handwriting Recognition, La Baule, pp. 255–260.

Grbavec, A., Blostein, D., 1995. Mathematics recognition using graph rewriting. In: Third Internat. Conf. on Document Analysis and Recognition, MontReal, pp. 417–421.
Ha, J., Haralick, R.M., Philips, I.T., 1995. Understanding mathematical expressions from document images. In: Third Internat. Conf. on Document Analysis and Recognition, MontReal, pp. 956–959.
Keshari, B., Watt, S.M., 2007. Hybrid mathematical symbol recognition using support vector machines. In: Ninth Internat. Conf. on Document Analysis and Recognition, Curitiba, pp. 859–863.
Kosmala, A., Rigoll, G., Lavirotte, S., Pottier, L., 1999. On-line handwritten formula recognition using hidden Markov models and context dependent graph grammars. In: Fifth Internat. Conf. on Document Analysis and Recognition, Bangalore, pp. 107–110.
Lapointe, A., Blostein, D., 2009. Issues in performance evaluation: A case study of math recognition. In: 10th Internat. Conf. on Document Analysis and Recognition, Barcelone, pp. 1355–1360.
Lavirotte, S., Pottier, L., 1998. Mathematical formula recognition using graph grammar. In: Proceedings of the SPIE, pp. 44–52.
Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324.
Lehmberg, S., Winkler, H.-J., Lang, M., 1996. A soft-decision approach for symbol segmentation within handwritten mathematical expressions. In: Internat. Conf. on Acoustics, Speech, and Signal Processing, Atlanta, pp. 3434–3437.
Macé, S., Anquetil, E., 2009. Eager Interpretation of on-line hand-drawn structured documents: The DALI methodology. Pattern Recognition 42, 3202–3214.
Miller, E.G., Viola, P.A., 1998. Ambiguity and constraint in mathematical expression recognition. In: The 15th National Conf. on Artificial Intelligence, Madison, pp. 784–791.
Mitra, J., et al., 2003. Automatic understanding of structures in printed mathematical expressions. In Seventh Internat. Conf. on Document Analysis and Recognition, Edinburgh, pp. 540–544.
Mouchère, H., et al., 2011. CROHME2011: Competition on recognition of online handwritten mathematical expressions. In: 11th Internat. Conf. on Document Analysis and Recognition, Beijing, pp. 1497–1500.
Mouchère, H., et al., 2012. CROHME 2012: Competition on recognition of online handwritten mathematical expressions. In: To appear in 13th Internat. Conf. on Frontiers in Handwriting Recognition, Bari.
Plamondon, R., Srihari, S.N., 2000. On-line and off-line handwriting recognition: A comprehensive survey. IEEE Trans. Pattern Anal. Machine Intelligence 22, 63–84.
Prusa, D., Hlavac, V., 2007. Mathematical formulae recognition using 2D grammars. In: Ninth Internat. Conf. on Document Analysis and Recognition, Curitiba, pp. 849–853.
Quiniou, et al., 2011. HAMEX – A handwritten and audio dataset of mathematical expressions. In: 11th Internat. Conf. on Document Analysis and Recognition. Beijing, pp. 452–456.
Raman, T.V., 1994. Audio system for technical readings. Ph.D. Thesis.
Rhee, T.-H., Kim, J.-H., 2009. Efficient search strategy in structural analysis for handwritten mathematical expression recognition. Pattern Recognition 42, 3192–3201.
Scott, M., George, L., 2010. Recognizing handwritten mathematics via fuzzy parsing. Technical report. University of Waterloo.
Shi, Y., Li, H.Y., Soong, F.K., 2007. A unified framework for symbol segmentation and recognition of handwritten mathematical expressions. In: Ninth Internat.l Conf. on Document Analysis and Recognition, Curitiba, pp. 85–858.
Szwoch, M., 2007. Guido: A musical score recognition system. In: Ninth Internat. Conf. on Document Analysis and Recognition, Curitiba, pp. 809–813.
Tapia, E., Rojas, R., 2003. Recognition of on-line handwritten mathematical formulas in the E-Chalk system. In: Seventh Internat. Conf. on Document Analysis and Recognition, Edinburgh, pp. 980–984.
Tapia, E., Rojas, R., 2005. Recognition of on-line handwritten mathematical expressions in the E-Chalk system – An extension. In: Eighth Internat. Conf. on Document Analysis and Recognition, Seoul, pp. 1206–1210.
Tokuyasuy, T.A., Chou, P.A., 1999. An iterative decoding approach to document image analysis. In: IAPR Workshop on Document Layout Interpretation and its Applications.
Viard-Gaudin, C., Lallican, P.-M., Knerr, S., Binter, P., 1999. The IRESTE On/Off (IRONOFF) dual handwriting database. In: Fifth Internat. Conf. on Document Analysis and Recognition, Bangalore, pp. 455–458.
Wang, X., Shi, G., Yang, J., 2009. The understanding and structure analyzing for online handwritten chemical formulas. In: Tenth Internat. Conf. on Document Analysis and Recognition, Barcelone, pp. 1056–1061.
Wilpon, J.G., Rabiner, L.R., Lee, C.-H., Goldman, E.R., 1990. Automatic recognition of keywords in unconstrained speech using hidden Markov models. IEEE Trans. Acoust. Speech Signal Process. 38, 1870–1878.
Yamamoto, R., Sako, S., Nishimoto, T., Sagayama, S., 2006. On-line recognition of handwritten mathematical expressions based on stroke-based stochastic context-free grammar. In: 10th Internat. Workshop on Frontiers in Handwriting Recognition, La Baule, pp. 249–254.
Yuan, Z., Pan, H., Zhang, L., 2008. A novel pen-based flowchart recognition system for programming teaching. Lect. Notes Comput. Sci. 5328, 55–64.
Zanibbi, R., et al., 2011. Stroke-based performance metrics for handwritten mathematical expressions. In: 11th Internat. Conf. on Document Analysis and Recognition, Beijing, pp. 334–338.
Zanibbi, R., Blostein, D., 2002. Recognizing mathematical expressions using tree transformation. Trans. Pattern Anal. Machine Intelligence 24, 1455–1467.
Zhang, L., Blostein, D., Zanibbi, R., 2005. Using fuzzy logic to analyze superscript and subscript relations in handwritten mathematical expressions. In: Eighth Internat. Conf. on Document Analysis and Recognition, Seoul, pp. 972–976.
Zhu, H., Tang, L., Liu, P., 2006. An mlp-orthogonal quassian mixture hybrid model for chinese bank check printed numeral recognition. Internat. J. Doc. Anal. Recognition 8, 27–34.

2014 14th International Conference on Frontiers in Handwriting Recognition

Text Alignment from Bimodal Mathematical Expression Sources

Sofiane MEDJKOUNE, Harold MOUCHERE
and Christian VIARD-GAUDIN
LUNAM University, University of Nantes
IRCCyN UMR CNRS 6597
Rue Christian Pauc BP 50609 44306, Nantes, France
firstname.lastname@univ-nantes.fr

Simon PETITRENAUD
LUNAM University, University of Le Mans
LIUM - EA 4023
Avenue Laënnec, 72085 LE MANS CEDEX 9,
Le Mans, France
simon.petit-renaud@lium.univ-lemans.fr

***Abstract*—In this paper we propose a new approach to merge mathematical expression recognition results coming from handwriting and speech modalities. Using a bimodal description of mathematical expressions allows taking advantage of the complementarities between both signals, and can disambiguate situations were a single modality would not be clear enough. To combine the signals coming from both modalities, we propose to represent them in the same space as a textual description. First, from the handwriting signal, we generate the Nbest mathematical expressions; each of them is next translated as different possible strings. From the audio signal, an automatic speech recognition system provides a transcript, which is also available as a string. A string comparison algorithm is achieved to select the best mathematical expressions. This bimodal system is evaluated on real bimodal data from the HAMEX dataset and the results are compared to a single modality (handwriting) based system.**

I. Introduction

Mathematical expression (ME) recognition problem is attracting more and more interest within the scientific community. This is mainly due to the usefulness of the mathematical language and the challenges that this kind of problems raises. A particular representation in two dimensions with many special symbols has been developed for centuries, to facilitate the way that humans communicate mathematics with each others. Even if this graphical representation greatly assists in the transmission of the information conveyed by the studied mathematical principle, the insertion of such bi-dimensional elements in electronic documents is difficult. In fact, the bi-dimensional nature of ME, combined with the huge number of elementary symbols which are involved in its writing, increase the difficulty of interacting with a computer using mathematics based on traditional interfaces (mouse/keyboard). To recognize mathematical expressions, the technological progress offers alternative interaction modes that are more natural for human beings. In particular, speech and handwriting are among the most common ones. These modalities are very complementary, especially for mathematical equation description. A very common case is the following: a lecturer is writing a ME on a classroom blackboard and is dictating it in the same time to prevent from any misinterpretation from the audience (note that the two signals are not necessary synchronized). An example of such ambiguities is given on figure 1. To perform an automatic interpretation of either one or the other of both signals, some difficulties are encountered. These latter are intrinsic to each modality and the complementarity we discussed above can be

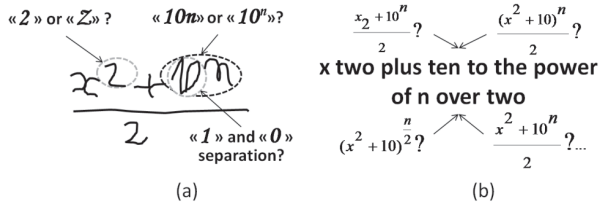


Fig. 1. Examples of intrinsinc ambiguities embeded by the (a) handwriting modality, (b) speech modality

used to increase the reliability of the interface in charge of mathematical expressions entering to a computer. Thus, in this work, we propose a bimodal architecture using the spoken and handwritten forms of the ME to recognize. More precisely, we exploit the Nbest list of mathematical expressions proposed by the system in charge of the handwriting modality, and by using a dedicated LaTeX string to text converter, we derive many different possible text translations. These translations are compared to the automatic transcription obtained from the system in charge of the speech signal interpretation. The best text alignment indicates the ME to keep as the best interpretation. The paper is organized as follows: in the second section, the handwriting based mathematical expression recognition (MER) is presented. Section III gives a short review on spoken MER. The fusion based MER approach we propose here is presented in section IV. We report the corresponding results in section V, and we conclude the paper in section VI.

II. Handwritten Mathematical Expression Recognition

We are considering online handwritten ME. This means that the raw data arriving to the handwriting recognition system is a sequence of elementary strokes which are ordered in time. In this work, we will consider that every symbol can be written with one or several strokes which are not necessarily consecutive, since some of them can be delayed. Most often, before starting the recognition process itself, the input signal undergoes a preprocessing step (spatial resampling, rescaling ...) [1], [2]. This preprocessing ensures consistency during the following processing steps, especially for the recognition one.

Generally, three sequential but interdependent steps have

2167-6445/14 $31.00 © 2014 IEEE
DOI 10.1109/ICFHR.2014.42

been identified to achieve handwritten ME recognition [1], [3]. The first step is the *segmentation* process in which the possible groups of strokes are formed. This stage is not trivial when it is supposed that interspersed symbols are authorized. Each *group* is called a segmentation hypothesis (*'sh'*). Ideally, each *'sh'* corresponds to a mathematical symbol. The recognition process is the second step. It aims to assign a symbol label (or a list of possible symbols) and a recognition score for each *'sh'*. The third step is the structural analysis. All the recognized symbols are used to make the final interpretation of the ME. This is done through a spatio-grammatical analysis. A drawback of such an approach, optimizing separately each step, is that the failure of one step can lead to the failure of the next one (error propagation). Rhee and Kim reported in [4] a solution to reduce this error propagation with the simultaneous optimization of the segmentation and recognition steps. However, in this case, the classifier is trained separately on isolated symbols. Later an improvement has been proposed by Awal and al. with a more global architecture [5]. The strengths of their system are the following. First of all, the recognition module is trained within the expressions directly from the outputs of the segmentation module. This allows a direct interaction between the different stages of the system (segmentation, recognition and 2D parsing). Secondly, during the segmentation step, a non-consecutive stroke grouping is allowed to form valid symbols. In addition, the classifier in charge of labeling each *'sh'* has the power to reject invalid hypotheses thanks to a *junk* class which is dedicated to label wrong segmentation hypotheses. Finally, the structural analysis (2D parsing) is controlled by both symbol recognition scores and a contextual analysis (spatial costs). The handwritten MER sub-part used in our architecture will be largely based on Awal and al.'s system.

III. Spoken mathematical expression recognition

Mathematical expression recognition based on automatic speech recognition (ASR) involves two main modules [6], [7]. The first one achieves the automatic speech recognition task. The output of this module provides a textual description which depends of the audio description and of the ASR reliability. This text is composed of words written with alphabetic characters as they are recognized by the ASR system. This text is ideally a fair description of the ME (it depends on the quality of word pronunciation by the speaker). Fig. I (b) gives an example of a possible recognized string by the ASR system "x two plus ten to the power of n over two". The second module is a parser, which processes the previous transcription in the 2D space to deduce the associated ME.

The automatic transcription is given by an ASR system which is quite similar to the one described in the case of handwriting modality. The main difference is the nature of the signal which is processed (acoustic in this case). This recognition procedure involves three stages. During the first one, the acoustic signal is filtered and re-sampled, then a frame description is produced, where a feature vector is computed for each window of 25 ms with an overlap of 10 ms. The features are the cepstral coefficients and their first and second derivatives [8]. Segmentation into homogeneous parts is operated in a second step. Resulting segments are close to minimal linguistic units. The last step is the decoding itself using models and tools learned within a training step (acoustical model, pronunciation dictionary and language model).

Parsing the resulting transcription from the previous module is a very hard task. In the rare existing systems [6], [7], the parsing is most of the time assisted by either introducing some dictation rules (in order to separate the numerator and the denominator of a fraction, for instance) or using an additional source of information (such as using a mouse to point the position where to place the different elements). By adding such constraints, the editing process becomes less natural and far from what is expected from this kind of systems.

The work we report in this paper concerns the French spoken language. The task of speech recognition in our system is carried out by a system largely based on the one developed at the LIUM [8], which kernel is one of the most popular worldwide speech recognition systems (CMU-Sphinx)[9].

IV. Bimodal mathematical expression recognition

A. The data fusion principle

The idea of multi-modal human-machine interaction comes from the observation of the human beings' interaction. Usually, people simultaneously use many communication modes to converse. In so doing, the conversation becomes less ambiguous. The main goal of this work is to mimic this procedure to be able to set up a multi-modal system dedicated to mathematical expressions recognition (MER).

Generally, data fusion methods are divided in three main categories [10], [11]: *early fusion* which happens at features levels; *late fusion* which concerns the intermediate decisions fusion and the last one is the *hybrid fusion* which is a mix of the two. Within each approach, three kinds of methods can be used to carry out the fusion process. Rules based approaches represent the first category and include methods using simple operators such as max, (weighted) mean or product. The second category is based on classifiers and the last one is based on parameter estimation.

Since we are interested in combining two heterogeneous signals (handwriting and audio streams), we decided to consider a late fusion strategy, to ensure to use suitable recognition systems with respect to each modality. In a previous work [12], we have merely proposed a bag of words approach to combine information coming from the audio description in the main stream processing of the handwritten signal. The problem of alignment of the two streams was not investigated during these previous works, which can highly penalize the combination process. Thus, the matter of this paper is to consider the audio and handwriting streams alignment in order to improve the global performance of the system. In the following sections we describe the architecture of the proposed collaborative system.

B. Data fusion for mathematical expression recognition

The proposed architecture for bimodal mathematical expressions recognition (BMER) is presented in Fig. 2. Its overall description is as follows.

The system input is a ME available at both spoken and handwritten forms. An automatic speech recognition (ASR)

system is in charge of the interpretation of the speech signal describing the ME (*cf.* section III). The output of this system is a text describing the ME. At this level, the result is still one dimensional, it is a standard text. This textual description is composed of two categories of words, the first one concerns words which are useful in a mathematical language point of view, we call them *keywords*. The other category includes all the other words which are present in the text, as *stop words*, only for linguistic fluentness. Only the keywords are of interest for us, thus we automatically filter the textual description to keep only this category of words. Similarly, the handwriting recognition module processes the on-line handwritten signal to form the basic symbol hypotheses from the raw signal (sequence of strokes), as explained in section II. After this step, a list of labels and their corresponding scores are assigned to each symbol. The set of resulting symbols is then parsed in the $2D$ plan to define the ME layout. In the previous work [12], we considered that the fusion process can be carried out at two levels: directly during the symbol recognition step and next during the structural analysis to identify the spatial relationships. In this novel approach we propose here, we delay the fusion operation until the interpretation step of the full ME by the handwriting based system. The main idea is to run the full process of ME recognition considering only the handwriting modality. This system is able to provide an $Nbest$ list of possible LaTeX strings corresponding to the input signal. Once this $Nbest$ list obtained, each LaTeX strings is sent to the *LaTeX2Text* module which elaborates various possible translations with regard to each LaTeX string. Accordingly, the audio stream, through its associated automatic transcription, is exploited to make a re-ranking of the list proposed by the handwriting system. This merging process is done with the help of the fusion units (grey boxes) in the architecture of figure 2 described below.

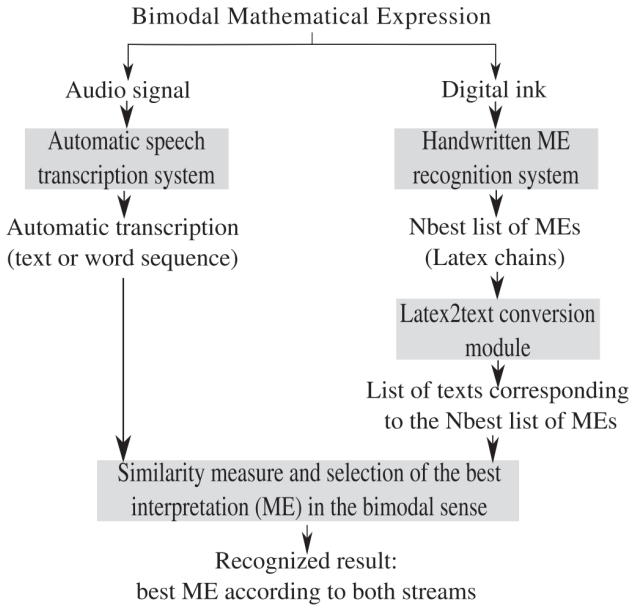


Fig. 2. The collaborative architecture for complete mathematical expression recognition

1) LaTeX2Text conversion module: The role of this module is to give a textual description associated with the LaTeX chain of a ME. This textual description is intended to be the most natural. It should be as close as possible to the description of a dictation provided by a speaker. Since various dictations are possible for a given ME LaTeX chain, the generator gives many different ways to dictate the same expression. In table I is given an example of such a procedure.

TABLE I. EXAMPLE OF DIFFERENT POSSIBLE TRANSLATIONS FOR A SAME LaTeX STRING

LaTeX chain	Associated translations in French (in English)
`$\frac{x^2+10^n}{2}$`	- *"x carré plus dix n sur deux"* *(x squared plus ten n over two)* - *"x au carré plus dix puissance n le tout sur deux"* *(x squared plus ten to the power n all over two)* - *"x puissance deux plus dix à la puissance n sur deux"* *(x to the power two plus ten to the power n over two)* - *"x carré plus dix puissance n divisé par deux"* *(x squared plus ten to the power n divided by two)* ⋮

2) Similarity measure and best solution selection module: The $Nbest$ list of MEs given by the handwriting recognition system and initially represented in a LaTeX form, are now represented in the textual description space. Each LaTeX chain produces one or more string(s). Each of these textual descriptions is compared to the transcription issued from the ASR system and a score is associated to each measure. The highest score gives the best matching and consequently the best ME to consider as the final solution. The similarity measure is based on the *Levenshtein distance* between strings. This metric is composed of three quantities: the number of substitutions denoted *Subst*, the number of the deletions (*Del*) and the number of the insertions (*Ins*). If this measure is normalized considering the number of words in the reference (the automatic transcription from the speech system in our case), denoted by *N*, we obtain the **W**ord **E**rror **R**ate (WER) metric. Equation 1 defines the WER which can exceed 100% since sometimes it is required to perform several operations to recover one word of the reference string:

$$WER = \frac{Ins + Del + Subst}{N} \quad (1)$$

It is also possible to compute the **W**ord **C**orrect **R**ate (WCR) as defined by equation 2, which is bounded between 0 and 100%:

$$WCR = 1 - \frac{Subst + Del}{N} \quad (2)$$

We select the solution in the $Nbest$ list which gives the best WCR with respect to the transcription of the audio signal. Indeed, since our goal is to keep the text describing a ME which has the higher common number of words compared to the audio description, we are not interested by the number of inserted words (Ins).

This similarity measure is calculated on preprocessed texts to remove all stop words and consider only keywords, as explained before.

V. EXPERIMENTAL RESULTS

In this section we present the performances of the system reported in this paper. First, we give an overview of the dataset we used. Then the performances of the mono-modal systems are presented. After that we report the results concerning our system compared to the baseline system based only on the handwriting signal and also compared to the previous architecture, we presented in [12] (based on fusion at lower levels: symbols and relations).

A. Dataset description

The data used to perform the experiment is from the *HAMEX* [13] database. This database includes a set of approximately 4 350 ME, each of them available in the spoken and the handwritten modalities. The vocabulary covered by *HAMEX* contains 74 mathematical symbols, including all the Latin alphabet letters, the ten digits, six letters from the Greek alphabet and various mathematical symbols (integral, summation...).

B. Specialized systems performance

The handwriting recognition task is accomplished with the on-line handwritten MER system that participated to $CROHME2012$[1] competition [14]. The results reported here concern a set of 519 MEs of the *HAMEX* test part which satisfies the *CROHME* (task 2) grammar and vocabulary (56 symbol classes). A set of 500 MEs of the *HAMEX* train part satisfying the same conditions as before are used to tune the different parameters we consider in the fusion system. Finally, the models of the ASR system are trained on the whole speech data of the HAMEX train part. Concerning the fusion process itself, the value of $Nbest$ ME is set experimentally to 10 using the validation database. We report on Table II the performances of the handwriting system considering that the valid solution is ranked first ($TOP1$), or ranked among the first two answers ($TOP2$) and so on.

TABLE II. PERFORMANCES OF THE HANDWRITING RECOGNITION SYSTEM

Evaluation level	*TOP1*	*TOP2*	*TOP3*	*TOP4*	*TOP5*	*TOP10*	*more*
Reco. rate [%]	34.10	42.08	44.6	45.6	45.75	47.68	48.36

In another side, the recognition rate of the automatic speech transcription system with respect to the whole vocabulary guiding the ASR system is 90.06%. If only keywords are considered for the evaluation, the recognition rate is increased to 97.21%. This rate is given at the word level, not as in Table II at the expression level, since at that stage the interpretation of the ME is not yet achieved.

As we can observe from Table II, the handwriting modality based system gives the right interpretation of the input signal in 34.1% of cases. If the first two answers are considered, 8% more are saved and if ten best solutions are taken into account from the output of this system we reach a recognition rate of 47.68%.

[1] http://www.isical.ac.in/~crohme/index.html

This observation combined with the performance with respect to the speech modality suggests that the combination of both modalities should increase the $TOP1$ recognition rate obtained with the handwriting based system alone. In the following the results of this procedure are reported.

C. The proposed system performances

Table III reports the comparison of the handwritten mathematical expressions recognition system and the bimodal based one, considering the fusion at symbols and relations levels [12] and considering the approach proposed here.

TABLE III. COMPARISON OF THE PERFORMANCES OF THE HANDWRITING RECOGNITION SYSTEM WITH THE FUSION BASED SYSTEM PROPOSED IN [12](SYST. I) AND THE ONE PROPOSED HERE (SYST. II)

Recognition rate in [%] of	Strokes	Symbols	Expressions with		
			Exact match	1 error at most	2 errors at most
Handwriting based system	80.05	82.93	34.10	46.44	49.52
Syst. I	86.73	88.21	41.82	50.67	53.37
Syst. II	86.65	89.30	42.00	51.06	52.02

In Table III we can observe that the system we proposed here (Syst. II) outperforms significantly the baseline system based on handwriting signal and only slightly the multi-modal system we proposed in [12] where the fusion is achieved at symbols and relations levels. It is clear that the bimodal aspect of the information allows not only to improve the recognition at the expression level, but also at the lower levels (strokes and symbols).

With the proposed approach (Syst. II), every solution that is in the handwritten $Nbest$ list is treated equally with respect to the audio transcript. In another words, the last proposal of the list could be selected if its similarity measure is the best one with respect to the audio transcript, even if the initial cost of this solution is very high compared to the $TOP1$ solution. To prevent this situation, we propose a variant of the proposed method, using a reject threshold to possibly shorten the NBest list.

In this regard, the new $Nbest$ list, denoted $Nbest'$ is given by equation 3:

$$Nbest' = \{Topj \in Nbest \ / \ \frac{|C_1 - C_j|}{C_1} \leq \alpha\}, \qquad (3)$$

where α is a parameter we fixed experimentally to 1.3 using the validation database. The variables C_1 and C_j are the initial costs (given by the handwriting based system) associated with the $Top1$ and $Topj$ MEs respectively.

Therefore, every solution which has a relative cost more than alpha times the $Top1$ solution will be discarded from the list.

The obtained results with this system (Syst II') are reported in the Table IV. It shows that the use of restricted solutions with too low relative costs improves the performances at all levels (stroke, symbol and ME). Here, the interesting point is

that the gain in term of ME recognized with one or two errors is very significant compared to the previous systems (Syst. I and Syst. II).

TABLE IV. COMPARISON OF THE PERFORMANCES OF THE HANDWRITING RECOGNITION SYSTEM WITH THE FUSION BASED SYSTEM PROPOSED IN [12](SYST. I) AND THE EXTENSION OF THE ONE PROPOSED HERE (SYST. II')

Recognition rate in [%] of	Strokes	Symbols	Expressions with		
			Exact match	1 error at most	2 errors at most
Handwriting based system	80.05	82.93	34.10	46.44	49.52
Syst. I	86.73	88.21	41.82	50.67	53.37
Syst. II'	87.13	90.83	42.97	53.09	57.34

The improvement brought by the current method with respect to the previous bimodal system (Syst. I) is not necessarily important, however it gives another point of view of where the fusion can happen (at the ME interpretation level). In addition, there are more and more MEs which are recognized with one or two errors than compared to Syst. II and Syst. I.

VI. CONCLUSION AND FUTURE WORK

In this work we presented a new approach for bimodal mathematical expressions recognition. The modalities in concern are speech and handwriting.

The main novelty of this work is to consider the combination process during the interpretation step. This procedure allows to prevent from the problem of the existing asynchrony between both streams during processing at lower levels (symbols and elementary relations).

The reported results showed the interest of such a processing. This can be seen either at expression level and in lower levels (strokes and symbols).

In a future work, as a first extension of the Syst. II', we plan to use both handwriting costs and similarity measures in a global cost function to give the final interpretation. We are also planning to exploit the two strategies of fusion we investigated (Syst. I and Syst. II') in order to tend to a more complete system where the bimodal information is exploited during symbols/relations identification and during the interpretation.

REFERENCES

[1] B. Dorothea and G. Ann, *Recognition of mathematical notation*, H. Bunke, P. Wang ed. Handbook of Character Recognition and Document Image Analysis, 1997.

[2] E. Tapia and R. Rojas, "A survey on recognition of on-line handwritten mathematical notation," Free University of Berlin, Tech. Rep., 2007.

[3] K. F. Chan and D. Y. Yeung, "Mathematical expression recognition: A survey," *International Journal of Document Analysis and Recognition*, vol. 3(1), pp. 3–15, 2000.

[4] T. H. Rhee and J. H. Kim, "Robust recognition of handwritten mathematical expressions using search-based structure analysis," in *Proc. of Int. Conf. on Frontier in Handwriting Recognition (ICFHR)*, 2008, pp. 19 – 24.

[5] A.-M. Awal, H. Mouchère, and C. Viard-Gaudin, "A global learning approach for an online handwritten mathematical expression recognition system," *Pattern Recognition Letters*, no. 35, pp. 68–77, 2014.

[6] R. Fateman, "How can we speak math," *Journal of Symbolic Computation*, vol. 25, no. 2, 1998.

[7] A. Wigmore, G. Hunter, E. Pflugel, J. Denholm-Price, and V. Binelli, "Using automatic speech recognition to dictate mathematical expressions: The development of the talkmaths application at kingston university," *Journal of Computers in Mathematics and Science Teaching (JCMST)*, vol. 28(2), pp. 177–189, 2009.

[8] P. Deléglise, Y. Estève, S. Meignier, and T. Merlin, "Improvements to the lium French ASR system based on cmu sphinx: what helps to significantly reduce the word error rate?" in *Proc. of Int. Conf. Interspeech*, 2009, pp. 2123–2126.

[9] "Cmu sphinx system," http://cmusphinx.sourceforge.net, Accessed on February, 20th, 2014.

[10] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli, "Multimodal fusion for multimedia analysis: A survey," *Multimedia Systems*, vol. 16(6), pp. 345–379, 2010.

[11] J. P. Thiran, F. Marqués, and H. Bourlard, *Multimodal Signal Processing - Theory and Applications for Human-Computer Interaction*. Elsevier, 2010.

[12] S. Medjkoune, H. Mouchere, S. Petitrenaud, and C. Viard-Gaudin, "Multimodal mathematical expressions recognition: Case of speech and handwriting," in *Human-Computer Interaction. Interaction Modalities and Techniques*, ser. Lecture Notes in Computer Science, M. Kurosu, Ed. Springer Berlin Heidelberg, 2013, vol. 8007, pp. 77–86.

[13] S. Quiniou, H. Mouchère, S. Peña Saldarriaga, C. Viard-Gaudin, E. Morin, S. Petitrenaud, and S. Medjkoune, "HAMEX - a handwritten and audio dataset of mathematical expressions," in *Proc. of Int. Conf. on Document Analysis and Recognition (ICDAR)*, 2011, pp. 452–456.

[14] H. Mouchère, C. Viard-Gaudin, D. H. Kim, J. H. Kim, and U. Garain, "ICFHR2012: Competition on recognition of online handwritten mathematical expressions (crohme 2012)," in *Proc. of Int. Conf. on Frontier in Handwriting Recognition (ICFHR)*, 2012, pp. 811–816.

IJDAR (2016) 19:173–189
DOI 10.1007/s10032-016-0263-5

ORIGINAL PAPER

Advancing the state of the art for handwritten math recognition: the CROHME competitions, 2011–2014

Harold Mouchère[1] · **Richard Zanibbi**[2] · **Utpal Garain**[3] · **Christian Viard-Gaudin**[1]

Received: 6 August 2015 / Revised: 16 December 2015 / Accepted: 12 February 2016 / Published online: 16 March 2016

Abstract The CROHME competitions have helped organize the field of handwritten mathematical expression recognition. This paper presents the evolution of the competition over its first 4 years, and its contributions to handwritten math recognition, and more generally structural pattern recognition research. The competition protocol, evaluation metrics and datasets are presented in detail. Participating systems are analyzed and compared in terms of the central mathematical expression recognition tasks: (1) symbol segmentation, (2) classification of individual symbols, (3) symbol relationships and (4) structural analysis (parsing). The competition led to the development of *label graphs,* which allow recognition results with conflicting segmentations to be directly compared and quantified using Hamming distances. We introduce *structure confusion histograms* that provide frequencies for incorrect subgraphs corresponding to ground-truth label subgraphs of a given size and present structure confusion histograms for symbol bigrams (two symbols with a relationship) for CROHME 2014 systems. We provide a novel analysis combining results from competing systems at the level of individual strokes and stroke pairs; this virtual merging of system outputs allows us to more closely examine limitations for current state-of-the-art systems. Datasets along with evaluation and visualization tools produced for the competition are publicly available.

Keywords Handwriting recognition · Mathematical expression recognition · Competitions · Performance evaluation

✉ Harold Mouchère
harold.mouchere@univ-nantes.fr

Richard Zanibbi
rlaz@cs.rit.edu

Utpal Garain
utpal@isical.ac.in

Christian Viard-Gaudin
christian.viard-gaudin@univ-nantes.fr

[1] LUNAM/University of Nantes/IRCCyN, rue Christian Pauc, 44306 Nantes, France

[2] Department of Computer Science, Rochester Institute of Technology, 1 Lomb Memorial Drive, Rochester, NY, USA

[3] Computer Vision and Pattern Recognition Unit (CVPRU), Indian Statistical Institute, 203, B.T. Road, Kolkata 700 108, India

1 Introduction

Research in automatic recognition of online handwritten mathematical expressions dates back to the 1960s. In online recognition, Anderson [1] developed an attributed context-free grammar for recognizing handprinted math expressions written on an input device similar to a tablet. After this initial attempt, several researchers have studied this problem at different paces. Significant research effort has been reported in the last 15–20 years, in part due to online input devices becoming more popular. Despite this increase in research activity, estimating progress became quite difficult mainly because of the lack of available benchmarking datasets and variety of evaluation metrics in use. Systems were evaluated primarily using private author-generated datasets which were not publicly available. Being unable to reproduce the results of others, researchers could not clearly judge their progress. This motivated the organization of a competition, which came to be named *CROHME: the Competition on Recognition of On-line Handwritten Mathematical Expressions* [2–5].

Since CROHME's inception, researchers have been gradually attracted toward the event. The number of participating

systems increased from five to eight in 4 years. Industrial research groups showed interest in and participated in CROHME. Participating systems have used several methodologies, and CROHME has provided a nice platform for comparing these methods and bringing out the relative merits of individual approaches. Each CROHME developed an original test dataset, which is separated from a training dataset. Evaluation was done by the organizers and subsequently checked by participating teams for consistency and to correct minor errors (e.g., in output file formats). In this way, CROHME has documented progress in the area, and the concerned research community is now aware of advances in terms of shareable resources, capabilities of different methods and evaluation results. So far, CROHME has influenced a particular community which had been conducting their research in a sporadic manner to work using a more consistent and scientific approach.

In this paper, we present the evolution of CROHME over the last 4 years, along with its contributions to handwritten math recognition research and related areas. These include the datasets, different tasks, evaluation metrics, tools, participating systems, and finally, analysis of results using both standard and novel methods. Preparation of CROHME data involves several issues including the number of allowable symbols, two-dimensional structures, and coding of data. The CROHME datasets started with a relatively simple set of samples in CROHME-2011, gradually adding more complicated expressions over the next 3 years. Figure 1 shows some real samples from the last CROHME training set which show the difficulty of the recognition tasks: symbol segmentation, symbol recognition, spatial relation recognition and expression parsing.

While the complexity of expressions to recognize increased with each competition, the recognition of complete expressions is very difficult. As a result, the competition tasks were modified to consider sub-tasks. This gave research groups options to participate in the event as per their convenience. For example, recognition of matrices (being a difficult task) was introduced as a new task in the 4th year, along with a separate competition for isolated symbol recognition.

Expression recognition results can be evaluated in various ways, and evaluation metrics have undergone a number of changes in the past 10–15 years. Initially, simple measures like number of correctly recognized symbols or structures were used [6]. Researchers found these measures insufficient for characterizing local recognition errors, and subsequently, tree-matching-based methods [7], and later label graph-based evaluation developed for CROHME [8] were introduced. Along with the competition tasks, the CROHME evaluation metrics evolved over time.

In addition to a detailed discussion of the main features of CROHME competitions, we will address the following questions. (1) Why is mathematical expression recognition a difficult problem? (2) Is it possible to specify which expressions are more complex than others? (3) How does one build a representative corpus of expressions? (4) How do we detect the weakness and specific errors made by a recognition system? (5) Can we merge results from different recognition systems? (6) As mathematical expressions are a two-dimensional language, can we detect which symbol configurations are more susceptible to being recognized incorrectly?

The structure of this paper is as follows: Section 2 describes the CROHME competition (tasks, corpus, datasets and protocol). Section 3 discusses evaluation metrics used for the competitions. Section 4 presents the participating systems and the methodologies systems use. New evaluation results and analysis of the results are presented in Sect. 5. Section 6 outlines the impact of the competition and issues to be addressed in future.

2 The CROHME competition

In this section, we present how the CROHME competitions have been organized. After defining the different tasks in the first subsection, Sect. 2.2 explains how we chose and collect the math expressions. Then, Sect. 2.3 analyses the datasets. The last subsection explains the submission and testing protocol of the competition.

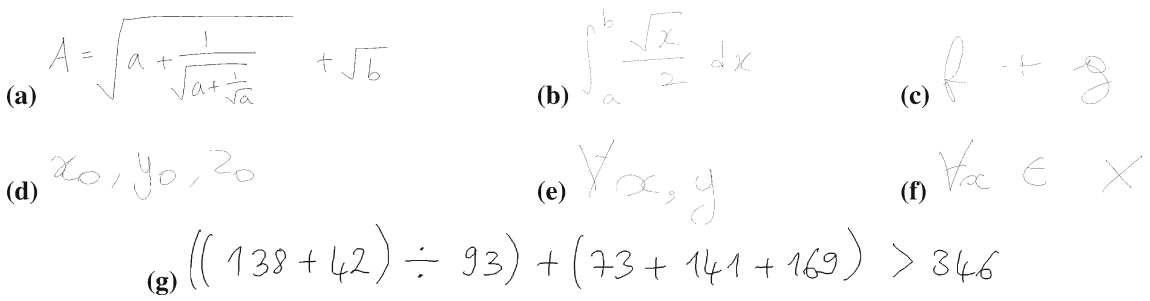
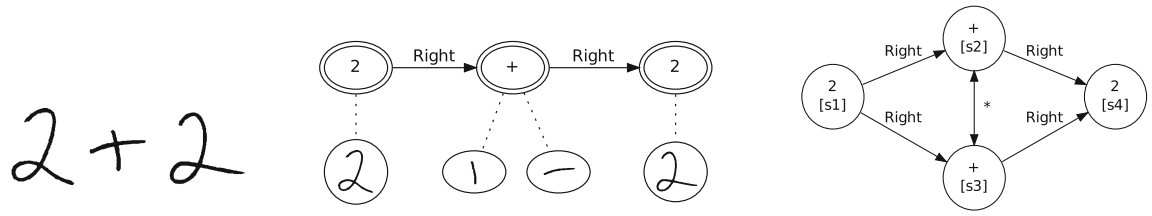


Fig. 1 Handwritten formulae from the CROHME training dataset



(a) Expression **(b)** Symbol layout tree **(c)** Stroke label graph

Fig. 2 A simple handwritten expression (**a**) and its interpretation as represented by a *symbol layout tree* (**b**) and *stroke label graph* (**c**). The expression contains four strokes, labeled $s1$–$s4$ in time order. The symbol layout tree represents spatial relationships between symbols, here with handwritten strokes associated with each node shown by dotted edges. The stroke label graph represents the same information, using strokes rather than symbol as nodes, and using both relationship (e.g., *Right*) and stroke merge (*) labels on edges

2.1 CROHME task definition

2.1.1 Mathematical notation

It is difficult to define mathematical notation precisely [9–11]. [1] The precise semantics of a formula depends on its domain of use, and the way in which an author chooses to define, and possibly re-define parts of their notation. In this way, mathematical notation is highly dialectical and is in essence a form of *natural* visual language [12].

For the CROHME competitions, samples of formulae from algebra and calculus have been used, which is traditional in math recognition research. We assume all input strokes belong to exactly one symbol —this removes symbols connected by a ligature (e.g., '$2x$' written with one stroke). Function names are treated as individual symbols (e.g., cos, sin, tan, lim). Figure 2a shows a handwritten expression, along with a *symbol layout tree* representing the spatial relationships between symbols in Fig. 2b. The correspondences of handwritten strokes to symbols in the layout tree are shown using dotted edges. Symbol layout trees have the leftmost symbol on the main writing line at the root of the tree, with symbols (nodes) connected by spatial relationships. For CROHME, spatial relationships include *Right, Above, Below, Superscript, Subscript* and *Inside* (for square roots). LaTeX and Presentation MathML[2] encode symbol layout trees without the correspondence of strokes to symbols. We call these *symbolic* encodings. Allowable expressions for different tasks are defined using LaTeX string grammars as described below.

2.1.2 Input and output

Stroke data is provided to participants in CROHME InkML (XML) files. Each stroke has a unique identifier and sequence of (x, y) locations. Participants must parse CROHME InkML files with stroke data and then return one of two output formats, depending upon the competition instance. For CROHME 2011–2013, a CROHME InkML file was generated that contained the symbol layout tree in Presentation MathML, along with annotations for the correspondence of symbols to stroke groups. In CROHME 2013–2014, Comma Separated Variable (.csv) files for *label graphs* (described below) could be returned instead.

2.1.3 CROHME competition tasks

Math notation is domain specific, and the difficulties of recognizing expressions depend upon which symbols and symbol relationships are used. For the competition, we define several difficulty levels, from simple expressions on a single writing line (e.g., '$x + 1$') to matrix expressions. Each level is called a *task* defined by a context-free grammar for a set of legal LaTeX expressions. These grammatical definitions mean that the expression sets are infinite, which is one difficulty of ME recognition. To parse LaTeX strings, we use the PEP[3] parser which is a free implementation of the Earley algorithm. Table 1 shows the increasing difficulty of the five tasks. As can be observed, the symbol set has gradually increased in size and now includes 101 symbols. This enables a good coverage of different scientific domains, while at the same time increasing recognition difficulty due to more easily confused classes, such as for digits '0' and '1' with letters '*O*' and '*l*.' Furthermore, the allowable symbol layouts have increased over time. Only six atomic spatial relationships between two symbols are used (Above, Below, Superscript, Subscript, Inside (for square roots), Adjacent at Right), but allowing additional structures increases the difficulty of the task by increasing the number of legal expressions. For example, in task 1 nested fractions and superscripts on function names are not allowed, but in the task 2 these constraints are removed. Table 1 shows the number of production rules used to define the LaTeX grammar. The increasing num-

[1] A well-written history of mathematical notation is available [11].

[2] http://www.w3.org/Math/.

[3] http://www.ling.ohio-state.edu/~scott/#projects-pep.

Table 1 CROHME expression grammars (tasks)

Year	Grammar	Symbols	# P	Additions (with examples)
2011	1/I	36 symbols :	38	No nested exprs. in fractions or sub/superscript
		$abcdeiknxyz0123456789\phi\pi\theta$ $+ - \pm \sin\cos \neq \leq > = () \surd$		$x^2 + y^2 > 1 \qquad \sqrt{b^2 - 4ac}$
	2/II	56 symbols, 20 added:	60	No recursion limits; complex structures included
		$ABCFj\alpha\beta\gamma\infty$ $\div \times \sum \log\tan \ldots \geq \rightarrow \lim \int !$		$\sqrt{1+\frac{1}{\sqrt{2}}}+\sqrt{1-\frac{1}{\sqrt{2}}}$ $\lim_{x\rightarrow\frac{\pi}{2}+0}\tan x = -\infty$
2012	3/III	75 symbols, 19 added:	95	Set operators and brackets
		$\{\}[]XY < tfgmrp/, .\exists\forall \in$		$\forall x \in X \quad \left[\frac{2}{3}x^{\frac{3}{2}}\right]_0^1$
2013	4/IV	101 symbols, 26 added:	155	nth-root
		$EGHILMNPRSTV\ hloqsuvw\vert'\sigma\Delta\lambda\mu$		$\sqrt[4]{648+648}+8$
2014	matrix/IV-matrix	101 symbols	168	Matrices within and containing expressions
				$A = \begin{pmatrix} 3 & 1 \\ 4 & 0 \end{pmatrix} \quad \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$

Year competitions where grammar/task was introduced, generally the tasks are used during two consecutive years; *Grammar* grammar/task identifier used in the competitions; *Symbols* number and list of used symbols in the corpus; *#P* number of production rules; *Additions* expressions added to the corpus

ber of rules reflects a higher specificity in the modeling of the language than an increase of the complexity of the language. Nevertheless, allowing more spatial relationships and symbols makes expressions more diverse and complex. The task grammars are provided in a package available with the CROHME dataset from the TC11 Web site.

2.2 Corpus construction

Before CROHME existed, researchers used to experiment with their own private datasets. Hence, fair comparisons between systems were hardly possible. So, one outcome of the CROHME project was to deliver high-quality datasets. The main issues to build a dataset are related to relevance, completeness, and correctness of the data. Specifically for the domain of mathematical expressions many parameters are involved: set of symbols (size, identity) and their distribution; corpus composition, extracted from a realm of discourse (algebra, thermodynamic, mechanic. . .); number of writers, number of expressions written per writer and number of times a given expression is written.

CROHME organizers used existing resources from different laboratories already working on ME recognition. In 2011 and 2012, the organizers merged several existing datasets from different laboratories: CIEL [13] and HAMEX [14] (University of Nantes), MfrDB [15] (Czech Technical University), ExpressMatch [16] (University of Sao Paulo), datasets from KAIST and CVPR/ISI and in 2013 MathBrush [17] (University of Waterloo). The expressions (stroke data and ground truth) coming from these datasets have been used directly by doing file format conversion and adding missing ground truth.

As explained in the introduction, these existing datasets suffer from a lack of representative situations. For example, in the MathBrush dataset [17] the expressions are generated randomly from a grammar. In [16], a small corpus of 56 expressions has been written by 25 writers. In [13], only 36 different expressions are selected to cover different scientific domains; then, each expression is written by different writers. In each such case, the final corpus is not representative of real usage.

Since CROHME 2013, we have designed a corpus better reflecting the current use of mathematical expressions. Scientific books or courses are interesting sources with lots of ME, but mostly, it is very domain specific, using only a subset of symbols and relations. Instead, we used the Wikipedia French pages and the Wikibooks English pages as a source of MEs. In Wikipedia pages, the math expressions are delimited by 'math' tags. We have extracted more than 164,000 ME, which are our pool of expressions. In addition, some filtering was introduced to control the content of the corpus:

- removing duplicate expressions: popular expressions such as $\sin^2 x + \cos^2 x = 1$ will be present on several pages.
- controlling LaTeX string length: all expressions will have between 3 and 50 LaTeX symbols. For example, the two following strings 'x^2' and 'x_i,' have a length of 3, but with only two printable symbols.
- defining a valid LaTeX symbol set: a list of acceptable LaTeX symbols and commands is defined.

- validating with a grammar parsing tool: a LaTeX grammar is defined and only successfully parsed expressions are accepted in the corpus.
- controlling the symbol frequency: the symbol frequency in the test set should be the same as in the corresponding training set.

In a real context, the symbol frequency is domain dependant. In CROHME, the symbol frequency in the test sets is made similar to the training sets to be representative. In the first years of the competition (2011 and 2012), the expressions were selected in the available datasets considering only the presence of selected symbols. It was sufficient because the number of the different symbols was low and they corresponded to the most frequent ones; so the subsets (training and test) were already balanced. However, when the number of symbols increases, we are getting closer to the tail of the distribution and when expressions are chosen among a huge set (like Wikipedia or Wikibook corpus) the choice cannot be completely randomized. As a result, in the test set of task 3 in 2012, most expressions have been randomly chosen from the Wikipedia FR corpus, but some expressions have been added manually to balance the symbol frequency.

In a more systematic way, since CROHME 2013, an algorithm has been used to build a corpus with the same frequency of symbols as in a reference corpus. We used an iterative algorithm which compares the current frequency of each term in the corpus under construction with regards to the targeted frequency of the same term in the reference corpus. At each iteration, the algorithm sorts with decreasing costs the candidate expressions from a pool of expressions. The cost $C_{c,r}(e)$ of a candidate expression (e), Eq. 1, has been defined as the sum of its term costs. A term cost $\text{cost}_{c,r}(t_i)$, Eq. 2, being defined as the negative log ratio of frequency terms t_i in the current dataset c with regard to the reference dataset r. $f_d(t_i)$ is the term frequency of a symbol t_i in d, which can be an expression or a full corpus.

$$C_{c,r}(e) = \sum_{t_i \in e} \text{cost}_{c,r}(t_i) \tag{1}$$

$$\text{cost}_{c,r}(t_i) = \log(f_r(t_i)) - \log(f_c(t_i)) \tag{2}$$

Thus, a candidate expression which contains many symbols that are underrepresented in the current corpus will be ranked before a candidate expression which contains many symbols that are overrepresented. A first version of this protocol has been defined in 2013 and the presented version has been used in 2014. Note that this approach can be extended to use the frequencies of other criteria such as the equation length or the spatial relation types. In 2014, the selection takes also into account the size of expressions: The size value is processed as other symbols, its frequency is used to compute the cost $C_{c,r}(e)$ of an expression.

Once the expressions are selected using the previous criteria, they are written once by writers from the different organizing laboratories (University of Nantes, RIT, ISI).

The next section presents several statistics to illustrate the diversity of the dataset and the coverage of the test set with regard to the training set.

2.3 Dataset properties

A detailed analysis of the corpus used in the main task (task 4) of the competition is provided in this section. It compares the frequencies of each symbol, of each spatial relation and the complexity of the expressions using different indicators across the training subsets and the test sets (2013, 2014). Note that the training part has been the same in 2013 and 2014 competition, only the test sets have been renewed. Thus, we compare the statistics from the training part to test parts 2013 and 2014.

Table 2 shows the frequencies of some symbols in the test sets in 2013 and 2014 compared to the training set. We can see that some symbols are very frequent (like '−,' '1' and '+') and other are very rare (like '∈,' '∀' and '∃') in all sets. Even if the term frequencies are quite well respected in both test sets, 2014 test better respects the proportions of the different symbols with regard to the training set.

The symbol frequencies do not allow to evaluate the complexity of the different ME sets. Mainly the complexity of an expression is based on the complexity of its MathML tree, that is why we present in Fig. 3 some statistics extracted from the MathML trees of the three sets:

- the maximum depth of the trees ignoring the `<mrow>`[4] elements, defined as degree of nestedness DoN in [18]: it represents how nested are the expressions (superscript in a superscript or fraction of subscripted symbols...), a depth of 0 is a one line expression;
- the sum of baselines: It counts the number of times a sub-expression is not on the same baseline as its mother expression, for example $x^2 + y^2$ and x^{y^2} and $\frac{x}{y}$ have 3 baselines;
- the number of distinct baselines, defined as geometric complexity GC in [18]: In the previous metric, some baseline can have the same level, and this metric counts the number of different levels which are used in the full expression, e.g., $x^2 + y^2$ uses 2 distinct baselines but x^{y^2} and $\frac{x}{y}$ use 3 distinct baselines;
- the spatial relation frequency: The previous metrics are quite independent of the nature of the spatial relations,

[4] The MathML `<mrow>` element is used to group sub-expressions, which usually contain one or more operators with their respective operands. This element renders as a horizontal row containing its arguments.

Table 2 Symbol frequencies for training and test sets of Task 4 in CROHME 2013 and 2014. Symbols are sorted by decreasing frequency in the training set. Only most and least frequent symbols are shown

	Train 2013/2014	Test 2013	Test 2014
–	7940 (9.254 %)	440 (7.233 %)	910 (9.083 %)
1	6219 (7.248 %)	314 (5.162 %)	721 (7.196 %)
2	6195 (7.220 %)	338 (5.556 %)	715 (7.136 %)
+	5409 (6.304 %)	267 (4.389 %)	622 (6.208 %)
x	5042 (5.876 %)	261 (4.291 %)	587 (5.859 %)
(	3945 (4.598 %)	295 (4.850 %)	458 (4.571 %)
)	3939 (4.591 %)	294 (4.833 %)	458 (4.571 %)
=	3611 (4.209 %)	319 (5.244 %)	434 (4.332 %)
a	2475 (2.885 %)	137 (2.252 %)	279 (2.785 %)
3	2458 (2.865 %)	117 (1.923 %)	289 (2.885 %)
n	2239 (2.610 %)	140 (2.301 %)	267 (2.665 %)
0	1795 (2.092 %)	128 (2.104 %)	214 (2.136 %)
$\sqrt{}$	1793 (2.090 %)	86 (1.414 %)	213 (2.126 %)
y	1765 (2.057 %)	82 (1.348 %)	225 (2.246 %)
4	1635 (1.906 %)	77 (1.266 %)	183 (1.827 %)
b	1599 (1.864 %)	81 (1.332 %)	185 (1.846 %)
z	1074 (1.252 %)	49 (0.806 %)	120 (1.198 %)
d	1063 (1.239 %)	89 (1.463 %)	119 (1.188 %)
5	1003 (1.169 %)	49 (0.806 %)	122 (1.218 %)
...			
{	69 (0.080 %)	6 (0.099 %)	7 (0.070 %)
}	69 (0.080 %)	6 (0.099 %)	7 (0.070 %)
>	56 (0.065 %)	9 (0.148 %)	7 (0.070 %)
σ	52 (0.061 %)	14 (0.230 %)	13 (0.130 %)
μ	46 (0.054 %)	17 (0.279 %)	7 (0.070 %)
Δ	35 (0.041 %)	21 (0.345 %)	5 (0.050 %)
λ	27 (0.031 %)	4 (0.066 %)	7 (0.070 %)
$\in$	14 (0.016 %)	9 (0.148 %)	3 (0.030 %)
$\forall$	8 (0.009 %)	3 (0.049 %)	2 (0.020 %)
$\exists$	4 (0.005 %)	2 (0.033 %)	4 (0.040 %)

and this last metric counts the number of time each spatial relation is used.

For each of these 4 statistics, the frequency is showed (using a log scale) by normalizing the counter by the number of expressions in the corresponding dataset. We can see that about one-third of the datasets are simple one line expressions (max depth of 0 and number of baseline or distinct baseline equal to 1 for 33 % of expressions in the training set, 25 % in test set 2013 and 30 % in test set 2014). Then half of expressions are not nested expressions (max depth of 1 for 45, 58 and 53 % of expressions). Probably these expressions are covered by the expressions with 2, 3 or 4 baselines (from 42 to 57 % of the expressions) and 2 or 3 distinct baselines (from 43 to 56 % of expressions). Furthermore, the tails of the histograms have been cut; thus, some expressions which are not shown here are very complex (about 2–5 %) with a max

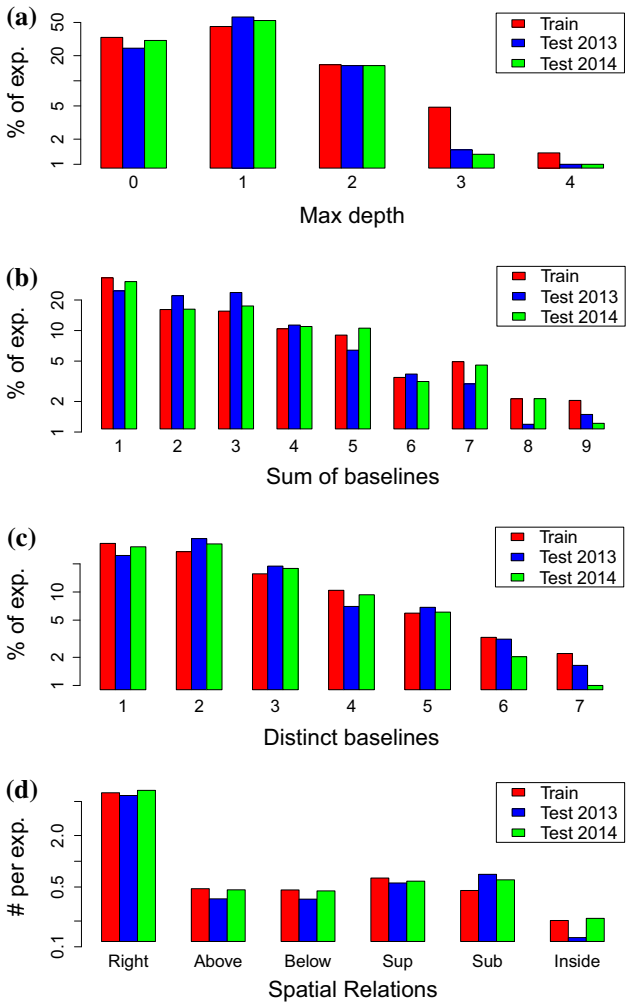


Fig. 3 Complexity statistics for training (*red*) and test sets for Task 4 in CROHME 2013 (*blue*) and 2014 (*green*). In **a**–**c**, maximum symbol layout tree depths, total number of baselines ('sum of baselines') and distinct baseline histograms are presented in order of increasing complexity (i.e., tree depth and number of baselines). Spatial relationship frequencies are presented in terms of the average number of occurrences per expression. Log scale is used for the *y*-axes (colour figure online)

depth of more than 4, more than 9 baselines and more than 8 distinct baselines. Finally, the spatial relation frequencies show that the most frequent relation is *Right* with 6 relations per expression in average, then *Superscript* (about 0.5 superscript per expression) followed by the *Subscript* relation (also about 0.5 per expressions) and *Above* and *Below* (0.4 per expressions). The relation *Inside* used only in square roots is rarer with 0.2 per expression.

2.4 Competition protocol

Each year there was something new in the competition (a new task, a new evaluation tool or a new file format); thus, we tried to give enough time to participants to submit a system taking into account these updates. Having enough time was a key

point because the training and testing data have to be created for the competition and this is a time-consuming process.

After the call for participation, we provide the registered participants with the evaluation tools, the file formats, the task definitions (i.e., the grammars and symbol lists) and an initial training set. During the next months, we eventually update the training set depending on our tests and the participant feedbacks. Then the participants have to submit a first draft system about four months after the call. This first submission allows to test the systems in real conditions and to fix some technical issues (mostly OS configurations or file format problems). Detailed results on the training set were sent back to participants to detect the problems. The final deadline is about 5 months after the call for participation. Then the participants have to submit their final systems, one per task. During the next 2 weeks, the organizers run each participating system on each test dataset. After the competition, the test data are available on demand and submitted on TC11 Web site.

This protocol requires a huge effort from the organization committee, but it has several advantages. Firstly it allows good collaboration and discussion between participants and organizers as several exchanges are necessary. Furthermore, this protocol is quite fair as no participant had access to the test set before the final submission (except for the organization teams). This protocol allows also organizers to correct some minor issues in the test data or evaluation tools after the final submissions, e.g., correct ground-truth errors in the test set or add new metrics and features in the evaluation tools.

3 Evaluation metric and system rankings

Evaluation of structural pattern recognition systems is often difficult because of the interaction between detected objects and their relationships. For example, when a relationship between one correct and one incorrectly segmented symbol is detected, how this should be evaluated? Obviously the relationship is incorrect because one of the symbols is incorrect, but can we quantify partially correct structure somehow? To address this question, a number of new metrics and a new structure representation were developed for CROHME. In particular, new stroke-based metrics allow partially correct recognition results to be located and measured precisely [8,19]. We discuss these metrics below.

3.1 LaTeX, MathML, CROHME InkML and symbol-level evaluation

3.1.1 Expression and structure recognition rates

Prior to CROHME, it was common to evaluate math recognizer performance using expression recognition rates, e.g., the percentage of LaTeX formula strings matching ground truth. The metric is simple to understand and provides a useful global performance metric. Expression rate has been used to rank participating systems in all of the CROHME competitions, first using canonicalized Presentation MathML (ensuring that identical expressions with different MathML representations match), and later label graphs, taking symbol segmentation into account (see the next Section). However, an absolute expression rate does not characterize partially correct recognition.

Previous attempts to quantify partially correct recognition in symbolic encodings have been made. The EMERS [7] metric is a tree edit distance using an Euler string representation for MathML trees. Symbol errors are weighted based on their distance from the main writing line (baseline) of the expression. The IMEGE metric [20] is an image-based comparison of LaTeX, using the number of matching pixels in a rendered LaTeX string with the associated ground truth LaTeX rendering. To prevent missing valid sub-expressions that have been 'shifted' in the rendered image, small image distortions are permitted.

For CROHME, a simpler and faster-to-compute metric was used to count partially correct expressions. For each system, we produced a list giving the percentage of expressions with matching MathML trees but with at most n incorrect symbol and relationship labels. The rate for $n = 0$ is the percentage of test expressions correctly recognized, followed by expression rates for increasing numbers of labeling errors. For expressions where the node and edge structure of the layout trees do not match, no number of relabelings can correct the expression. For example, '$2x_{+1}$' has two label disagreements with '$2x + 7$' (the subscript for '+,' and '1' not matching '7'), but these differences can be corrected with two relabeling operations. Conversely, the expression '2^x+1' cannot match '$2x + 7$' because the '2' has a different number of child nodes. Expression rates for $n \leq 3$ symbol and relationship label errors are reported in each CROHME competition.

For CROHME 2012, we also proposed a new formula structure (*STRUCT*) expression rate, which matches layout tree structure, ignoring symbol labels and stroke segmentation. For example, if the expression '$x^2 - 1$' is recognized as '$2^a + b$,' it will be considered as correct. As another example, recognizing the expression in Fig. 2 as '5×5' will also be considered correct. In this way, formula structure detection can be evaluated separately from symbol segmentation and classification.

3.1.2 Symbol and relationship detection

The correspondence between strokes and symbols provided in CROHME InkML files allows us to compute the percentage of target symbols correctly segmented (*SYM_Seg* in CROHME 2011–2013) and the number of correctly seg-

mented and classified target symbols (*SYM_Rec* in CROHME 2011–2013). These are measures of symbol *recall*. One can also measure the *precision* of recognized symbols, which is the percentage of correctly detected or detected and classified symbols. During CROHME 2014, we realized that symbol relationships can also be evaluated using recall and precision, measured over pairs of symbols with a relationship and their associated strokes. Relationships are detected correctly if both symbols in the relationship are correctly segmented and have a defined relationship in ground truth, and correctly detected and classified if the relationship labels also agree.

3.2 Label graphs and stroke-based metrics

3.2.1 Label graphs

Beginning with CROHME 2013 a second structural representation called *stroke label graphs* was used [19], as illustrated in Fig. 2c. A label graph defines structure using a directed graph over handwritten strokes. Strokes (nodes) are labeled by their associated symbol, and directed edges between stroke pairs are labeled by either a spatial relationship between two symbols (e.g., 'Right'), no relationship ('_') or a pair of directed edges for strokes belonging to the same symbol ('*'). For legibility, 'no relationship' edges are omitted in Fig. 2c, but Fig. 2c represents a *complete* graph with every nodes and directed edge labeled. An adjacency matrix over strokes with labels may be used [8]; the number of labels in a label graph is given by n stroke labels, plus $2\binom{n}{2}$ directed edge labels, giving n^2 labels in total. In Fig. 2c, $4^2 = 16$ labels are defined.

Every possible symbol layout tree for a set of input strokes can be represented by a label graph, along with partial and illegal interpretations (e.g., a forest of symbol layout trees). Symbol-level evaluation metrics may also be computed directly from label graphs [8]. Symbols are defined by cliques of '*' (merge) labels or labels that match labels for attached strokes—we use '*' edges in Fig. 2. Using these stroke cliques, we can recover the symbol layout tree from a label graph after recovering symbols. For efficiency, in our tools we use connected component analysis over stroke labels to identify the grouping of strokes into symbols.

Hierarchical structure can be represented using sets of labels for nodes and edges, where labels for each level of structure are disjoint [5,8]. One can then filter for labels from a specific 'level.' This is how we obtained results at different levels for the matrix recognition task in CROHME 2014. Label graphs may be easily combined and/or filtered in various ways, as we do to identify which expressions can be correctly recognized if all correct node and edge labels from participant systems are joined in Sect. 5. Non-hierarchical structure can also be represented with label graphs.

3.2.2 Label hamming distances

We can use label graphs to compute *exact* disagreement for input strokes between two interpretations. This resolves the problem with symbolic encodings such as LaTeX being unable to put disagreeing segmentations at the symbol level into correspondence. We define Hamming distances for stroke labels (ΔC), conflicts of segmentation at individual edges (ΔS, where only one of two graphs has a 'merge strokes' label) and edge relationship-type conflicts (ΔR). The absolute (total) Hamming distance ΔB is the sum of these component label errors ($\Delta B = \Delta C + \Delta S + \Delta R$). We also defined a variation that weights segmentation edges lower to compensate for their frequency (ΔE).

3.2.3 Stroke-level error analysis

Label graphs permit highly detailed error analyses: Now, when a symbol is mis-segmented by a system, we can determine precisely which strokes were grouped correctly or incorrectly for the target symbol. This additional information allows us to make new analyses such as that presented in Sect. 5.3, where we enumerate small subgraphs in ground truth (e.g., for pairs of related symbols) and then count the number of times different errors are made for each. We can now count and visualize incorrect label graphs representing any combination of classification, segmentation and relationship mis-labelings for strokes in a given target subgraph.

4 Participating systems

In this section, we provide a summary of techniques that have been used in the CROHME competition [2–5]. Results from different systems for the CROHME 2014 competition are discussed in the next Section. For brevity, the identifiers given in Table 3 are used to identify participants. Table 3 also provides references pertaining to participant systems. Relatively current surveys of techniques for recognizing typeset and handwritten mathematical expressions are available [6,9,10,21,22] along with a more general introduction [23].

4.1 Language models

Here we briefly summarize the symbolic, spatial and/or temporal constraints used by CROHME systems. These are used to validate hypotheses and/or constrain search in participant systems.

4.1.1 Symbols

For symbol classes, some participants represent all CROHME symbol classes separately, while others (e.g., *Wat*) use a

Table 3 CROHME participants (2011–2014)

Id	Research group	References	CROHME 2011	2012	2013	2014
Ath	Athena Research Center (Greece)	[24]	✓	✓		✓
Czt	Czech Technical University (Czech Republic)	[25,26]			✓	
Mys	MyScript/Vision Objects (France)			☑	☑	☑
Nan	University of Nantes, IRCCyN (France)	[13,14,27,28]	✓	✓	✓	✓
Ria	Rochester Institute of Technology (USA)	[29–35]	✓	✓	✓	✓
Rib	Rochester Institute of Technology Imaging Science (USA)	[31,35,36]				✓
Sab	Sabanci University (Turkey)	[24,37]	✓	✓	✓	
Sap	University of São Paulo (Brazil)	[16,38]			✓	✓
Tok	Tokyo University of Agriculture and Techn. (Japan)	[39–41]			✓	✓
Val	Universitat de Politècnica de Valencia (Spain)	[42–44]	☑*	✓	☑*	☑*
Wat	University of Waterloo (Canada)	[45,46]		✓		

Related work cited by groups is provided in the *Refs.* column. The rightmost portion of the table shows competitions each group participated in

☑ Winning system

☑* Winning system trained only on CROHME training data

smaller set of classes along with rules to tokenize (i.e., combine) symbols into larger symbols, e.g., representing $\leq$ using routines to combine a horizontal line below a $<$ into $\leq$, or individual letters into a function name (e.g., for 'cos,' which is a symbol class in CROHME). *Nan* was the first to include a 'reject' class to represent invalid segmentations. To make computation feasible, nearly all systems make a restriction upon the maximum number of strokes in a symbol (most commonly 4 strokes).

4.1.2 Spatial relationships between symbols

Many systems estimate the typographic class of symbols, representing where a symbol sits on the writing line. Commonly these include ascenders (e.g., '*d*'), descenders (e.g., '*y*') and symbols lying between the writing line and center line or '*x*-line' (e.g., '*a*'). These typographic classes are then used to constrain the locations of symbols in particular relationships (e.g., subscript, vs. adjacent-at-right versus superscript). Mathematical types are also used to constrain relationships (e.g., rejecting subscripted digits (2_1) and superscripts or subscripts for '+'). Another important element is how spatial regions are defined. Many methods partition space around a symbol using rectangular regions and then test for relationships of neighboring symbols or sub-expressions that lie within these regions. Alternatively, neighboring symbols or strokes are used, often with a maximum distance for valid neighbor relationships.

4.1.3 Expression grammars

Two generalizations of context-free grammars have been used to define legal expressions by participants.[5] The first generalization is two-dimensional context-free grammars that allow horizontal, vertical and scripted concatenation in production rules *(Nan, Tok, Val, Wat)*. In all cases, the grammars incorporate fuzzy *(Wat)* or probabilistic weights; some incorporate these in the grammar production rules, but more commonly the score for an interpretation is defined using symbol and relationship classification scores (e.g., in a linear combination) to avoid biasing scores toward smaller trees. The second generalization is probabilistic graph grammars, where production rules may have graphs on the left- and right-hand side production rules to represent more complex patterns *(Sab)*. These systems enumerate a space of possible layout trees and then return the highest ranked interpretation as output. To make this tractable, low-confidence symbol and relationship types are pruned during parsing (e.g., considering only top-10 symbol classes).

Three systems *(Ria, Rib, Sap)* use simple 'baseline' grammars representing only the horizontal adjacency of symbols on a writing line and rectangular spatial regions around baseline symbols [48]. Parsing with these grammars involves a greedy top-down search, locating the main baseline of the expression and then recursively locating the main baseline in each subregion. These grammars do not consider mathemat-

[5] The first example of such a grammar for math recognition was presented by Chou in the late 1980s [47].

ical types and do not insure that operators have all of their arguments.

4.2 Preprocessing stroke data

The stroke data provided in CROHME came from a variety of countries and from a number of devices including electronic whiteboards, tablet computers, writing tablets, and Anoto pens that capture physical pen strokes.[6] Stroke sampling rates and coordinate systems differ greatly between devices and capture platforms. Stroke points are represented using integers or real values, and sometimes include negative coordinates. Participants have used a variety of approaches to handle this, including normalizing expression size (e.g., fixing the expression height at 1.0 and then maintaining the relative width of the original stroke data), and resampling strokes to compensate for differences in stroke resolution arising from sampling and writing speed (faster movement yields fewer samples). Stroke resampling techniques have included line density projection interpolation *(Tok)*, uniform distance stroke resampling *(Ria, Rib, Nan)* and cubic splines *(Ria in 2013 and 2014)*.

4.3 Training

Participants have used a variety of approaches to setting system parameters for preprocessing, language models and recognition. Systems using stochastic context-free grammars have had their rule probabilities tuned both manually and algorithmically (using the Inside-Outside algorithm), and a fuzzy-based grammar system had membership functions that also required tuning *(Wat)*. A number of systems trained their recognition modules stage-wise (e.g., first symbol segmentation and/or classification, spatial relationship classification, and then parsing/search parameters). For example, each module of the *Val* system is trained independently [43]: the isolated symbol recognizer (HMM or BLSTM in [44]) using extracted isolated symbols and spatial relation classifier (SVM) from pairs extracted from the training expressions. A pair of research groups (as *Nan*) utilized an explicit reject class in their classifiers, and trained classifiers directly from expression data, dynamically generating instances for the 'reject' class within a global training architecture. Some participating systems were trained on additional data alongside the provided CROHME training set *(Czt, Tok 2013, Sap 2013)*, or on a completely different training set *(Mys)*.

4.4 Recognition operations

The recognition operations can be split into four primary sub-tasks: (1) segmenting symbols, (2) classification of symbols, (3) classification of spatial relationships and (4) parsing expression symbols and structure (i.e., the search strategy or processing pipeline). Even if some systems try to perform two or more steps simultaneously in 'holistic' approaches, they all extract information for these sub-tasks.

4.4.1 Symbol segmentation

Segmentation involves grouping strokes into symbols. To reduce the number of segmentations considered, participants always use some form of continuity constraints. Symbols drawn with multiple strokes have a temporal and/or a spatial continuity: the strokes are close in time or in space. Once the space of segmentation hypotheses is defined, systems can enumerate possible segmentations before evaluating each one. Some participants design a dedicated recognizer for symbol segmentation. However, most participants use symbol recognition results to guide segmentation: If the segment corresponds to a symbol with high confidence, then the hypothesis is kept. The reject option is between these two ideas: modeling invalid segments using the reject class and then allowing this to be considered alongside concrete symbol classes (i.e., not filtering segmentations detected as invalid from the hypothesis space).

After using symbol recognition results to evaluate segmentations, some participants select a segmentation of strokes into symbols, but most delay this decision until later processing. Depending upon the parsing method, there are two ways to delay segmentation: first, in a lattice which stores candidate segmentations, or secondly allowing the parsing process to enumerate possible segmentations. If parsing is bottom-up (e.g., CYK), these two methods for considering candidate segmentations are equivalent, as all valid segments needed to be computed, evaluated and stored.

4.4.2 Symbol classification

A wide variety of features and classification algorithms were used for symbol classification. Even if decisions can be delayed (e.g., to the parsing stage), all participants consider symbol recognition as an isolated task—i.e., the features, classification algorithms and training are completed separately from other recognition steps.

We can distinguish two families of feature types: online features which use the sequence of points in strokes, and off-line features which use the image of a symbol candidate. The advantage of online features is that they use information unavailable in an image such as timing data, exact pen position and paths in crossing or overwriting strokes. However, this additional information leads to more variability for some classes due to differing stroke orders, variations in writing speed, etc. Using both online and off-line features can be useful, and participants using this combination obtain

[6] http://www.anoto.com/.

the best results in isolated symbol recognition results. This fusion of online and off-line features can be done early at the feature level (e.g., by concatenation of feature vectors) or later using a classifier combination.

The classification algorithms employed are diverse. We can distinguish classifiers that use features sensitive to time (Nearest Neighbor with template matching or DTW, Hidden Markov Models, Recurrent Neural Network) and those considering symbol shape using off-line features (MLP, SVM, Decision trees). Some classifiers include a reject class, allowing invalid symbols to be identified.

4.4.3 Spatial relationship classification

Spatial relationships can be detected using different elements, either through symbol to symbol relationships, or sub-expression to sub-expression relationships. Early solutions use intuitive features based on relative positions of bounding boxes for symbols and/or sub-expressions, or the position of a symbol relative to a symbol on a detected baseline. These raw features have the drawback of not taking into account the shape of symbols. A single bounding box arrangement can be shared by different symbol relationships (e.g., consider the bounding boxes for a handwritten A^2 vs. pa). One way to mitigate this problem is to use the typographic class of the symbols within a relationship, e.g., descenders, ascenders and large operators (e.g., $\int$).

In the final year of the competition, layout features have been defined using shape contexts *(Val, Ria, Rib)* which provide visual information absent in the lower-resolution bounding box-based features. From a given layout feature, it is possible to recognize the spatial relations with a classifier (e.g., SVM or set of thresholds) or define weights/costs that can be used in evaluating interpretations during parsing. Similar to simple classification hypotheses, geometric or probabilistic weights can be used to generate a space of possible structural interpretations, allowing multiple interpretations to be considered during parsing. The *Wat* system is unique in that the 'Above' relationship is not considered, reducing spatial relationship classification from a 6-class to a 5-class problem.

4.4.4 Parsing

Most participants define the formula recognition problem as a global optimization, requiring the minimization of a cost or maximization of a joint probability for symbols and relationships. Formula rank scores are defined by a combination of symbol, relationship and production rule confidences (e.g., using a linear combination or geometric mean). It is natural to compute these scores and use them in the parsing process.

Participants using a context-free grammar-based language model employ the Cocke–Yonger–Kasami (CYK) parsing algorithm. We can split these participants into two groups, based on how they initialize the CYK table. *Nan, MyS, Czt* generate symbol hypotheses and use them as the leaves (terminals) of the parsing table. The second group (*Tok, Val*) use strokes as terminals, but they have to fill multiple lines of the table during initialization—the first line for one stroke symbol candidates, the second line for two stroke symbol candidates, up to the maximum number of strokes allowed in a symbol. The resulting search space between these initialization strategies can be similar, particularly when the temporal stroke order is preserved. The main difference comes from how symbol hypotheses are generated (see the discussion above). Probabilistic graph grammars (*Sab*) have also been used, using a parsing algorithm similar to CYK: initialized by symbol hypotheses, merging symbols into sub-expressions by applying grammar rules, and stopping when all strokes are consumed. One advantage of this approach is that grammar rules are not restricted to Chomsky Normal form and defined using graphs, making the grammars more intuitive to define and read.

These approaches are bottom-up algorithms. Conversely, a successful top-down parsing strategy has been proposed by *Wat*. They define a Relational CFG parser using Unger's algorithm. Each production rule explicitly defines how sub-expressions are arranged, whether horizontally or vertically. In this approach, the 'Above' relationship is not directly detected. Indeed vertical structures are parsed from top to bottom with 'Below' relations. The complexity of the stroke partitioning in this algorithm (polynomial) is reduced by terminal re-ordering before rule applications.

Remaining approaches use recursive baseline detection from a fixed symbol segmentation (*Ath, Ria, Sap*). Once symbols on a baseline have been detected, symbols located in vertical relationships with baseline symbols are detected. We can distinguish systems using minimum spanning trees (MST) identifying vertical structures before right, superscript or subscript relationships (*Ria 2013–2014, Rib*). While relatively fast, these baseline-based approaches are less accurate than the CFG-based approaches. They employ much less strict language models that permit invalid interpretations and are greedy algorithms with limited backtracking.

5 Results and new challenges

As the detailed results using the official metrics described in Sect. 3.2 are available in the competition papers, we choose to present the results following a different point of view in order to address the following questions: (i) how difficult is the recognition task? (ii) how do the participating systems behave toward this complexity? (iii) is it difficult to do better than what the participating systems produced? and (iv) what issues remain unsolved?

Table 4 CROHME 2014 expression rates (a) before and (b) after merging system results. In (a), the second row shows the number of expressions recognized only by the corresponding system. In (b), the number of correctly recognized expressions obtained after merging all stroke labeling decisions from 2 to 7 systems is shown, both in order of increasing and decreasing expression recognition rate as shown at the left

a. CROHME 2014 test set expression rates

	Mys	Nan	Ria	Rib	Sap	Tok	Val	Any Sys.
Correct	618	257	187	187	148	253	367	707
Unique	164	5	3	3	0	8	23	–

b. Expression rates after label merging

#Syst. (%)	Increasing	Decreasing
2	307 (31.14)	716 (72.62 %)
3	422 (42.80 %)	751 (76.17 %)
4	563 (57.10 %)	774 (78.50 %)
5	668 (67.75 %)	786 (79.72 %)
6	727 (73.73 %)	800 (81.14 %)
7	810 (82.15 %)	

In this section, we focus on the last competition results for analyzing which type of expressions still need attention. After this analysis, we show how to use the proposed tools and framework based on the label graph as described in Sect. 3.2 to merge results from different systems and do a high-level error analysis.

5.1 Expression-level results with regard to expression complexity

Table 4a counts the number of correct expressions from the last test set (Part 4 test set 2014), which are well recognized by each system. For example, out of 986 expressions, the best performing system recognized 618 expressions (62.15 %) and the system ranked last recognized 148 (15.05 %). While checking expression-wise performance of the systems, we note that 707 expressions are recognized by at least one system and 279 expressions are never recognized. It indicates that there are 89 expressions ($707-618=89$), which are not recognized by the best system but are recognized by some of the other systems. The second row shows the number of expressions, which are recognized only by the corresponding system (and not by any of the other systems). It shows that each system has a different behavior, and they all have their own strengths and weaknesses. Furthermore, if an oracle is available to choose the right system for each expression, theoretically it would be possible to reach at a recognition rate of $707/986 = 71.7\,\%$. The merging of the decisions coming from different systems is discussed in the next section.

Let us now focus on the remaining 279 expressions which are never recognized. Figure 4 shows the repartition of these expressions with regard to the complexity criteria defined in Sect. 2.3. The histograms show that the CROHME data still have challenging expressions to be recognized. If we consider the number of symbols per expression as a complexity measure, more than 41 % expressions having more than 11 symbols are never recognized. The expressions with less symbols are nicely recognized, and only 13 % of the expressions with less than 5 symbols are never recognized. The misrecognition rate for expressions with 6–10 symbols is 22 %. If we consider more structural criteria, the proportion of never recognized expressions always increases with the complexity. It rises from 13 % of expressions with only one baseline to 19 % with 2 baselines, 33 % with 3 baselines and then between 32 and 57 % for more complex expressions. This is more evident if we take into account only the number of distinct baselines: The proportion of never recognized expressions triples from 13 to 39 % when expressions have 3 baselines like a subscript and a superscript in the same time or just a simple fraction. The last histogram (maximum depth) well sums up the situation: one-third of the test set (~300 exp.) is simple and quite well recognized (only 13 % are never recognized exp.); one-half (~500 exp.) has a first level of difficulties and are still difficult to recognize (31% are never recognized) and more complex expressions are still challenging.

5.2 Merging system outputs

In the previous section, we showed that if it is possible, thanks to an oracle, to choose the right recognition result among the answers from the 7 participants, we could reach the recognition rate of 71.7 % (see Table 4, 707 among the 986 expressions from the test set 2014). This is the best we can do with late decision merging (expression level). The label graphs allow stroke-level and stroke pair-level evalua-

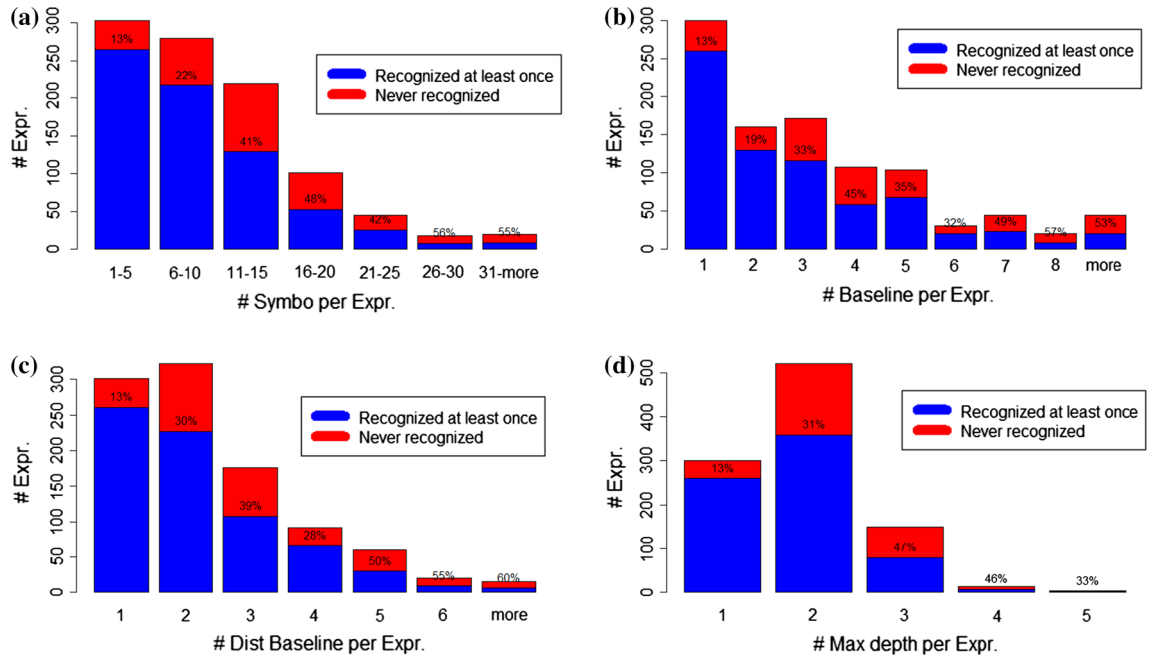


Fig. 4 Frequency of expressions with differing complexities. Complexity is characterized by **a** number of symbols, **b** number of baselines, **c** number of distinct baselines, and **d** maximum layout tree depth. Frequencies are shown for the full 2014 test set (*red* + *blue bars*), and for expressions recognized correctly by none of the participants (*red*). Percentages on *red bars* give the ratio of unrecognized expressions to all expressions belonging to the column/parameter (colour figure online)

tion of the different systems (see Sect. 3.2). We can use these label graphs to perform an early merging of the system decision. Indeed, as we did at the expression level, we can merge all local decisions using an oracle which knows the right answer. To implement this theoretical merging, we use the multi-label version of the label graph with a specific metric to compare the resulting graph with the ground truth. Let us describe the main outline of this process. First a multi-label graph is built for each expression by aggregating the output files from each participant. Then these multi-label graphs are compared with the ground truth: For a solution (stroke and edge labels) if the correct label is present among all decisions, we keep only this solution else we replace it by an error label. Finally, these merged and corrected label graphs are compared again by using the standard evaluation tool to extract recognition rate and error counters.

The right part of Table 4 shows the expression recognition rate for the best possible early merging of each local decision (strokes and relations) for 2 or more participants. As the contribution of each participant is not the same, we progressively add the systems sorted by their global recognition rate: increasing (the two weakest systems are merged first) or decreasing (the two best systems are merged first).

The first observation is that each system contributes to the merging; indeed, the number of correct expressions always increases whatever the order of the systems. Table 4, in addition, shows that one system has no expressions which are recognized only by it, but at the stroke and relation level it participates significantly (10 expressions added in the last step of the decreasing order). The second point is that this merging strategy can reach a better recognition rate of 82.15 % which is much more than the 71.7 % by considering only the expression level results. It means that among the 279 expressions which are never recognized, 103 expressions can be recognized using an early merging of the system results. Some expressions are partially recognized by several systems, and the correct expression can be retrieved from these partial results. Note that no language model has been used in this merging process. It means that if some errors introduce completely wrong structures, even by correcting the stroke labels, it will be difficult to retrieve the correct structure. Nevertheless, once correct and incorrect nodes and edges are detected, we can measure the primitive and object-level metrics. The globally merged system incurs 214 errors on stroke labels and 600 on edge labels which are only a quarter of the errors shown by the best system results (845 errors on strokes and 2489 edges, cf. [5]). If we consider the object-level metrics, the symbol recall rate is 98.32 % (well segmented and well recognized) and relation recall rate is 97.13 % (well detected with valid objects and relationships). Compared to the system *Mys*, symbol recognition has been increased more than spatial relation recognition (+4.41 vs. +2.87 %). These corrections allow recovering half of their miss-recognized expressions (−47 %, from 368 to 176 wrong expressions).

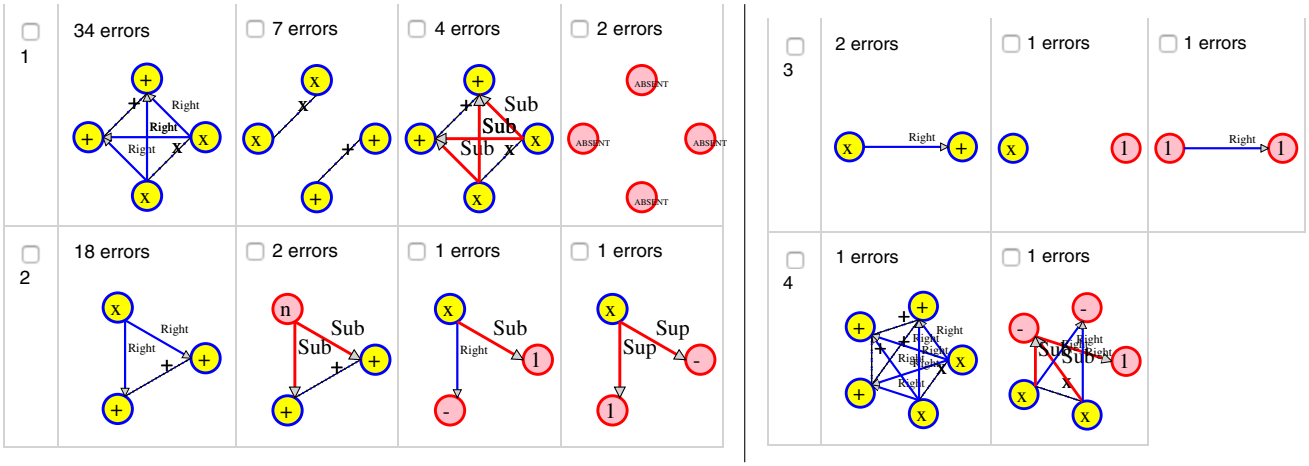


Fig. 5 Structure confusion histogram for Target '$x+$.' These results are represented in an HTML file, computed using the *Nan* outputs for the 2014 test set. The first column shows the different stroke-level ground-truth graphs and total number of misrecognitions, numbered: *1* (4 strokes), *2* (3 strokes), *3* (2 strokes), and *4* (5 strokes). The remaining columns in each row show specific errors and their counts for each ground-truth stroke pattern. The 'tick' (selection) boxes allow individual files in which errors occur to be selected and exported. A number of confusions have been omitted for space

5.3 Bigram error analyses

As described in Sect. 3.2, the CROHME metrics are based on counting the errors at stroke level (stroke labels or relation label between strokes) or object level. However, it also possible to count the errors at subgraph level without a big increase of complexity. In this section, we show what can be done at the bigram level where two symbols are linked by a spatial relation.

Starting from the ground-truth symbol layout tree, we enumerate each pair of symbols which are connected in this tree. Using corresponding strokes, it is possible to check whether this subgraph is correct or not in the recognized expression. One drawback of this approach is that sometimes one primitive level error can lead to several wrong bigrams because the corresponding symbol appears in several subgraphs. Nevertheless, this counting allows a high-level error analysis. Figure 5 presents a small capture of the HTML file generated by this error analysis tool applied to *Nan* system. For this system, a total of 4056 errors are counted for 1489 different symbol bigrams. Figure 5 shows the errors for only one particular bigram pattern $x+$. For a particular symbol bigram, many different conditions at stroke level exist in the test set. For instance, the bigram '$x+$' appears with 4 different conditions in the test set (from 2 strokes to 5 strokes). In Fig. 5, the wrong labels are drawn in red. With the 4-stroke case (row 1), out of the 34 errors which appear, 7 errors concern a missing spatial relation between x and $+$, 4 errors concern a wrong spatial relation (*Sub* instead of *Right*, x_+), missing nodes (*ABSENT* label) happen twice; in the 3-stroke cases (row 2), 2 errors concern wrong symbol and spatial relation label (n_+) and 1 error accumulates a wrong segmentation and wrong labels (x_1-), ...

Table 5 presents the most frequent bigram errors for each system and for all systems together. For each system, the three most frequent errors are selected (in bold font) and can be compared with errors from other systems. Figure 6 shows one example which explains one occurrence of the error on the bigram $+1$ for the system *Nan*. Furthermore, this figure shows one possible visualization of the errors in the corresponding label graph: Five strokes are used (numbered in nodes from 0 to 4), 3 nodes have the correct label (blue ones) and 2 are incorrect (wrong segmentation of the x); all edges are misrecognized (the correct label is in brackets, the label '_' stands for 'no edge').

It is interesting to note that the systems have different behaviors but also share some common difficulties. The two sub-expressions $x+$ and $\frac{1}{}$ are the most frequent errors, maybe because these 4 symbols are the most frequent ones (see Table 2). Furthermore, $x+$, $x\times$ and $\times x$ are very frequent, it can be explained because these bigrams combine several difficulties. They are multi-stroke, thus at first, there is difficulty in segmentation phase. These symbols can easily be confused keeping a valid structural sub-expression (if xx is recognized instead of $x\times$ it could still be a valid expression). The $-$ $=$ and $=$ $-$ subexpressions are difficult to recognize because the horizontal lines can be merged to build an extended fraction bar. For example, $=-$ can be recognized as $\frac{-}{}$. Note that the horizontal bar can be a fraction bar or a minus sign. It is surprising that the most of frequent symbol bigram errors have left–right relation. The only complex structure is $\frac{1}{}$, and several errors are possible. For instance, merging of the symbol in a $+$ sign or in a digit 1 written with a horizontal stroke.

If we look at the merged system results, we can say that the most difficult bigrams to recognize are $\sum^n$ and $\sum_i$ (mainly

Table 5 Symbol bigram error counts (CROHME 2014). Errors are shown for each system, globally for all systems (All) and using the merged label output from all systems. The top-3 errors in each column are shown in bold

Bigram	Number of errors								
	All	Merged	Mys	Nan	Ria	Rib	Sap	Tok	Val
$\frac{1}{}$	358	1	5	27	**60**	**91**	**108**	**52**	15
$x+$	285	1	11	**55**	41	41	56	**55**	**26**
$(x$	224		5	**32**	**44**	32	52	**50**	9
$+1$	224		7	**34**	29	44	**67**	38	5
$=-$	209		6	20	34	49	**60**	22	18
$=1$	180			17	33	**52**	46	26	5
00	160			31	**53**	31	20	17	5
$-=$	157	1		17	21	**55**	39	15	7
$x\times$	145	6	**24**	22	20	24	22	10	**23**
$\times x$	141	5	**23**	20	19	24	22	10	**23**
$f($	136		4	30	21	18	21	19	**23**
$\bar{n}$	130	**8**	8	16	21	28	24	21	12
$\sum^n$	114	**12**	**17**	17	17	12	17	17	17
$\sum_i$	63	**8**	9	9	9	12	11	9	9

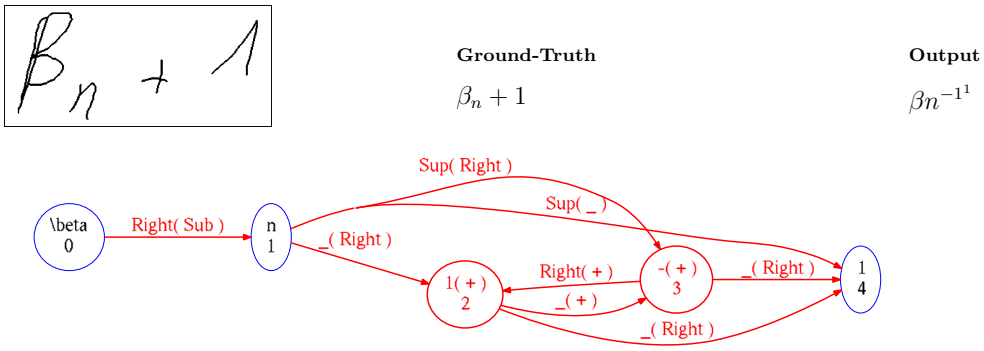


Fig. 6 Misrecognized expression from *Nan* System and corresponding label graph error Visualization. The five input strokes are numbered from *0* to *4*, appearing at the bottom of each node. *Red* nodes and edges represent incorrect labels, with the correct label shown in *brackets*. This sample illustrates one error for the '+1' pattern: *the left–right* link between the '+' and the '1' is wrong (*Sup* instead of *Right*) and '+' is over-segmented into '−1.' *Note* on edges, '_' represents no relationship (colour figure online)

confusion between relations *Above* with *Sup* and *Below* with *Sub*). Among the presented cases, the three most frequent errors use spatial relations with complex symbols ($\sum$, fraction bar) and the other types use very confusing symbols $\times$ and x. Actually 303 different bigrams are listed in the full error list of the merged system and 705 are listed in the error list for the best participating system, i.e, *Mys*.

6 Open problems and conclusion

The analysis of CROHME results shows two important aspects. Firstly, the community of CROHME has proposed efficient solutions to solve the problem for simple cases: Small expressions with simple structures have only about 13 % error. The second point is that complex expressions are still challenging. However, complex expressions do not heavily degrade the overall performance of the systems as a system can easily achieve accuracy like 30 % just by recognizing simple expressions present in the dataset. The error analysis shows that symbol segmentation and recognition are still difficult in the context of complex expressions. The problem is not in the symbol classifier (as shown in the isolated symbol classification task in CROHME 2014) but in taking into account a more global or local context. Furthermore, we show that having efficient tools to analyze complex elements like math expressions is important. They can highlight specific problems which can be hidden in the mass of information. The merging experiment shows that the different systems have different strengths, which are often comple-

mentary. Thus, it is important that the community continue to share the tools, datasets and their systems in order to capitalize upon the efforts done by each one. Finally, it could be concluded that the organization of CROHME is indeed a huge effort which lead to setting up a full framework for handwritten math recognition, and it has a tremendous contribution for advancement of the related research field.

To conclude, exciting perspectives for this field of research can be suggested. One of the most promising one would be to incorporate in a deeper way statistical language models. Using basic n-gram, or more elaborated skip-gram, models defined either at the symbol or at the sub-expression level seem appealing. Such models have proved to be very effective for improving text recognition performance. Of course, the adaptation is not straightforward because of the nature of the 2D structures, which are not present in regular texts. Extending this concept, it may be tempting to get rid of the use of formal CFG to guide the interpretation stage and to consider more semantic approaches, so that mathematical concepts could be revealed. Following this idea, it would be interesting to look in the direction of Continuous Bag of Words for computing vector representations of sub-expressions.

Acknowledgments We thank Drs. Kim and Kim (KAIST, South Korea), for their help during the first CROHME. Part of this research was supported by the National Science Foundation (USA) Grant No. IIS-1016815.

References

1. Anderson, R.H.: Syntax-directed recognition of hand-printed two-dimensional mathematics. In: Symposium on Interactive Systems for Experimental Applied Mathematics: Proceedings of the Association for Computing Machinery Inc., Symposium, pp. 436–459. ACM, New York, NY, USA (1967)
2. Mouchère, H., Viard-Gaudin, C., Kim, D.H., Kim, J.H., Utpal, G.: Crohme 2011: competition on recognition of online handwritten mathematical expressions. In: Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR). Beijing, China (2011)
3. Mouchère, H., Viard-Gaudin, C., Kim, D.H., Kim, J.H., Utpal, G.: Icfhr 2012—competition on recognition of on-line mathematical expressions (crohme 2012). In: Proceedings of the 13th International Conference on Frontiers in Handwriting Recognition (ICFHR). Bari, Italy (2012)
4. Mouchère, H., Viard-Gaudin, C., Zanibbi, R., Garain, U., Kim, D.H., Kim, J.H.: Icdar 2013 crohme: third international competition on recognition of online handwritten mathematical expressions. In: Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR). Washington, DC, USA (August 2013)
5. Mouchère, H., Viard-Gaudin, C., Zanibbi, R., Utpal, G.: Icfhr 2014—competition on recognition of on-line mathematical expressions (crohme 2014). In: Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition (ICFHR). Creta, Greece (2014)
6. Chan, K., Yeung, D.: Mathematical expression recognition: a survey. Int. J. Doc. Anal. Recognit. **3**(1), 3–15 (2000)
7. Sain, K., Dasgupta, A., Garain, U.: Emers: a tree matching-based performance evaluation of mathematical expression recognition systems. Int. J. Doc. Anal. Recognit. **14**(1), 75–85 (2011)
8. Zanibbi, R., Mouchère, H., Viard-Gaudin, C.: Evaluating structural pattern recognition for handwritten math via primitive label graphs. In: IS&T/SPIE Electronic Imaging, pp. 865817-1–865817-11. International Society for Optics and Photonics (2013)
9. Blostein, D., Grbavec, A.: Recognition of mathematical notation. In: Bunke, H., Wang, P. (eds.) Handbook of Character Recognition and Document Image Analysis, pp. 557–582. World Scientific Publishing Company (1997)
10. Zanibbi, R., Blostein, D.: Recognition and retrieval of mathematical expressions. Int. Doc. Anal. Recognit. **15**(4), 331–357 (2012)
11. Cajori, F.: A History of Mathematical Notations, vol. 2. The Open Court Publishing Company, Chicago (1929)
12. Marriott, K., Meyer, B., Wittenburg, K.B.: Visual language theory, ch. In: A Survey of Visual Language Specification and Recognition, pp. 5–85. Springer, New York (1998)
13. Awal, A.-M., Mouchère, H., Viard-Gaudin, C.: Towards handwritten mathematical expression recognition. In: Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR), pp. 1046–1050 (2009)
14. Quiniou, S., Mouchère, H., Saldarriaga, S., Viard-Gaudin, C., Morin, E., Petitrenaud, S., Medjkoune, S.: Hamex—a handwritten and audio dataset of mathematical expressions. In: Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR), pp. 452–456 (2011)
15. Stria, J., Bresler, M., Průša, D., Hlavc, V.: Mfrdb: Database of annotated on-line mathematical formulae. In: Proceedings of the 13th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 542–547 (2012)
16. Aguilar, F.D.J., Hirata, N.S.: Expressmatch: a system for creating ground-truthed datasets of online mathematical expressions. In: Proceedings of 10th IAPR International Workshop on Document Analysis Systems (DAS), pp. 155–159. IEEE (2012)
17. MacLean, S., Labahn, G., Lank, E., Marzouk, M., Tausky, D.: Grammar-based techniques for creating ground-truthed sketch corpora. Int. J. Doc. Anal. Recognit. **14**(1), 65–74 (2011)
18. Garain, U., Chaudhuri, B.B.: A corpus for OCR research on mathematical expressions. Int. J. Doc. Anal. Recognit. **7**(4), 241–259 (2005)
19. Zanibbi, R., Pillay, A., Mouchère, H., Viard-Gaudin, C., Blostein, D.: Stroke-based performance metrics for handwritten mathematical expressions. In: Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR), pp. 334–338. IEEE (2011)
20. Álvaro, F., Sánchez, J.-A., Benedí, J.-M.: An image-based measure for evaluation of mathematical expression recognition. In: Sanches, J., Micó, L., Cardoso J. (eds.) Pattern Recognition and Image Analysis, vol. 7887 of Lecture Notes in Computer Science, pp. 682–690. Springer, Berlin (2013)
21. Garain, U., Chaudhuri, B.: OCR of Printed Mathematical Expressions. Springer, New York (2007). doi:10.1007/978-1-84628-726-8_11
22. Tapia, E., Rojas, R.: A survey on recognition of on-line handwritten mathematical notation. In: Technical Report, Free University of Berlin, January (2007)
23. Blostein, D., Zanibbi, R.: Processing mathematical notation, chap. 5.6. In: Doermann, D., Tombre, K. (eds.) Handbook of Document Image Processing and Recognition. Springer, London (2014)
24. Simistira, F., Katsouros, V., Carayannis, G.: A template matching distance for recognition of on-line mathematical symbols. In: Proceedings of the 11th International Conference on Frontiers in Handwriting Recognition (ICFHR), (Montréal), pp. 415–420 (2008)

25. Stria, J., Průša, D.: Web application for recognition of mathematical formulas. In: Proc. Conf. Theory and Practice of Information Technologies, (Vrátna dolina, Slovak Republic), pp. 47–54 (2011)
26. Stria, J., Bresler, M., Průša, D., Hlavác, V.: Mfrdb: Database of annotated on-line mathematical formulae. In: Proceedings of the 13th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 542–547 (2012)
27. Awal, A.-M., Mouchère, H., Viard-Gaudin, C.: Improving online handwritten mathematical expressions recognition with contextual modeling. In: Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 427–432 (2010)
28. Awal, A.-M., Mouchère, H., Viard-Gaudin, C.: A global learning approach for an online handwritten mathematical expression recognition system. Pattern Recognit. Lett. **35**(1), 68–77 (2014). Frontiers in Handwriting Processing
29. Hu, L., Zanibbi, R.: HMM-based recognition of online handwritten mathematical symbols using segmental k-means initialization and a modified pen-up/down feature. In: Proceedings of International Conference Document Analysis and Recognition, pp. 457–462. Beijing, China (Sept. 2011)
30. Hu, L., Hart, K., Pospesel, R., Zanibbi, R.: Baseline extraction-driven parsing of handwritten mathematical expressions. In: Proceedings of International Conference Pattern Recognition, pp. 326–330. Tsukuba Science City, Japan (Nov. 2012)
31. Hu, L., Zanibbi, R.: Segmenting handwritten math symbols using AdaBoost and multi-scale shape context features. In: Proceedings of International Conference Document Analysis and Recognition, pp. 1180–1184. Washington, USA (2013)
32. Eto, Y., Suzuki, M.: Mathematical formula recognition using virtual link network. In: Proceeding of International Conference Document Analysis and Recognition, pp. 762–767. Seattle, USA (2001)
33. Davila, K., Ludi, S., Zanibbi, R.: Using off-line features and synthetic data for on-line handwritten math symbol recognition. In: Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 323–328. Crete, Greece (2014)
34. Zanibbi, R., Blostein, D., Cordy, J.R.: Recognizing mathematical expressions using tree transformation. IEEE Tran. Pattern Anal. Mach. Intel. **24**(11), 1455–1467 (2002)
35. Álvaro, F., Zanibbi, R.: A shape-based layout descriptor for classifying spatial relationships in handwritten math. In: ACM Symposium Document Engineering, pp. 123–126. Florence, Italy (2013)
36. Liwicki, M., Bunke, H.: Feature selection for HMM and BLSTM based handwriting recognition of whiteboard notes. Int. J. Pattern Recognit. Artif. Intell. **23**(5), 907–923 (2009)
37. Celik, M., Yanikoglu, B.: Mathematical formula recognition using a 2D stochastic graph grammar. In: Proceedings of International Conference Document Analysis and Recognition, pp. 161–166. Beijing, China (2011)
38. Julca-Aguilar, F., Hirata, N., Viard-Gaudin, C., Mouchère, H., Medjkoune, S.: Mathematical symbol hypothesis recognition with rejection option. In: Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 500–504. Crete, Greece (2014)
39. Le, D., Phan, T.V., Nakagawa, M.: A system for recognizing online handwritten mathematical expressions and improvement of structural analysis. In: Proceedings of 11th IAPR International Workshop on Document Analysis Systems (DAS). Tours, France (2014)
40. Zhu, B., Gao, J., Nakagawa, M.: Objection function design for MCE-based combination of on-line and off-line character recognizers for on-line handwritten Japanese text recognition. In: Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR), pp. 594–599. Beijing, China (2011)
41. Lee, A., Nakagawa, M.: A tool for ground-truthing online handwritten mathematical expressions. In: International Graphonomics Society Conference. Nara, Japan (2013)
42. Álvaro, F., Sánchez, J., Benedí, J.: Recognition of printed mathematical expression using two-dimensional stochastic context-free grammars. In: Proceedings of International Conference Document Analysis and Recognition, pp. 1225–1229. Beijing, China (2011)
43. Álvaro, F., Sánchez, J., Benedí, J.: Recognition of online handwritten mathematical expressions using 2D stochastic context-free grammars and Hidden Markov Models. Pattern Recognit. Lett. **35**, 56–67 (2014)
44. Álvaro, F., Sánchez, J., Benedí, J.: Offline features for classifying handwritten math symbols with recurrent neural networks. In: Proceedings of International Conference Pattern Recognition, p. (to appear) (2014)
45. Labahn, G., Lank, E., MacLean, S., Marzouk, M. S., Tausky, D.: Mathbrush: a system for doing math on pen-based devices. In: Proceedings of Document Analysis Systems, pp. 599–606. Nara, Japan (2008)
46. MacLean, S., Labahn, G.: A new approach for recognizing handwritten mathematics using relational grammars and fuzzy sets. Int. J. Doc. Anal. Recognit. **16**(2), 139–163 (2013)
47. Chou, P.A.C.: Recognition of equations using a two-dimensional stochastic context-free grammar. In: Pearlman, W.A. (ed.) Visual Communications and Image Processing IV, vol. 1199 of SPIE Proceedings Series, pp. 852–863 (1989)
48. Zanibbi, R., Blostein, D., Cordy, J.R.: Recognizing mathematical expressions using tree transformation. Pattern Analysis and Machine Intelligence, IEEE Transactions on **24**, 1455–1467 (2002)

Pattern Recognition Letters 35 (2014) 46–57

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

An annotation assistance system using an unsupervised codebook composed of handwritten graphical multi-stroke symbols

Jinpeng Li *, Harold Mouchère, Christian Viard-Gaudin

IRCCyN (UMR CNRS 6597), L'UNAM, Université de Nantes, France

ARTICLE INFO

Article history:
Available online 13 December 2012

Keywords:
Graphical symbol knowledge extraction
Graphical symbol retrieval
Spatial relations
Minimum Description Length principle
On-line handwriting

ABSTRACT

Many present recognition systems take advantage of ground-truthed datasets for training, evaluating and testing. But the creation of ground-truthed datasets is a tedious task. This paper proposes an iterative unsupervised handwritten graphical symbols learning framework which can be used for assisting such a labeling task. Initializing each stroke as a segment, we construct a relational graph between the segments where the nodes are the segments and the edges are the spatial relations between them. To extract the relevant patterns, a quantization of segments and spatial relations is implemented. Discovering graphical symbols becomes then the problem of finding the sub-graphs according to the Minimum Description Length (MDL) principle. The discovered graphical symbols will become the new segments for the next iteration. In each iteration, the quantization of segments yields the codebook in which the user can label graphical symbols. This original method has been first applied on a dataset of simple mathematical expressions. The results reported in this work show that only 58.2% of the strokes have to be manually labeled.

1. Introduction

Graphical symbols which are the lexical units of graphical languages are composed of a spatial layout of single or several strokes. Usually everybody share some conventions about the symbol shape. These conventions allow individuals to read graphical messages comprising similar symbols. Many existing recognition systems (Tappert et al., 1990) analogously require the definition of the character or symbol set, and rely on a training dataset which defines the ground-truth at the symbol level. A machine learning algorithm in recognition systems consequently can be trained to recognize symbols from large, realistic corpora of ground-truthed input. Such datasets are essential for the training, evaluation, and testing stages of the recognition systems. However, collecting all the ink samples and labeling them at the symbol level is a very long and tedious task. Hence, it would be very interesting to be able to assist this process, so that most of the tedious work can be done automatically, and that only a high level supervision need to be defined to conclude the labeling process.

In this regard, we propose to extract automatically a finite set of relevant patterns, called codebook within an unlabeled dataset. Searching relevant patterns and extracting them aim to reduce the redundancy in appearance of basic regular shapes and regular layout of these shapes in a large collection of handwritten scripts.

For the targeted application, which is related to an on-line handwritten corpus of mathematical numerical expressions, we consider that the basic units are the strokes, a sequence of points between a pen-down and a pen-up. Should this assumption not be verified, then an additional segmentation process will have to be undergone, so that every basic graphical unit belongs to a unique symbol. Conversely, a symbol can be made of one or several strokes, which are not necessarily drawn consecutively, i.e. we do not exclude interspersed symbols. Afterward, a symbol is made of a single stroke or several strokes within the confines of specific spatial composition. The problem is to identify symbols from a large collection of handwritten strokes in spatial layouts. Let us illustrate some simple examples to understand the problems.

Imagine a document with only two different shapes of stroke, e.g. "−" and ">". Without any context, "−" and ">" might be regarded as two different symbols "minus" and "greater than" respectively. Each stroke corresponds directly to a single symbol. If two strokes are placed together like "→" we can imagine it becomes another symbol "arrow". A stroke is only a part of symbol. Eventually, the same kind of stroke according to the context will be either a single symbol or a piece of a more complex symbol. So the first problem pointed out is searching different shapes of strokes, termed as *graphemes*.

Let us put two strokes together: it exists many composition rules named *spatial relations*. Applying two same graphemes, two

* Corresponding author.
E-mail addresses: jinpeng.li@univ-nantes.fr (J. Li), harold.mouchere@univ-nantes.fr (H. Mouchère), christian.viard-gaudin@univ-nantes.fr (C. Viard-Gaudin).

0167-8655/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2012.11.018

different symbols, "≻" and "→", can be constructed. The only difference between them is that "−" is arranged on the right side in "≻" while on the left side in "→". This left and right relation is easily defined manually.

It is possible to design new symbols made of more different graphemes and spatial relations. For instance, a new symbol "↔" is constructed using the grapheme set $\{<, -, >\}$. We can say that "−" is *between* "<" and ">". In this case, *between* implies a relationship among three strokes which is the cardinality of this spatial relation (Clementini, 2009). In this paper the cardinality of spatial relation is limited to two strokes: from a reference stroke to an argument stroke; that is a pairwise spatial relation. However, with only 3 strokes we have to consider 6 different pairs of strokes to envisage all appropriate alternatives, for example ("<","−"), ("−", "<"), ("<",">"), etc. The number of spatial relation couples will grow rapidly with the increasing number of strokes in a layout (Li et al., 2011). Searching automatically different pairwise *spatial relations* will be the second problem.

Considering a more complicated example, Fig. 1(a) shows four different symbols, "arrow", "connection", "process", and "terminator". However, the ground-truths are unknown in advance. To avoid the ambiguity that some strokes share the same grapheme, the stroke is referenced by their index $(\cdot)$. Which set of strokes (a segment) represents a symbol? Why the combination of the strokes {(1),(2),(3)} is a valid symbol (actually "arrow")? An intuitive answer is that the spatial composition is "frequent"; it exists two similar patterns in the layout, {(1),(2),(3)} and {(5),(6),(7)}, comprising same graphemes and same spatial relations respectively (which are from the previous two problems). But the equally frequent combination of less strokes {(1),(2)} does not mean a symbol. Moreover, the third arrow {(11),(12)} only contains two strokes but its shape is similar with the previous two arrows. Graphical symbols with the same ground-truth can contain different number of strokes and different graphemes. Hence, the third problem is how to search some repetitive patterns in a layout yielding to the *graphical symbols*. A segmentation will therefore be generated at the symbol level.

By grouping graphemes in segments, we obtain a small finite set of symbol hypothesis called codebook with a higher semantic level. This codebook requires less annotation operations like in Fig. 1(b): only 3 segments have to be labeled instead of 6 symbols including 13 strokes in Fig. 1(a). But all similar segments in a cluster of the codebook do not contain the same ground-truth: different symbols can be mixed in one cluster. For instance, the stroke (4) of symbol "connection" and the stroke (13) of symbol "terminator" are merged in the same cluster because of two similar shapes. The ground-truth not only depends on the similar shape but also depends on context and meaning. Annotating the segments in a codebook will be the fourth problem.

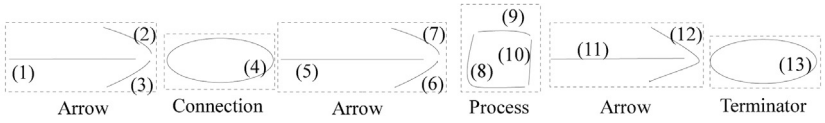
Our previous work Li et al. (2011) studies the unsupervised symbol segmentation using the MDL principle and Li et al. (2012) is specified for the spatial relation learning. This paper proposes to use the unsupervised symbol segmentation using the MDL principle to reduce symbol labeling cost. Section 2 gives a brief survey of cluster labeling in text and in off-line characters, of codebook generation using the unsupervised natural language learning built on two-dimensional spatial relations, and of the annotation on a codebook. The proposed learning framework is revealed in Section 3. In this framework, we extract the codebook composed of multi-stroke symbols which the user can label. Section 5 describes an annotation measure to evaluate the performance on the on-line handwriting corpora. At the end, the conclusion of this work is presented in Section 6.

2. State of the art

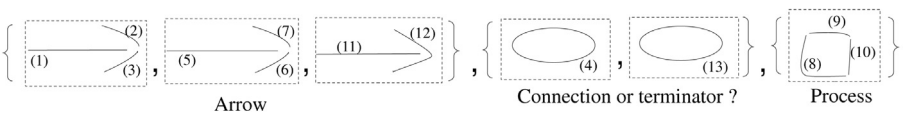
According to authors knowledge there is no existing work about unsupervised symbol extraction on on-line handwriting for annotation assistance. However, several related works will be discussed in this section: reducing annotation workload, handwriting grapheme extraction, and graphical symbol analysis.

2.1. Reducing annotation workload

Unsupervised annotation system already exists on text corpora; it partitions a large collection of text segments into clusters, and then labels each cluster automatically. Many work focus on extracting the label candidates or some keywords from the collection of text segments (Treeratpituk and Callan, 2006). However extracting the label candidates on handwritten graphical corpora in text format would be too difficult without recognition systems; our goal is to annotate the symbols from a raw handwritten dataset so that the recognition systems can be trained on them. Our work focus therefore on grouping the handwritten scripts into several clusters, and then labeling them manually. A similar offline handwriting annotation system Vajda et al. (2011) proposes the idea to label a large number of isolated characters; clustering them into several clusters of characters, and labeling the clusters in order to reduce the human effort. This work shows that over 80% symbol



(a) A handwritten flowchart contains four different graphical symbols, "arrow", "connection", "process", and "terminator" but these symbols are unknown. Searching the undiscovered symbols and grouping them into a small set can ease greatly the annotation workload. Note that all the strokes are marked by the index (.) to avoid the ambiguity



(b) A generated code book from the handwritten flowchart in Figure 1a includes three sets of segments. Some sets of segments represent only one symbol while different symbols are mixed in the others.

Fig. 1. Reducing the human effort on labeling symbols.

labeling workload have been saved. But the critical problem of character segmentation (e.g. combining three strokes as a arrow in Fig. 1(a)) has not been discussed; all the isolated characters are well segmented in advance.

2.2. Codebook extraction in handwriting

In view of the first problem mentioned in the introduction, extracting the graphemes in the field of both offline and on-line handwriting has made much progress. Many offline biometric systems (Bulacu and Schomaker, 2006; Bulacu et al., 2007; Jain and Doermann, 2011) generate the codebook using a clustering method (e.g. *k*-means, self-organizing map, etc.) so that a codebook-based probability distribution function can be employed to identify or verify the writers. It is necessary to cut the offline ink (to segment it) beforehand (Bulacu and Schomaker, 2006; Bulacu et al., 2007). The basic components are then extracted for generating the codebook. In these cases the codebook aims to be representative of a writer, but not to match to language symbols. Considering the on-line handwriting the base elements are the strokes, so we can build directly the codebook at the single-stroke level in this paper.

Concerning the codebook at multi-stroke level in (Tan et al., 2009), the handwritten graphical symbols, which are well segmented and recognized via an industrial recognition system, are clustered into different allographs using *k*-means. Varied features are used for this clustering. The two dimensional graphical symbol containing several strokes are re-sampled into fixed number of points, and embedded then in the vector space so that we can compare the similarity between two graphical symbols. However, the order of strokes has not been discussed; the embedding is stroke-order-sensitive. We need a stroke-order-free algorithm to obtain the distance between two graphical symbols composed of many strokes; writers may copy the same symbol with different stroke orders. Moreover the segments comprising different numbers of strokes would be the same symbol. For example, three arrows are alike in Fig. 1(a), but the third arrow contains two strokes while the others contain three strokes. In our work, we make use of the Hausdorff distance which is used in contour matching on offline data (images) (Huttenlocher et al., 1993). The Hausdorff distance can also avoid the problem of the stroke-order problem (two same symbols with the different orders of strokes) and of the stroke-number problem (two same symbols with the different numbers of strokes). Furthermore, although Rucklidge (1995) studies how to adapt to the affine transformation between two segments using the Hausdorff distance, it is too slow to use in our system. We prefer the classical Hausdorff distance.

A gap exists between these two levels of codebook: the single-stroke level and at multi-stroke level. While in (Tan et al., 2009) the segmentation stage can rely on the recognition tool, this is not possible in unsupervised symbol extraction. The transition, from the single-stroke level to multi-stroke level, is taken into account in our work by discovering the graphical multi-stroke symbols (the third problem in introduction), and inserting them in the codebook.

2.3. Unsupervised graphical symbol learning

To tackle this problem, most of the works are using heuristic approaches (Alexander Clark and Lappin, 2010). One of the famous approaches is the Minimum Description Length (MDL) principle (Rissanen, 1978) which assumes that the best lexicon (a set of symbols) minimizes the description length of lexicon and of observations using the extracted lexicon. Using this MDL principle, a recall rate of 90.5% for symbols is reported (Marcken, 1996) on the Brown English corpus (Francis and Kučera, 1982), which is a text dataset. We propose to extend this kind of approach on real graphical languages where not only left to right layouts have to be considered.

On an on-line graphical corpus, the units (strokes) are positioned in two-dimensional spatial relations. In this case, the search space for the combination of units which makes up possible lexical units is much more complex since it is no longer a linear one. We can describe these two-dimensional spatial relations with a graph. Thus a graph mining technique is required to extract the repetitive pattern on the graph. Such a task is performed with the SUBDUE (SUBstructure Discovery Using Examples) system (Cook and Holder, 2011). It is a graph based knowledge discovery method which extracts the substructures in a graph using the MDL principle. To create the relational graph between strokes, we have to model the spatial relations, which is the second problem stated in the introduction.

2.4. Spatial relation learning

A traditional modeling of spatial relations is represented in three levels (Clementini, 2009): the topological relations, the orientation relations, and the distance relations. The topological characteristics are preserved under transformations such as translation, rotation, and scaling. A simple example of topological relations is the intersection of two strokes. The orientation relations calculate the directional information between two strokes (Bouteruche et al., 2006). For instance the stroke *A* is on the right of the stroke *B*. The distance relations describe how far apart two strokes are.

Most of existing systems dealing with handwriting need some spatial relations between strokes. For instance, Bouteruche et al. (2006) uses a fuzzy relation position (orientation relations) for the analysis of diacritics on on-line handwritten text. In (Delaye et al., 2009), authors add a distance information to design a structural recognition system for Chinese characters. In the context of handwritten mathematical expression recognition in (Rhee and Kim, 2009), authors use the three levels of spatial relations to create a Symbol Relation Tree (SRT) using six predefined spatial relations: inside, over, under, superscript, subscript and right. However, using a simple set of predefined spatial relations is obviously not enough for describing "a new" (or "an unknown") complex graphical language. We may lose some unknown spatial relations which are important for a specified graphical language. Moreover, it is hard to predefine manually all the useful spatial relations. In our previous work (Li et al., 2012), we use a clustering technique to discover the spatial relations rather than some predefined spatial relations. Then, we use the learned spatial relations to discover the graphical symbols. The discovered graphical symbols are then added into the codebook.

2.5. Labeling codebook

Labeling the codebook will be the last problem. In reality, it is difficult to learn a codebook containing all clusters that are exact symbols. Many clusters would mix the symbols with the sub-parts of other symbols. This is over segmented. In this paper the labeling operation have to take into account the over segmented symbols.

In next section, we propose an iterative learning framework to extract the codebook composed of multi-stroke symbols. The user can then label such codebook to ease the annotation workload.

3. Proposed unsupervised multi-stroke symbol codebook learning framework

Our proposed automatic multi-stroke symbol extraction system is illustrated in Fig. 2. The on-line handwriting is imported in system, and the codebook is then exported for the annotation. Six main steps have to be taken into account in the system which is

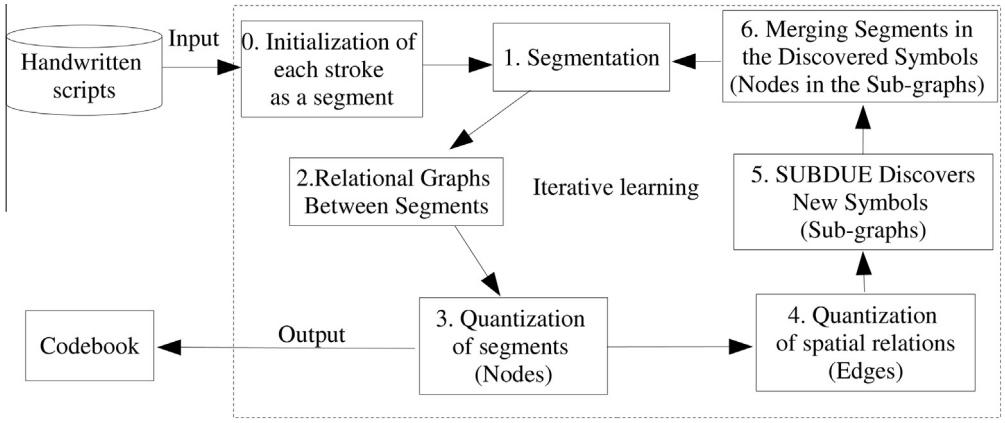


Fig. 2. Automatic multi-stroke symbol extraction system.

an iterative learning. In raw on-line data, the basic unit is the stroke. In our iterative learning framework, we consider the segment as the basic unit which may contain a multi-stroke structure. The initial segmentation is set up with each single stroke. We build firstly the relational graph between the segments with nodes for the segments and spatial relations for the edges. After the quantization of segments and spatial relations, we make use of the SUBDUE system to discover the new symbols (sub-graphs). The segments in a new symbol will be merged into a new segment for the next iteration. We start the iterative learning with building the relational graph between segments.

3.1. Relational graph construction between segments

After obtaining the segmentation, this section presents the construction of the relational graph between the segments inspired by SRT (Rhee and Kim, 2009). We define the node as the segment and the edge as spatial relation. A spatial relation is considered as a relationship from a reference segment to an argument segment. In other words, the relational graph is directed. This allows for instance to distinguish between the two following horizontal layouts of two segments "|" and "−": "−|" or "|−", which are two different symbols. Concerning the complexity, suppose we have n_{seg} different segments, to create a complete directed graph for all the vertices (segments), the number of directed edges is

$$2 \cdot C_{n_{seg}}^{2} = n_{seg}(n_{seg} - 1), \tag{1}$$

where $C_n^m = \frac{n(n-1)\ldots(n-m+1)}{m(m-1)\ldots 2\cdot 1}$ (Chartrand, 1985). In that case, the search space would be far more too complex to search patterns in the complete directed graph. Therefore, the number of out-directed edges from a reference segment should be limited to n_c closest segments where $n_c <= n_{seg} - 1$ since we, human, have a limited perceived visual angle (Baird, 1970); we prefer some symbols composed of the closest segments. The reduced number of directed edges is then:

$$n_{seg} \cdot n_c. \tag{2}$$

However, if n_c is too small, we could lose some symbols. The distance for the closest segments is defined by the Euclidean distance between two closest points in the two sets of points (the two segments) respectively. Formally, suppose that we have two segments, $seg_x = \{\ldots, pt_i, \ldots\}$ and $seg_y = \{\ldots, pt_j, \ldots\}$ where pt_i and pt_j are the points in the segments, we define the spatial distance between two segments as:

$$cdist_{seg}(seg_x, seg_y) = \min_{pt_i \in seg_x, pt_j \in seg_y} dist_{pt}(pt_i, pt_j), \tag{3}$$

where $dist_{pt}(pt_i, pt_j)$ is the Euclidean distance. Considering a reference segment seg_{ref}, we can find the closest segment, $CSeg(seg_{ref}) = \arg\min_{seg_p \in \{seg_i\}} cdist_{seg}(seg_{ref}, seg_p)$ where $CSeg(seg_{ref})$ is not necessary equal to $CSeg$ ($CSeg$ (seg_{ref})).

As an example, Fig. 3(a) illustrates the generated relational graph ($n_c = 2$) in first iteration for the flowchart in Fig. 1(a). In first iteration, each segment is composed of a single stroke. If we had created a complete directed relational graph for 13 segments, the number of edges would be using Eq. (1): $13 \times (13 - 1) = 156$. After pruning, the maximum of number of edges is $13 \times 2 = 26$ using Eq. (2) where $n_c = 2$. The number of edges is much smaller than that of the complete directed graph.

To search the patterns in the graph, we have to quantify the segments (nodes) and the spatial relations (edges). In next section, a clustering method is presented to group the segments.

3.2. Quantization of segments (nodes)

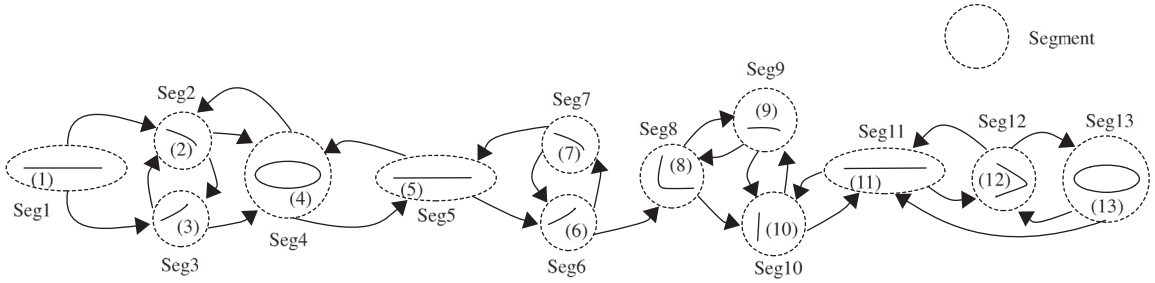
In previous section, the relational graph between segments is constructed. In this section, these segments are quantified so that we can generate the codebook in which we can annotate the clusters.

Clustering techniques are used for producing the codebook for the membership of each segment. It exists many clustering methods, hierarchical clustering (Lance and Williams, 1967), k-means (Tan et al., 2009), self-organizing map (Kohonen, 1988), neural gas (Martinetz and Schulten, 1991), etc. We prefer an agglomerative hierarchical clustering using the average metric (Lance and Williams, 1967) since it is easy to tune the number of clusters. We propose to use the Hausdorff distance to obtain the distance between two segments; it is widely used in image contour matching (Huttenlocher et al., 1993).

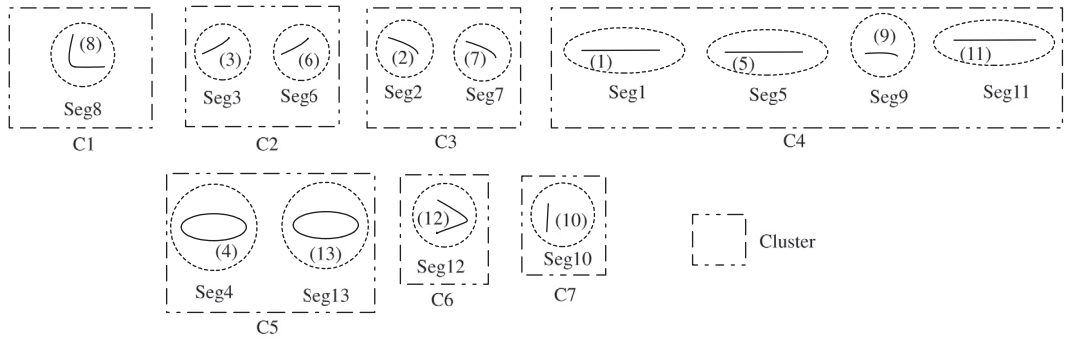
A segment could contain a single stroke or several strokes. Before the matching, a segment is normalized into a unit bounding box $\{x \in [-1, 1], y \in [-1, 1]\}$ by keeping the ratio, and all the strokes in the segment are resampled to a fixed number of 30 points respectively. The Hausdorff distance for the shape between two segments (seg_x and seg_y) is defined as

$$haufdist_{seg}(seg_x, seg_y) = \max(h(seg_x, seg_y), h(seg_y, seg_x)), \tag{4}$$

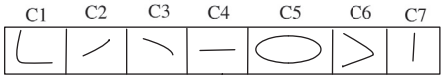
where $h(seg_p, seg_q) = \max_{pt_i \in seg_p} \min_{pt_j \in seg_q} (dist_{pt}(pt_i, pt_j))$. For hierarchical clustering, we use the Lance and Williams formula (1967) which provides an efficient computational algorithm. The membership of each segment is then generated: all the segments are grouped into n_p clusters. Once the number of n_p clusters is selected, all the segments are tagged with these virtual labels. We define the center sample seg_c who minimizes the sum of Hausdorff distances to the other samples in the cluster C:



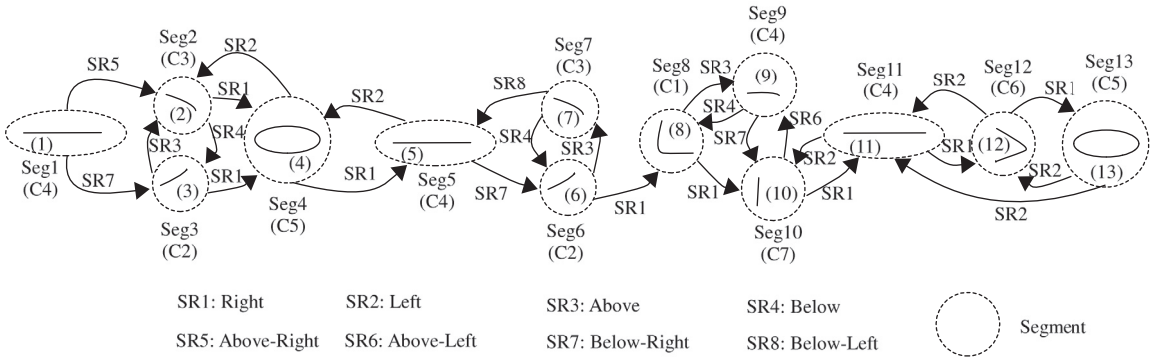
(a) The relational graph is produced in first iteration for the flowchart in Figure 1a. Each single stroke is considered as a segment in first iteration.



(b) The segments are grouped into $n_p = 7$ clusters in first iteration from the segments (nodes) in Figure 3a



(c) The codebook is visualized by choosing the center sample in each cluster. In the last iteration, user can label all these chosen samples.



(d) The relational graph from the Figure 3a after the quantization of segments into $n_p = 7$ graphemes and the quantization of spatial relations into $n_{sr} = 8$ categories.

Fig. 3. The learning procedure during the first iteration.

$$seg_c = \arg\min_{seg_p \in C} \left(\sum_{seg_q \in C} haufdist_{seg}(seg_p, seg_q) \right). \quad (5)$$

The center samples will be organized as the visualized codebook. According the iterative algorithm in Fig. 2 this quantization step can also be the last step before the output.

For instance, Fig. 3(b) displays that the segments are partitioned into $n_p = 7$ clusters in first iteration from the segments in Fig. 3(a). In reality, it exists a great number of samples in a cluster. We choose the center sample in each cluster, so that the visualized codebook in Fig. 3(c) is generated. User can therefore label these samples in the codebook at a higher level. This procedure is the quantization of segments. Afterward, we quantify the spatial relations in relational graph.

3.3. Quantization of spatial relations (edges) between segments

We have created a relational graph between the segments. In the previous example, the number of edges in the graph is 26 as

mentioned in Section 3.1 which means that it exists 26 spatial relation couples between two segments. In this section, we quantify the spatial relation couples into n_{sr} categories. We extract firstly the features of spatial relations. The spatial relation can be represented in three levels (Clementini, 2009): distance relations, orientation relations, and topological relations. In our previous work (Li et al., 2012), seven features are proposed aiming at describing these three levels. As shown in (Li et al., 2012), we did a feature selection experiment to find the best feature combination from the seven features. We found that four fuzzy directional features (right, left, above, and below) work well on the simple Calculate dataset, and outperform the others on the more challenging FC dataset. The *k-means* clustering algorithm is applied to generate n_{sr} spatial relation prototypes. All the edges in relational graph are therefore grouped into n_{sr} categories.

As an example, Fig. 3(d) shows that the nodes and edges in relational graph in Fig. 3(b) are quantified into $n_p = 7$ different shapes of segment $(C1, C2, \ldots, C7)$ and $n_{sr} = 8$ different spatial relations $(SR1, SR2, \ldots, SR8)$. To better understand the method, all the spatial relations are marked with spatial significations in Fig. 3(d). In reality, such significations do not exist.

In the next section, we will see how to extract the repetitive sub-graphs from the relational graph. The set of repetitive sub-graphs is probably composing the lexicon (i.e. the set of symbols used in the language).

3.4. Discover repetitive sub-graphs using Minimum Description Length

In previous section, we get the relational graph for a graphical language. This section presents an algorithm from Cook and Holder (1994) using Minimum Description Length (MDL) principle (Rissanen, 1978) to extract repetitive substructures in graph, which will be considered in our context as the lexical units. We expect that most of them are the symbols. In unsupervised language learning model, the MDL principle implies that the best lexical unit minimize the description length of both the lexical unit and of the graph using lexical unit (Marcken, 1996). Formally given a graph G, we try to choose the lexical unit u which minimize the description length

$$DL(G,u) = I(u) + I(G|u), \tag{6}$$

where $I(u)$ is the number of bits to encode the lexical unit u and $I(G|u)$ is the number of bits to encode the graph G using the lexical unit u. Cook and Holder (1994) gives the precise definition of $DL(G,u)$. The system SUBDUE (SUBstructure Discovery Using Examples) (Cook and Holder, 2011) extracts iteratively the best lexical unit (substructure) using the MDL principle. A unit could be a hierarchical structure (Jonyer et al., 2000; Li et al., 2011).

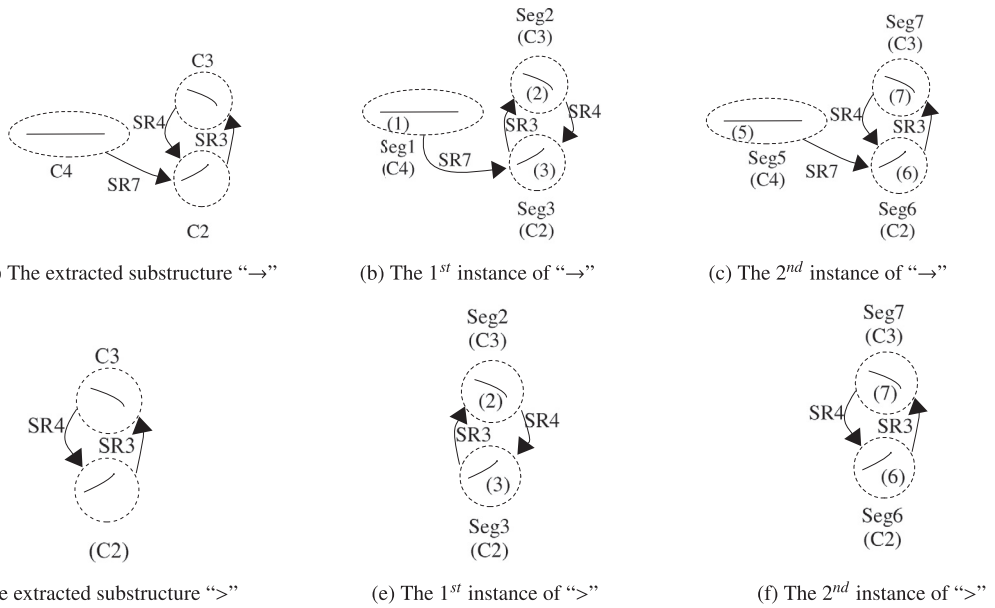
For explaining the discovery procedure, we extract a lexical unit from the same example in Fig. 3(d). The flowchart in Fig. 3(d) is the only one flowchart in the training set, but obviously in real data it exists many other varied flowcharts using the same symbol set. Fig. 4(a) and (d) illustrate two possible lexical units containing two instances respectively. Considering all the flowcharts in the training set, if the occurrence number of "$>$" and the occurrence number of "$\rightarrow$" are almost equal, the MDL principle prefers the substructure "$\rightarrow$" composed of more nodes and edges which can get a higher compression ratio in the graph. Similarly, if the occurrence number of "$>$" is much larger than that of "$\rightarrow$", we will extract the "$>$" as lexical unit according to the MDL principle.

In each iteration, we can discover $n_u \geqslant 1$ lexical units as the multi-stroke symbols using the SUBDUE system. Note that the terms, lexical unit, substructure, and symbol are equivalent in this paper. The SUBDUE system will discover a lexical unit corresponding to many symbol instances. In the next section, we will present the iterative learning by merging the segments from a discovered symbol instance into a new segment.

3.5. Iterative learning

In previous section, we have discovered one or more new symbols. However these new symbols will change the original spatial relations which were between their subparts and the rest of the relational graph. Moreover, the new symbols and old segments may be similar. It would be better to redo the learning procedure.

Let's illustrate an example to understand the problem. We suppose that the symbol "$>$" showed in Fig. 4(d) is extracted rather than "$\rightarrow$" showed in Fig. 4(a). The segments in the symbol instance will be integrated to another object. We can consider such object as a new segment. For instance in Fig. 5(a), the new segments Seg14 (Seg2 and Seg3) and Seg15 (Seg7 and Seg6) are created. We find that the shapes of the new segments Seg14 and Seg15 are similar



(a) The extracted substructure "$\rightarrow$" (b) The 1^{st} instance of "$\rightarrow$" (c) The 2^{nd} instance of "$\rightarrow$"

(d) The extracted substructure "$>$" (e) The 1^{st} instance of "$>$" (f) The 2^{nd} instance of "$>$"

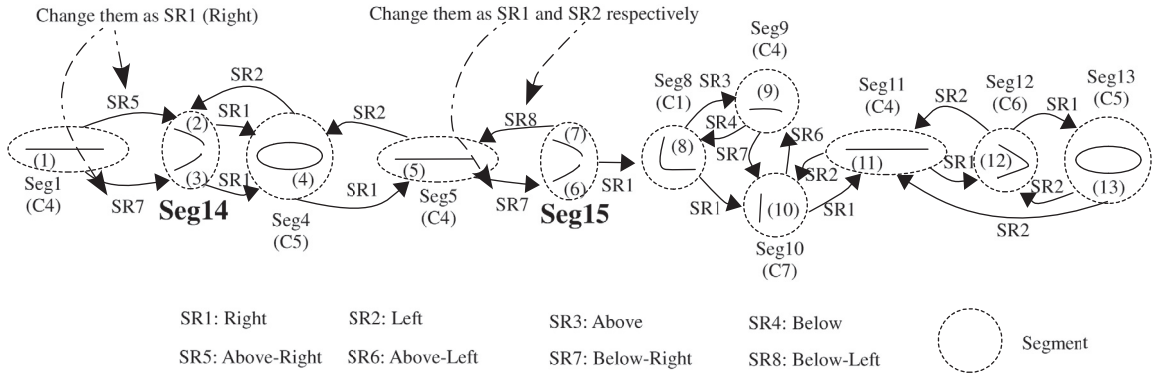
Fig. 4. Two possible symbols in first iteration in Fig. 3(d).

with that of the old segment Seg12. Thus a new codebook is needed to be calculated.

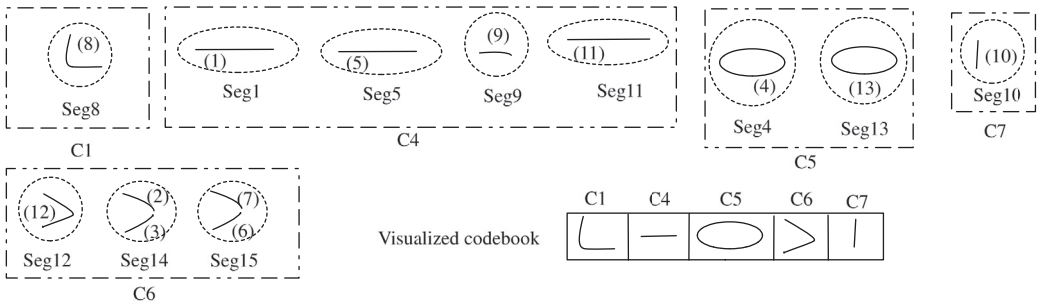
Considering the spatial relations in Fig. 5(a), Seg1 associates no longer the relation SR5(above-right) nor SR7(below-right) with Seg14 which is the combination of Seg2 and Seg3. Neither do the relation between Seg5 and Seg15. In fact, Seg14 is put on the right (SR1) side of Seg1. Therefore it would be better to recalculate the relations between the all segments.

After the merging, the second iteration begins. We group the segments into the second iteration codebook shown in Fig. 5(b). We can see that three ">", which are composed of different numbers of strokes, are in the cluster "C6". Similarly, the new relational graph in second iteration is generated in Fig. 5(c). The spatial relations (edges) are rebuilt. Fig. 5(d) illustrates extracted three arrows in the relational graph in second iteration. We combine the segments in the instances of arrow to the new segments, Seg16, Seg17, and Seg18. The codebook in third iteration is obtained in Fig. 6(a). We, user, can label the symbols in this visualized codebook.

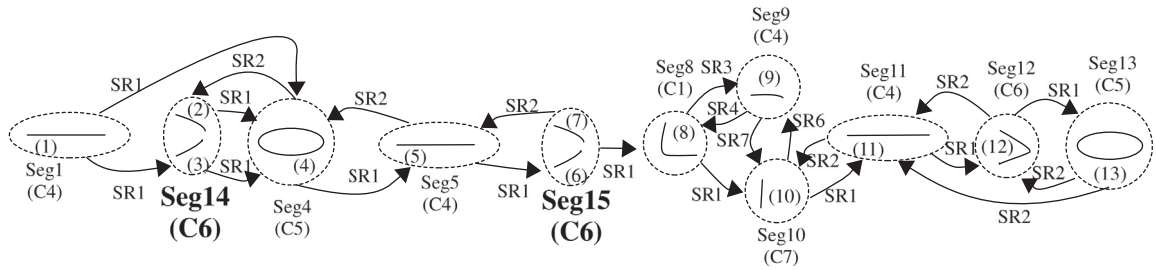
In each iteration, the quantization of segments (Section 3.1) and the quantization of spatial relation (Section 3.3 are time-consuming.



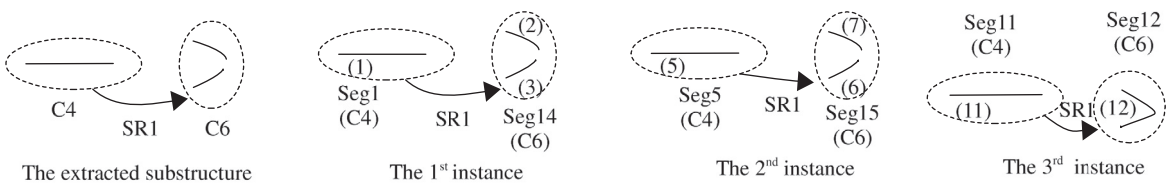
(a) The new relational graph after merging the segments in the symbol instance



(b) The codebook in the second iteration

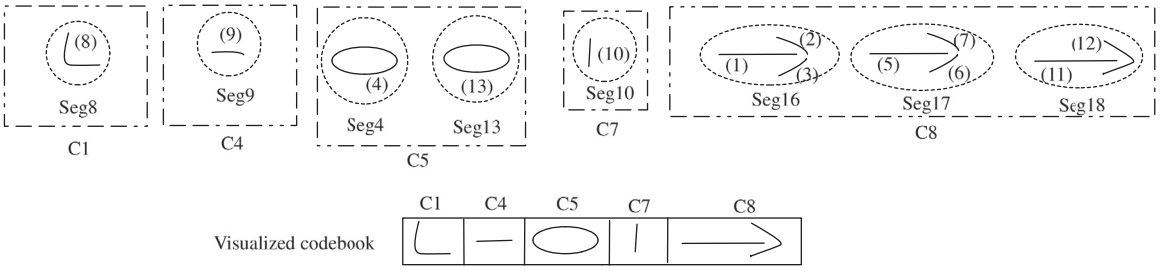


(c) The quantified relational graph in the second iteration

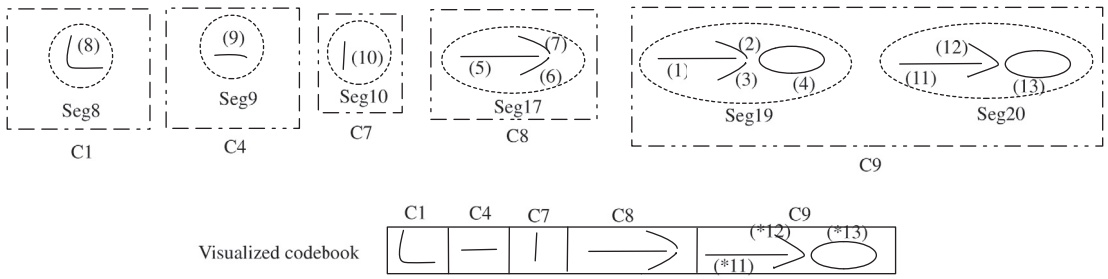


(d) The extracted substructure in the relational graph in second iteration

Fig. 5. The learning procedure in second iteration.



(a) The codebook in the third iteration



(b) The codebook in the fourth iteration

Fig. 6. The codebooks in later iterations.

Increasing the number of discovered symbols n_u in each iteration can speed up the system running. Moreover, n_u may change in each iteration and the total number of discovered lexical units may change. But we do not have a criteria that stops symbol discovery process in each iteration. In this paper, only a fixed n_u will be evaluated. The total number of discovered lexical units $n_u * (n_{it} - 1)$ will be assessed on the training part of our dataset.

4. Annotation using the codebook

In previous sections, the codebook composed of multi-stroke symbols have been obtained. We choose the center segment in the cluster to generate the visualized codebook. The user labels therefore these chosen segments stroke by stroke. In this section, we discuss how to label the segments in dataset with the small labeled codebook.

In the visualized codebook, the segments in a cluster are not always the same single symbol. Different symbols may be mixed in a cluster since the label is dependent on the context. Only the shape of segment cannot decide the label. It is an unsolved problem in this paper.

For instance, the two different labels "Connection" and "Terminator" are mixed in the cluster "C5" in the codebook illustrated in Fig. 6(a) since the label is dependent on the context. If we learn a cluster "$\rightarrow o \rightarrow$", "$o$" is easily recognized as the label "Connection".

Another frequent phenomenon is that a segment, which contains several symbols (e.g. "$\rightarrow o \rightarrow$"), is over learned. The user can separate and label them. A mapping algorithm have been developed to search for the corresponding labeled stroke from the unlabeled segment to the labeled segment. The mapping procedure is involved in normalizing the segments into a unit bounding box, and in searching then for the labeled stroke with the closest Hausdorff distance as the corresponding stroke. After the mapping process, the symbols are segmented and labeled.

As an example, using the third iteration codebook in Fig. 6(a), we continue with the learning procedure and the fourth iteration codebook is attained in Fig. 6(b). The segments in the cluster "C9", which include two symbols "$\rightarrow$" and "O", are over learned. The user label the segments in visualized codebook stroke by stroke. The labeled segment of "C9" in visualized codebook is shown on the left side in Fig. 7(a). However, the other segments in "C9" are unlabeled on right side in Fig. 7(a). A mapping procedure is required to find the corresponding labeled stroke. Considering two segments $\{(*11), (*12), (*13)\}$ and $\{(1), (2), (3), (4)\}$ with different numbers of strokes, Fig. 7(b) shows the mapping procedure which normalizes the segments and looks for the corresponding labeled stroke. The numbers of strokes between two mapping segments are not necessary equal. The mapping pairs $\{\{(1) \rightarrow (*11)\}, \{(2) \rightarrow (*12)\}, \{(3) \rightarrow (*12)\}, \{(4) \rightarrow (*13)\}\}$ are achieved. The symbol "Arrow" $\{\{(1) \rightarrow (*11)\}, \{(2) \rightarrow (*12)\}, \{(3) \rightarrow (*12)\}\}$ and the symbol "Terminator" $\{\{(4) \rightarrow (*13)\}\}$ are segmented and labeled.

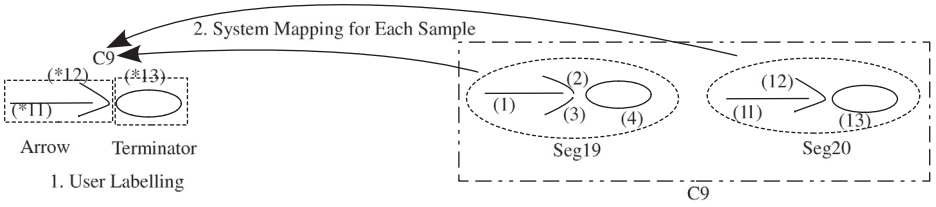
5. Experiment

In this section, we firstly present a cost function to evaluate the labeling procedure. Two on-line handwriting corpora are then described, and the labeling procedure using the different learned codebooks is tested on them.

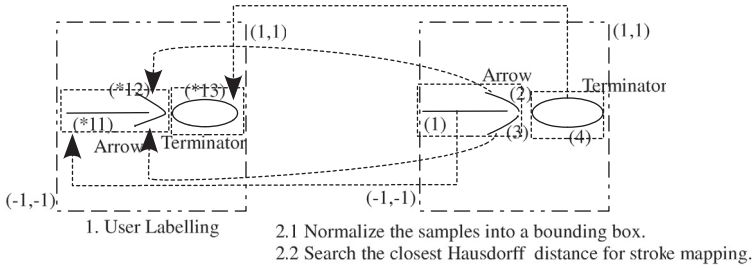
5.1. Labeling cost

In previous section, the visualized codebook is manually labeled. To evaluate the system performance, the chosen segments in visualized codebook are automatically labeled according to the available ground-truths (instead of a manual operation). We then execute the mapping procedure described in Section 4 to label all other segments. Since the user labels the segments and raw handwritten scripts in dataset stroke by stroke, we define the labeling cost C_{label} at the stroke level:

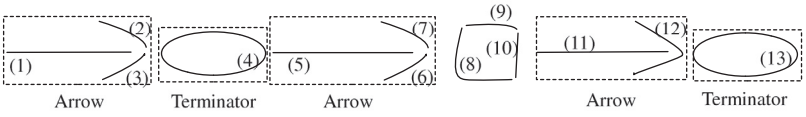
$$C_{label} = \frac{N_c + N_{db} - N_{correct}}{N_{db}}, \tag{7}$$



(a) System mapping for each segment in the same cluster



(b) Mapping unlabeled strokes in a segment



(c) The labeled handwritten flowchart derived from Figure 1a using the fourth iteration codebook in Figure 6b

Fig. 7. System labeling in the fourth iteration.

where N_c is the number of strokes in the codebook, N_{db} is the number of strokes in the dataset, and $N_{correct}$ is the number of strokes which are correctly labeled in the dataset. N_c means how many strokes have to be manually labeled in the codebook, and $N_{db} - N_{correct}$ means how many strokes in the dataset have to be corrected or labeled. N_c and N_{db} can be easily obtained by counting how many strokes in the codebook and dataset respectively. We compute $N_{correct}$ according to the number of strokes in well segmented and well labeled symbols. If $C_{label} < 1$, the system reduces the human effort of labeling. The lower labeling cost is preferable. In fact, we can consider C_{label} as the ratio of strokes in dataset which still need a manual operation.

As an example, Fig. 7(c) shows that the handwritten flowchart in Fig. 1(a) is segmented and labeled using the codebook in the fourth iteration showed in Fig. 6(b). Some segments in the visualized codebook are unknown; it is a part of symbol. We leave them as unlabeled. The number of strokes N_{db} in this handwritten flowchart is 13, the number of strokes N_c in the fourth iteration codebook is 9, and the number of strokes $N_{correct}$ in well segmented and labeled symbols is 9. The labeling cost is therefore

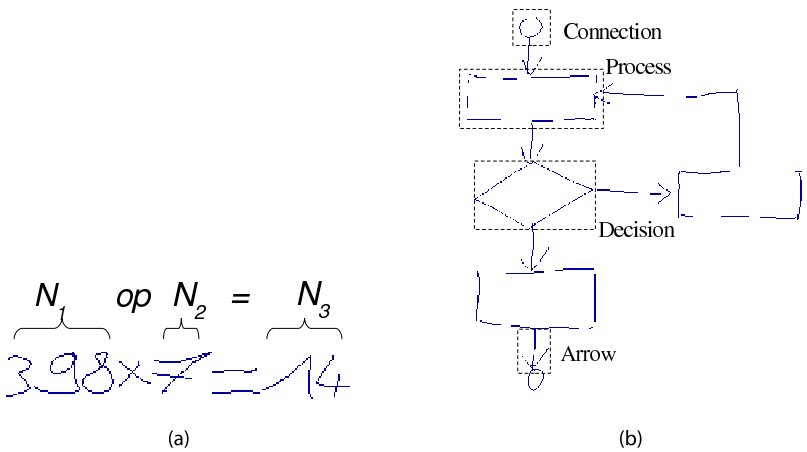


Fig. 8. Two different handwritten graphical languages: (a) a synthetic expression from the Calculate database composed of real isolated symbols and (b) an example of flowchart in FC database.

Table 1a
Multi-stroke symbol learning on two databases. (a) Graphical symbols are composed of different stroke numbers on two databases.

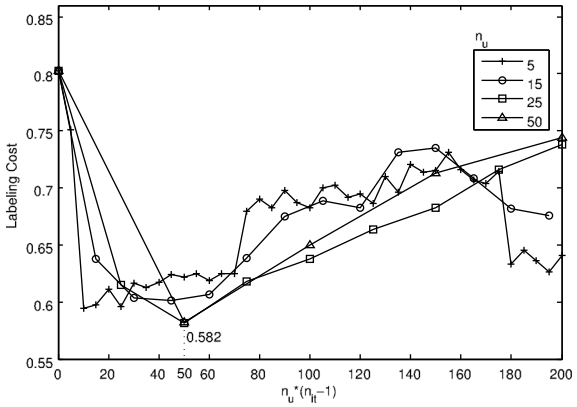
Stroke number		1	2	3	4	5	6	7	8	10
Calculate	Training	2981	2180	234	24	3	4	1		
	Test	1692	1207	114	11	11				
FC	Training	739	1486	745	528	101	32	8	2	
	Test	507	1005	569	282	92	30	7	1	1

$C_{label} = \frac{9+13-9}{13} = 1$. From this example, we fail to label symbols in two cases. Firstly, during the labeling on the codebook in Fig. 6(b), we cannot give labels to "C1", "C4", "C7" since these are sub-part of symbols. Secondly, a context-dependent problem leads to a labeling failure. In the "C9" of Fig. 6(b), "◯" in "→ ◯" can be labeled with either "terminator" or "connection". In the next section, we will present two handwriting datasets to assess our symbol labeling.

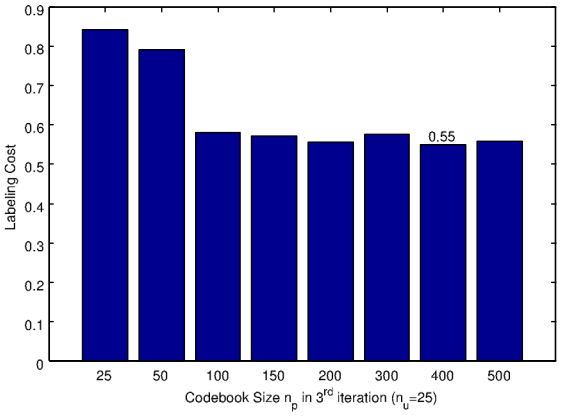
5.2. Dataset description

The first simple database is a synthetic handwriting database named Calculate Awal et al. (2010) of realistic handwritten expressions synthesized from isolated symbols. The expressions in Calculate are produced according to the grammar $N_1\ op\ N_2 = N_3$ where N_1, N_2 and N_3 are numbers composed of 1, 2 or 3 digits from $\{0, 1, \ldots, 9\}$. The distribution of number of digits for $N_{i=\{1,2,3\}}$ is 70% of 1 digit, 20% of 2 digits and 10% of 3 digits randomly. Furthermore, *op* represents the operators $\{+, -, \times, \div\}$. Fig. 8(a) shows an example in Calculate with N_1, N_2, N_3 and *op* containing 3 digits, 1 digit, 2 digits and "×" respectively. Calculate is composed of a training part and a test part from 897 expressions by 180 writers and 497 expressions by 100 writers respectively.

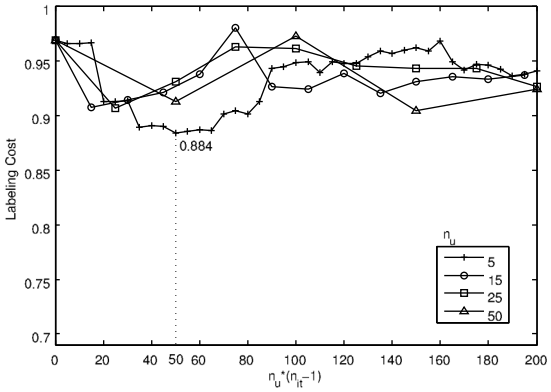
The second more complex handwriting database is a realistic handwritten flowchart database named FC database (Awal et al., 2011). We use only the six different graphical symbols that represent the basic operations (data, terminator, process, decision, connection, arrows) without any handwritten text, as displayed in Fig. 8(b). A total number of 419 flowcharts are written by 36 writers. Those flowcharts are divided into a training part and a test part. Table 1a shows the distribution on different multi-stroke



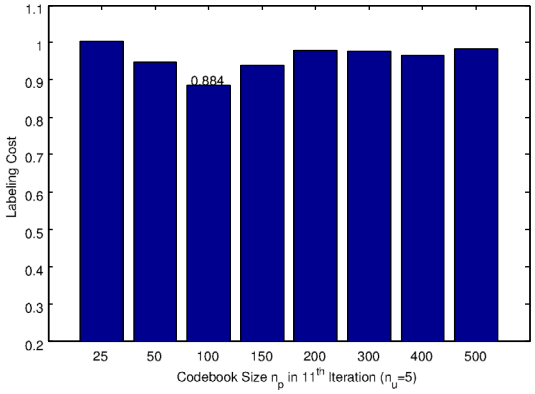
(a) Labeling cost with a fixed number of 100 prototypes and different numbers of discovered lexical units n_u during the iterative procedure on the training part of Calculate database



(b) Labeling cost with different numbers of segment prototypes (codebook sizes) in 3^{rd} iteration on training part of Calculate database (n_u = 25)



(c) Labeling cost with a fixed number of 100 prototypes and different number of discovered lexical units n_u during the iterative procedure on the training part of FC database



(d) Labeling cost with different numbers of segment prototypes (codebook sizes) in 11^{th} iteration on training part of FC database (n_u = 5)

Fig. 9. The experiment results on two databases.

symbols. Our proposed method will be trained on the training part of two databases. The learned parameters are then evaluated on the test part.

5.3. Results

Because of the limited paper length, we present only the evaluation of the number of discovered lexical units n_u in each iteration, and of the codebook size n_p. The other variables are predefined (considering the results on the training part and running time trade-off). The number of closest strokes n_c to create the relational graph in Section 3.1 (Li et al., 2011) is 2. The number of spatial relation prototypes n_{sr} is 5 (Li et al., 2012).

On the training part of two databases, we evaluate the labeling cost with a fixed number of $n_p = 100$ segment prototypes (codebook sizes) during the iterative learning procedure with different numbers of discovered lexical units n_u in each iteration. The best n_u in each iteration will be obtained, and then we test different numbers of segment prototypes n_p. Using the best n_p and best n_u, the system is run on the test part of the two databases to evaluate its robustness.

Using the predefined parameters, Fig. 9(a) illustrates the labeling cost with the fixed number of 100 prototypes during iterative learning procedure using different n_u on the training part of database Calculate. The x-axis denotes

$$n_u \times (n_{it} - 1), \tag{8}$$

the total number of lexical units where n_{it} is the iteration number, and the y-axis shows the labeling cost. The zero discovered lexical unit in x-axis means that the codebook is composed of single stroke codebook as it is the case in the 1st iteration. We can that the lower labeling costs are in the range between 10 and 80 discovered lexical units. The codebook using $n_u = 25$ at the 3rd iteration ($50 = 25 \times (3 - 1)$ in x-axis) reports the lowest labeling cost 58.2%. Only 58.2% of the strokes have to be labeled. Table 1b gives more details on correctly labeled symbols in terms of the stroke number per symbol. The labeling cost rises after the 3rd iteration. Since single-stroke and two-stroke symbols are merged into bigger segments, the Hausdorff distance cannot correctly group more complex multi-stroke segments during hierarchical clustering. More and more single-stroke and two-stroke symbols cannot be correctly labeled.

We test then different codebook sizes n_p by fixing $n_u = 25$ and 3 iterations in Fig. 9(b). We run the system from 1st to 3rd iteration for each n_p respectively, and the labeling cost in 3rd iteration is shown in Fig. 9(b). The labeling cost remains stable roughly 58% after 100 prototypes. The lowest labeling cost is 55% using $n_p = 400$. When applied on the test part, we got a labeling cost of 58.2% using $n_p = 400$ segment prototypes and 3 iterations with $n_u = 25$. The correctly labeled symbols are distributed as 1310 symbols of 1 stroke, 546 symbols of 2 strokes, and 2 symbols of 3 strokes.

Table 1b
Multi-stroke symbol learning on two databases. (b) Number of multi-stroke symbols correctly labeled on the training part of Calculate database ($n_u = 25$).

n^u		Stroke number			
$n_u \times (n_{it} - 1)$	n_{it}	1	2	3	4
0	1	1718	0	0	0
25	2	1392	940	0	2
50	**3**	**1496**	**1027**	**5**	**2**
75	4	1224	1003	7	1
100	5	1283	896	7	1
125	6	1207	839	2	1
150	7	949	885	5	1
175	8	807	829	3	0
200	9	590	840	1	1

Table 1c
Multi-stroke symbol learning on two databases. (c) Number of multi-stroke symbols correctly labeled on the training part of FC database ($n_u = 5$).

n^u = 5		Stroke number				
$n_u \times (n_{it} - 1)$	n_{it}	1	2	3	4	5
0	1	371	0	0	0	0
25	6	481	188	0	12	0
50	**11**	**518**	**280**	**12**	**14**	**0**
75	16	442	239	10	14	0
100	21	240	93	51	15	0
125	26	219	82	53	21	4
150	31	187	44	44	22	6
175	36	232	82	49	34	5
200	41	217	101	56	35	5

The same experimental protocol is run on the more complex FC database. On the training part of FC database, the labeling cost is displayed in Fig. 9(b) using a fixed number of 100 prototypes during iterative learning procedure with different n_u. Comparing the fluctuation between Fig. 9(a) and (b), the performance is not as stable because of the more complex flowchart symbols. Fig. 9(b) shows that the lowest labeling cost of 88.4% is attained in the 11th iteration using $n_u = 5$. It means 50 discovered lexical units in x-axis, which is same with that on the Calculate database. More precise details on stroke number per symbol are shown in Table 1c. Even more and more more-than-three-stroke symbols are correctly labeled after the 11th iteration, decreasing of correctly labeled single-stroke and two-stroke symbols lead to a lower labeling cost. Fig. 9(d) shows that $n_p = 100$ reports the lowest labeling cost 88.4% using $n_u = 5$ and $n_{it} = 11$. On the test part of FC database, a high labeling cost of 99.5% is reported in 11th iteration since the learned configuration is not well fit for the test part. The distribution of correctly labeled symbols is that 127 symbols of 1 stroke, 19 symbols of 2 strokes, 2 symbols of 3 strokes, and 2 symbols of 4 strokes.

6. Conclusion and discussion

In this paper, we propose an iterative learning framework to assist the handwritten graphical symbol labeling. Each stroke is initialized as a segment. A relational graph between the segments is then built. We quantify the segments (nodes) and the spatial relations (edges). The repetitive sub-graphs (symbols) are extracted according to the Minimum Description Length (MDL) principle, and are merged as the new segments for the next iteration. As a result of the quantization of the segments, a visual codebook is produced. The visual codebook is organized by the center segment in each cluster so that the visual codebook can be labeled by a user. Furthermore, we proposed a labeling cost to evaluate this symbol labeling process.

Our approach reports the lower labeling cost 58.2% on the training part of Calculate database. It means that our work can ease the human workload largely. This mathematical expression database contains 54.9% single-stroke symbols. The single-stroke codebook reports the labeling cost 80.2% in 1st iteration. Our multi-stroke symbol discovery procedure can reduce the labeling cost by 22%. The result is very attractive and interesting on this database.

But the labeling cost is higher on the more challenging FC database which contains more multi-stroke symbols: 79.7% symbols are multi-stroke on the training part. The crucial problem is how to find out a better symbol segmentation. Indeed our experiments show that many clusters contain the most frequent combinations of sub-parts of symbol on complex database. To avoid such unwanted combinations, we have to study a more precise graphical

symbol detection criteria based on the MDL principle. Another axis to improve our approach is to enhance the matching distance between two sets of strokes during the clustering. The Hausdorff distance uses only the two sets of points instead of the two sets of sequences (strokes); the information of sequences is lost.

References

Alexander Clark, C.F., Lappin, S., 2010. The Handbook of Computational Linguistics and Natural Language Processing. Wiley-Blackwell.

Awal, A.M., Mouchère, H., Viard-Gaudin, C., 2010. A hybrid classifier for handwritten mathematical expression recognition. In: Proc. Electronic Imaging: Document Recognition and Retrieval XVI, vol. 7534, p. 753410.

Awal, A.M., Feng, G., Mouchère, H., Viard-Gaudin, C., 2011. First experiments on a new online handwritten flowchart database. In: Document Recognition and Retrieval XVIII, vol. 7874, pp. 78740A1–10.

Baird, J.C., 1970. Psychophysical analysis of visual space. Pergamon Press, Oxford, London.

Bouteruche, F., Macé, S., Anquetil, E., 2006. Fuzzy relative positioning for on-line handwritten stroke analysis. In: 10th Internat. Workshop on Frontiers in Handwriting Recognition, 2006.

Bulacu, M., Schomaker, L., 2006. Combining multiple features for text-independent writer identification and verification. In: Proc. 10th Internat. Workshop on Frontiers in Handwriting Recognition, pp. 281–286.

Bulacu, M., Schomaker, L., Brink, A., 2007. Text-independent writer identification and verification on offline arabic handwriting. In: Internat. Conf. on Document Analysis and Recognition, pp. 769–773.

Chartrand, G., 1985. Introductory Graph Theory. Dover Publications.

Clementini, E., 2009. A conceptual framework for modelling spatial relations. Ph.D. Thesis, INSA, LYON.

Cook, D.J., Holder, L.B., 1994. Substructure discovery using minimum description length and background knowledge. J. Artif. Intell. Res. 1, 231–255.

Cook, D.J., Holder, L.B., 2011. Substructure discovery using examples. <http://ailab.wsu.edu/subdue/>.

Delaye, A., Mac, S., Anquetil, E., 2009. Modeling relative positioning of handwritten patterns. In: 14th Biennial Conf. of the Internat. Graphonomics Society, pp. 152–156.

Francis, N.W., Kučera, H., 1982. Frequency Analysis of English Usage: Lexicon and Grammar, vol. 18. Houghton Mifflin.

Huttenlocher, D.P., Klanderman, G.A., Rucklidge, W.A., 1993. Comparing images using the Hausdorff distance. IEEE Trans. Pattern Anal. Machine Intell. 15 (9), 850–863.

Jain, Rajiv, Doermann, David, 2011. Offline writer identification using K-adjacent segments. In: Internat. Conf. on Document Analysis and Recognition, pp. 769–773.

Jonyer, I., Holder, L.B., Cook, D.J., 2000. Graph-based hierarchical conceptual clustering. Internat. J. Artif. Intell. Tools 2, 107–135.

Kohonen, T., 1988. Self-Organization and Associative Memory. Springer-Verlag.

Lance, G.N., Williams, W.T., 1967. A general theory of classificatory sorting strategies: 1. Hierarchical systems. Comput. J. 9 (4), 373–380.

Li, J., Mouchère, H., Viard-Gaudin, C., 2011. Unsupervised handwritten graphical symbol learning-using minimum description length principle on relational graph. In: Knowledge Discovery and Information Retrieval, pp. 172–178.

Li, J., Mouchère, H., Viard-Gaudin, C., 2012. Quantify spatial relations to discover handwritten graphical symbols. In: Document Recognition and Retrieval XIX, pp. 82970F1–8.

Marcken, C.D., 1996. Linguistic structure as composition and perturbation. In: Meeting of the Association for Computational Linguistics, pp. 335–341.

Marcken, C.D., 1996. Unsupervised language acquisition. Ph.D. Thesis, Massachusetts Institute of Technology.

Martinetz, T., Schulten, K., 1991. A neural gas network learns topologies. Artif. Neural Networks, 397–402.

Rhee, T.H., Kim, J.H., 2009. Efficient search strategy in structural analysis for handwritten mathematical expression recognition. Pattern Recognition 42 (12), 3192–3201.

Rissanen, J., 1978. Modeling by shortest data description. Automatica 14 (5), 465–471.

Rucklidge, W.J., 1995. Locating objects using the hausdorff distance. In: Proc. 5th Internat. Conf. on Computer Vision, pp. 457–464.

Tan, G.X., Viard-Gaudin, C., Kot, A.C., 2009. Automatic writer identification framework for online handwritten documents using character prototypes. Pattern Recognition 42 (12), 3313–3323.

Tappert, C.C., Suen, C.Y., Wakahara, T., 1990. The state of the art in online handwriting recognition. IEEE Trans. Pattern Anal. Machine Intell. 12 (8), 787–808.

Treeratpituk, P., Callan, J., 2006. Automatically labeling hierarchical clusters. In: Proc. 6th National Conf. on Digital Government Research, pp. 161–176.

Vajda, S., Junaidi, A., Fink, G.A., 2011. A semi-supervised ensemble learning approach for character labeling with minimal human effort. In: Internat. Conf. on Document Analysis and Recognition, pp. 259–263.

Table des figures

2.1 Exemples de documents structurés traités dans mes travaux. (a) un diagramme, extrait de la base FC [31], (b) un document ancien, extrait du projet CIREFI, (c) des équations, extraites de la base CROHME [1], (d) un geste multipoint, extrait de [19]. 25

2.2 L'écriture en ligne est une séquence de traces constituées de points. Extrait de la Thèse de M. Awal(Fig 4). 26

2.3 Schéma général de la reconnaissance de documents structurés. . . . 27

2.4 Exemple de graphe d'hypothèses. Extrait de Thèse de F. D. Julca-Aguilar (Fig 3.9). 30

3.1 Exemple d'apprentissage global pour l'expression $3+5=8$. Extrait de [2] (Fig. 4). 37

3.2 Modèles gaussiens de la différence de taille de la relation "superscript". Extrait de [2] (Fig. 5). 39

3.3 Deux segmentations d'une expression manuscrite. Extrait de [41] (Fig. 6). 41

3.4 MathML *Content* et *Presentation* de l'expresion $(x+2)^3$. Extrait de [41] (Fig. 2). 41

3.5 Exemple de formulaire (vide) utilisé pour la collecte de l'écriture. . 43

3.6 Architecture pour la reconnaissance d'expressions mathématiques bimodales, extrait de la soumission à HMS. 46

3.7 Structure du système de reconnaissance utilisant une grammaire de graphes. Extrait de Thèse de F. D. Julca-Aguilar (Fig. 3.8). 49

3.8 Utilisation du contexte du symbole comme caractéristique. Extrait de Thèse de F. D. Julca-Aguilar (Fig. 4.7). 50

3.9 Exemple de règle de grammaire de graphes (définition du non terminal `INTEGRAL`). Extrait de la Thèse de F. D. Julca-Aguilar (Fig 3.4(b)). 51

3.10 L'expression $2+2$ est écrite en 4 traits (comme dans la figure 4.1.a). (a) nœuds et arcs reconnus par le BLSMT. (b) arcs ajoutés pour obtenir un graphe valide. Extrait de [52]. 53

3.11 (a) L'expression $\frac{d}{dx}a^x$ écrite en 8 traits ; (b) arbre vérité terrain à couvrir (arcs de segmentation en pointillé) ; (c) ensemble d'arcs générés pour créer les chemins de reconnaissance, les traits pleins sont les arcs corrects, les arcs en pointillés sont en surnombre et les deux en tirets sont manqués. 54

4.1 Une expression manuscrite en 4 traits (a), son interprétation représentée par un *symbol layout tree* qui met en relation des symboles composés de traits (b) et son *stroke label graph* qui est la projection du graphe précédent sur les traits (c). Extrait de [1], Fig. 2. 59

4.2 Histogramme des erreurs de structures pour la sous expression '$x+$' pour notre système testé sur la base de test CROHME 2014. Extrait de [1] (Fig. 5). 61

5.1 Principe de l'extraction de symboles multi-traits. Extrait de [3] Fig. 2. 66

5.2 Construction du graphe représentant un geste multipoint. (a) le geste original (b) interpolation polynomiale, (c) création des nœuds de segments, (d) ajout des arcs de relations spatiales et temporelles. 68

Bibliographie

[63] Kim Marriott, Bernd Meyer, Kent B. Wittenburg. “Visual Language Theory”. In : sous la dir. de Kim Marriott et Bernd Meyer. New York, NY, USA : Springer-Verlag New York, Inc., 1998. Chap. A Survey of Visual Language Specification and Recognition, p. 5-85. ISBN : 0-387-98367-8. URL : http://dl.acm.org/citation.cfm?id=286111.286113.

[64] Ronald Rosenfeld. “Two decades of statistical language modeling : Where do we go from here”. In : *Proceedings of the IEEE*. 2000, p. 2000.

[65] Matthias Zimmermann, Horst Bunke. “Optimizing the integration of a statistical language model in HMM based offline handwritten text recognition”. In : *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. T. 2. IEEE. 2004, p. 541-544.

[66] R. Plamondon, S. N. Srihari. “Online and off-line handwriting recognition : a comprehensive survey”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.1 (jan. 2000), p. 63-84. ISSN : 0162-8828. DOI : 10.1109/34.824821.

[67] Pierre Michel Lallican, Christian Viard-Gaudin, Stefan Knerr. “From off-line to on-line handwriting recognition”. In : 2000, p. 303-312.

[68] Utpal Garain, B. B. Chaudhuri. “Segmentation of touching symbols for OCR of printed mathematical expressions : an approach based on multifactorial analysis”. In : *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*. Août 2005, 177-181 Vol. 1. DOI : 10.1109/ICDAR.2005.212.

[69] A. R. Ahmad, C. Viard-Gaudin, M. Khalid. “Lexicon-Based Word Recognition Using Support Vector Machine and Hidden Markov Model”. In : *2009 10th International Conference on Document Analysis and Recognition*. Juil. 2009, p. 161-165. DOI : 10.1109/ICDAR.2009.248.

[70] S. Lehmberg, H. J. Winkler, M. Lang. "A soft-decision approach for symbol segmentation within handwritten mathematical expressions". In : *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*. T. 6. Mai 1996, 3434-3437 vol. 6. DOI : 10.1109/ICASSP.1996.550766.

[71] Anh Duc Le, Truyen Van Phan, Masaki Nakagawa. "A System for Recognizing Online Handwritten Mathematical Expressions and Improvement of Structure Analysis". In : *Proceedings of 11th IAPR International Workshop on Document Analysis Systems (DAS)*. IEEE. 2014, p. 51-55.

[72] Lei Hu. "Features and Algorithms for Visual Parsing of Handwritten Mathematical Expressions". PhD. Rochester Institute of Techology (Computing et Information Sciences), 2016. URL : http://www.cs.rit.edu/~rlaz/files/LeiHuPhDDissertation_May2016.pdf.

[73] F. Alvaro, J.A. Sanchez, J.M. Benedi. "Recognition of online handwritten mathematical expressions using 2D stochastic context-free grammars and Hidden Markov Models". In : *Pattern Recognition Letters* 35 (2014), p. 56-67.

[74] Scott MacLean, George Labahn. "A new approach for recognizing handwritten mathematics using relational grammars and fuzzy sets". English. In : *International Journal on Document Analysis and Recognition (IJDAR)* 16.2 (2013), p. 139-163. ISSN : 1433-2833. DOI : 10.1007/s10032-012-0184-x.

[75] Bertrand Coüasnon, Aurélie Lemaitre. "Recognition of Tables and Forms". In : *Handbook of Document Image Processing and Recognition*. 2014. DOI : 10.1007/978-0-85729-859-1.

[76] S. Lavirotte, L. Pottier. "Optical formula recognition". In : *Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on*. T. 1. Août 1997, 357-361 vol.1. DOI : 10.1109/ICDAR.1997.619871.

[77] Mehmet Celik, Berrin A. Yanikoglu. "Probabilistic Mathematical Formula Recognition Using a 2D Context-Free Graph Grammar". In : *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR)*. Beijing, China, 2011, p. 161-166.

[78] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, J. Schmidhuber. "A Novel Connectionist System for Unconstrained Handwriting Recognition". In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.5 (mai 2009), p. 855-868. ISSN : 0162-8828. DOI : 10.1109/TPAMI.2008.137.

[79] Ahmad-Montaser A.M. Awal, Romain Cousseau, Christian Viard-Gaudin. "Convertisseur d'équations LATEX2Ink". In : *Colloque International Francophone sur l'Ecrit et le Document*. Rouen, France, oct. 2008, p. 265-280. URL : https://hal.archives-ouvertes.fr/hal-00518460.

[80] Utpal Garain. "Automatic Recognition Of Printed and Handwritten Mathematical Expression". THE INDIAN STATISTICAL INSTITUTE, 2005.

[81] Emilie Poisson. "Architecture et apprentissage d'un système hybride neuro-markovien pour la reconnaissance de l'écriture manuscrite en-ligne". Ecole polytechnique de l'Université de Nantes, 2005.

[82] Scott MacLean, George Labahn. "A Bayesian model for recognizing handwritten mathematical expressions". In : *Pattern Recognition* 48.8 (2015), p. 2433-2445. ISSN : 00313203. DOI : 10.1016/j.patcog.2015.02.017. arXiv : arXiv:1409.5317v1.

[83] Taik Heon Rhee, Jin Hyung Kim. "Efficient search strategy in structural analysis for handwritten mathematical expression recognition". In : *Pattern Recognition* 42.12 (2009). New Frontiers in Handwriting Recognition, p. 3192-3201. ISSN : 0031-3203. DOI : http://dx.doi.org/10.1016/j.patcog.2008.10.036.

[84] Utpal Garain, Bidyut B Chaudhuri. "A corpus for OCR research on mathematical expressions". In : *International Journal of Document Analysis and Recognition (IJDAR)* 7.4 (2005), p. 241-259.

[85] Francisco Álvaro Muñoz, Joan Andreu Sánchez Peiró, José Miguel Benedí Ruiz. *IMEGE : Image-based Mathematical Expression Global Error*. DSIC-PRHLT Technical Report. Universitat Politècnica de València., 2011.

[86] Edward C Kaiser. "Multimodal new vocabulary recognition through speech and handwriting in a whiteboard scheduling application". In : *Proceedings of the 10th international conference on Intelligent user interfaces*. ACM. 2005, p. 51-58.

[87] Anjali Bala, Abhijeet Kumar, Nidhika Birla. "Voice command recognition system based on MFCC and DTW". In : *International Journal of Engineering Science and Technology* 2.12 (2010), p. 7335-7342.

[88] Philippe Smets, Robert Kennes. "The transferable belief model". In : *Artificial intelligence* 66.2 (1994), p. 191-234.

[89] Arthur Chan, G Evandro, S Rita, R Mosur, R Ronald, S Yitao, HD David, S Mike. *The Hieroglyphs : building speech applications using CMU Sphinx and related resources*. 2007.

[90] Yannick Esteve, Paul Deléglise, Sylvain Meignier, Simon Petitrenaud, Holger Schwenk, Loic Barrault, Fethi Bougares, Richard Dufour, Vincent Jousse, Antoine Laurent. "Some recent research work at lium based on the use of cmu sphinx". In : *les actes de CMU SPUD Workshop, Dallas (Texas)*. 2010.

[91] Serge Belongie, Jitendra Malik, Jan Puzicha. "Shape matching and object recognition using shape contexts". In : *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24.4 (2002), p. 509-522.

[92] George Labahn, Edward Lank, Scott MacLean, Mirette S. Marzouk, David Tausky. "MathBrush : A System for Doing Math on Pen-Based Devices". In : *Proc. Document Analysis Systems*. Nara, Japan, 2008, p. 599-606.

[93] Frank D. J. Aguilar, Nina S. T. Hirata. "ExpressMatch : A System for Creating Ground-Truthed Datasets of Online Mathematical Expressions". In : *Proc. Document Analysis Systems*. 2012, p. 155-159.

[94] Jan Stria, Martin Bresler, Daniel Prusa, Václav Hlavác. "MfrDB : Database of Annotated On-Line Mathematical Formulae". In : *Proceedings of the 13th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. Bari, Italy, 2012, p. 542-547.

[95] Bertrand Coüasnon. "Dmos, a generic document recognition method : Application to table structure analysis in a general and in a specific way". In : *International Journal of Document Analysis and Recognition (IJDAR)* 8.2-3 (2006), p. 111-122.

[96] Adrien Delaye, Cheng-Lin Liu. "Contextual text/non-text stroke classification in online handwritten notes with conditional random fields". In : *Pattern Recognition* 47.3 (2014), p. 959-968.

[97] Adrien Delaye, Cheng-Lin Liu. "Multi-class segmentation of free-form online documents with tree conditional random fields". In : *International Journal on Document Analysis and Recognition (IJDAR)* 17.4 (2014), p. 313-329.

[98] Laurent Guichard, Joseph Chazalon, Bertrand Coüasnon. "Exploiting Collection Level for Improving Assisted Handwritten Words Transcription of Historical Documents". In : *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. China, sept. 2011, p. 875-879. DOI : 10.1109/ICDAR.2011.179.

[99] Diane J. Cook, Lawrence B. Holder. "Substructure discovery using minimum description length and background knowledge". In : *Journal of Artificial Intelligence Research* (1994), p. 231-255.

[100] Kenrick Kin, Björn Hartmann, Tony DeRose, Maneesh Agrawala. “Proton : multitouch gestures as regular expressions”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2012, p. 2885-2894.

[101] Hao Lü, Yang Li. “Gesture coder : a tool for programming multi-touch gestures by demonstration”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2012, p. 2875-2884.

[102] Kaspar Riesen, Horst Bunke. *Graph classification and clustering based on vector space embedding*. World Scientific Publishing Co., Inc., 2010.

[103] VeróNica Romero, Alicia FornéS, NicoláS Serrano, Joan Andreu SáNchez, Alejandro H Toselli, Volkmar Frinken, Enrique Vidal, Josep LladóS. “The ESPOSALLES database : An ancient marriage license corpus for off-line handwriting recognition”. In : *Pattern Recognition* 46.6 (2013), p. 1658-1669.

[104] Ian Lenz, Honglak Lee, Ashutosh Saxena. “Deep learning for detecting robotic grasps”. In : *The International Journal of Robotics Research* 34.4-5 (2015), p. 705-724.

[105] Clément Farabet, Camille Couprie, Laurent Najman, Yann Lecun. “Learning Hierarchical Features for Scene Labeling”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (août 2013), p. 1915-1929. DOI : 10.1109/TPAMI.2012.231.

[106] Jianpeng Cheng, Mirella Lapata. “Neural Summarization by Extracting Sentences and Words”. In : *Proceedings of ACL*. 2016.

[107] Yoshua Bengio. “Deep Learning of Representations for Unsupervised and Transfer Learning.” In : *ICML Unsupervised and Transfer Learning* 27 (2012), p. 17-36.