



**HAL**  
open science

# Composition manuscrite de gestes graphiques sur des interfaces tactiles multipoints

Somia Rahmoun

► **To cite this version:**

Somia Rahmoun. Composition manuscrite de gestes graphiques sur des interfaces tactiles multipoints. Traitement du signal et de l'image [eess.SP]. 2012. dumas-00725332

**HAL Id: dumas-00725332**

**<https://dumas.ccsd.cnrs.fr/dumas-00725332>**

Submitted on 24 Aug 2012

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

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



Rapport de stage de Master Recherche  
**Composition manuscrite de gestes graphiques sur des interfaces  
tactiles multipoints**

Rahmoun Somia

Master informatique, spécialité Recherche en informatique

Établissement: ISTIC, Université de Rennes 1

Encadrants: Eric Anquetil et Harold Mouchère

Équipe: INTUIDOC, IRISA

IVC, IRCCyN

Juin 2012

# Résumé

L'utilisation des surfaces tactiles est actuellement de plus en plus répandue dans divers domaines, cette technologie vise à se rapprocher le plus possible d'une utilisation simple et naturelle. L'arrivée du nouveau type de surfaces multipoints, permettant plusieurs point de contact a ouvert un nouveau champ de possibilités pour la composition et la manipulation des documents numériques. Jusqu'ici ces interfaces n'autorisent souvent que des commandes basiques de manipulation, comme par exemple agrandir une image en utilisant deux doigts ou encore passer d'une page à l'autre en glissant le doigt sur la surface de l'écran. Le travail effectué lors de ce stage vise l'exploration de ce domaine en abordant la reconnaissance de commandes gestuelles graphiques multipoints plus évoluées.

Plusieurs approches pour la reconnaissance de forme existent dans la littérature, les méthodes structurelles à base de graphe reposent sur une représentation analytique des formes, modélisant les différentes parties de la forme ainsi que les relations existantes entre ces parties. Nous représentons les gestes multipoints à reconnaître, sous forme de graphes de relations spatiales et temporelles. Pour comparer et reconnaître des symboles, les graphes sont affectés par les imperfections des outils de comparaison disponibles. Les techniques de projection de graphe permettent de pallier cette complexité en cherchant à repositionner les graphes dans un espace vectoriel et permettre ainsi l'utilisation d'outils statistiques. Notre approche est basée sur les techniques de projection de graphes. Nous proposons d'étendre une méthode [44] de l'état de l'art, qui projette un graphe en comptant le nombre d'occurrence de motifs prédéfinis, ces motifs sont des sous-structures de graphes construits indépendamment du système de reconnaissance pour lequel ils sont employés. Nous proposons d'extraire ces motifs directement de l'ensemble des graphes manipulés, en sélectionnant les sous-structures de graphe les plus fréquentes dans la base de graphes considéré. Ce qui permet un lien plus fort avec la nature des graphes manipulé.

**Mots-clés :** apprentissage et reconnaissance de formes, surface tactile, composition multipoint, graphes, relations spatiales et temporelles.

# Table des matières

Introduction générale.....	4
1. Composition sur surfaces sensibles ou tactiles.....	5
Introduction.....	5
1.1. L'encre électronique.....	5
1.2. Interfaces de saisie monopoints.....	5
1.2.1. Geste mono-stroke.....	6
1.2.2. Geste multi-stroke .....	6
1.3. Interfaces de saisie multipoints .....	7
1.1. Manipulation tactile multipoint : état de l'art.....	8
Conclusion .....	8
2. Systèmes de reconnaissance en ligne.....	10
Introduction.....	10
2.1. Reconnaissance de gestes multipoints : état de l'art.....	10
2.2. Problématique.....	11
2.3. Systèmes de reconnaissance en-ligne de formes monopoints : état de l'art.....	12
2.3.1. Approche « statistique » de reconnaissance de symboles .....	12
2.3.2. Approche « structurale » de reconnaissance de symboles.....	12
2.3.3. Reconnaissance de symboles par alignement de séquences .....	16
Conclusion .....	17
3. Contribution.....	18
Introduction.....	18
3.1. Architecture du système proposé.....	18
3.2. Description des graphes utilisés .....	19
3.2.1. Construction des graphes.....	19
3.2.2. Etiquetage des graphes .....	20
3.3. Projection des graphes .....	24
3.4. Phase d'apprentissage.....	26
3.5. Phase de reconnaissance.....	27
Conclusion .....	27
4. Expérimentations .....	28
4.1. Objectifs .....	28
4.2. Base de gestes.....	28
4.3. Protocole.....	29

4.3.1.	Evaluation de l'importance de l'information topologique.....	30
4.3.2.	Evaluation de l'importance de l'information liée à l'étiquetage .....	30
4.3.3.	Découpage de l'ensemble des données .....	30
4.4.	Expériences .....	31
	Conclusion .....	31
	Conclusion générale .....	32
	Bibliographie.....	33

# Introduction générale

Lorsqu'un utilisateur souhaite composer un document numérique (schéma, diagramme ou autres), plusieurs options s'offrent à lui. La première, qui est aujourd'hui la plus courante, consiste à utiliser des logiciels dédiés basés sur une interaction clavier/souris. Ces logiciels offrent de nombreuses possibilités de représentation et fournissent de bons résultats. Mais ils imposent à l'utilisateur un apprentissage approfondi pour la manipulation du logiciel et une mémorisation des fonctionnalités offertes. La seconde consiste à dessiner le document sur une feuille de papier, de la numériser et d'utiliser un logiciel d'interprétation d'images dédié pour reconnaître le document. Cette méthode ne demande aucun effort à l'utilisateur, pour la composition du document, mais nécessite un temps de traitement qui peut être élevé avec un résultat d'interprétation qui n'est souvent pas très satisfaisant [1-2]. La troisième méthode, celle qui nous intéresse dans ce travail, est la composition directe, à la volée, du document sur surfaces tactiles. Cette méthode consiste à composer le document à main levée directement sur le système informatique et à utiliser un logiciel d'interprétation en-ligne de tracés manuscrits. Cette technologie vise à se rapprocher le plus possible d'une utilisation simple et naturelle, non pas en demandant à l'utilisateur d'apprendre son fonctionnement, mais en adaptant le fonctionnement aux utilisateurs. Ces systèmes de reconnaissance de formes commencent à faire leur apparition [3]. Ces interfaces sont dites *orientées stylo* et n'autorisent souvent qu'un seul point de contact pour l'interaction.

L'arrivée sur le marché en 2007 des nouvelles tables et tablettes tactiles *multi-points* (permettant plusieurs points de contact), ont ouvert un nouveau champ de possibilités d'usage pour la composition et la manipulation des documents numériques. Jusqu'ici ces interfaces n'autorisent souvent uniquement que des commandes basiques de manipulation, comme par exemple agrandir une image en utilisant deux doigts ou encore passer d'une page à l'autre en glissant le doigt sur la surface de l'écran. Nous visons à travers ce travail l'exploration de ce domaine en approfondissant l'élaboration, la modélisation et la reconnaissance de commandes gestuelles graphiques multi-points plus évoluées.

Ce rapport est composé de quatre parties, le premier chapitre aborde le contexte générale de la composition sur surfaces sensibles ou tactiles, nous présentons dans cette section les deux types d'interfaces existantes dont les surfaces tactiles multi-points auxquelles nous nous intéressons dans ce travail, nous parlerons également de leurs utilisations dans le domaine de l'IHM (Interaction Homme Machine). Dans le chapitre suivant, nous présentons le domaine de la reconnaissance de forme qui est au cœur de notre travail de recherche. Nous présentons ici les différentes approches de l'état de l'art en ce qui concerne la représentation et la reconnaissance des formes. Le troisième chapitre, présente notre contribution, où nous détaillons le système de reconnaissance de gestes multi-points que nous proposons. Et enfin, dans le dernier chapitre nous présentons le protocole permettant de tester notre système.

Ce travail a été effectué au sein de l'équipe *INTUIDOC* (INTuitive User Interaction for DOCUMENT) du laboratoire *IRISA* (Institut de Recherche en Informatique et Systèmes Aléatoires) de Rennes, en collaboration avec l'équipe *IVC* (Images et Vidéo Communications) du laboratoire *IRCCyN* (Institut de recherche en communications et cybernétique) de Nantes.

# 1. Composition sur surfaces sensibles ou tactiles

## Introduction

Nous commençons notre rapport par présenter le contexte général des dispositifs d'acquisition sensitif. Nous expliquons toutes les notions de base qui nous permettront de situer la problématique que nous cherchons à soulever à travers notre travail. Nous dégageons de ce chapitre trois principales sections ; les interfaces de saisie monopoints : qui permettent un seul point de contact, les interfaces multipoints qui nous intéressent et qui supportent plusieurs points de contact à la fois, et nous terminons notre chapitre par la présentation de quelques travaux de l'état de l'art qui exploitent des interfaces multipoints pour le développement de diverse applications.

### 1.1. L'encre électronique

Le résultat de l'acquisition des tracés dessinés par un utilisateur, généralement à l'aide d'un stylo sur une surface dédiée, est un signal dit *en-ligne*. Le terme *encre électronique*, est aussi souvent utilisé pour désigner le rendu visuel de ce signal. Le terme *tracé* désigne les coordonnées des points, ordonnées dans le temps, représentant le parcours du stylo entre un poser et un lever. Les tracés que nous manipulons sont représentés comme une séquence de segments reliant ces points. La Figure 0-1 présente un exemple de tracé représenté à l'aide du signal en-ligne.

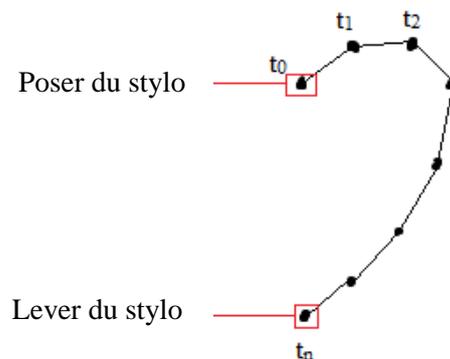


Figure 0-1 Illustration du signal en-ligne

D'autres informations, telles que la pression du stylo sur la surface, son inclinaison, etc. sont parfois disponibles. Notons que le signal en-ligne représente nécessairement des tracés manuscrits [4].

### 1.2. Interfaces de saisie monopoints

Les interfaces d'interaction dites *monopoint* sont des surfaces tactiles qui ne permettent qu'un seul point de contact : un stylo ou un doigt. Les Figure 0-2 Figure 0-3 illustrent les deux types d'interface monopoint.



Figure 0-2 Interface de composition orientée stylo [4]

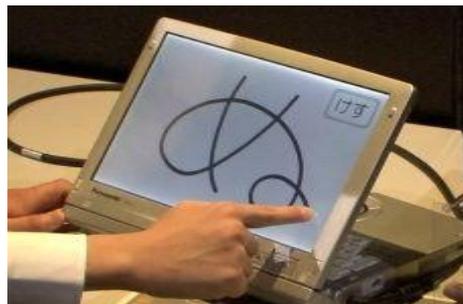


Figure 0-3 Utilisation du doigt pour la composition

Un mot ou un symbole saisi peut être représenté par un seul tracé, ou par une combinaison de ces derniers. Sur la Figure 0-2 la lettre *C* est représentée par un seul tracé, on parle de *geste mono-stroke*, alors que le mot *IMADOC* est représenté par la combinaison de six tracés, cette composition est appelé *geste multi-stroke*. Nous présentons plus en détail ces deux typologies de tracés dans la suite de ce document.

### 1.2.1. Geste mono-stroke

Les premiers systèmes de reconnaissance de formes en ligne proposés étaient généralement *mono-stroke* (voir Figure 0-4). Cette approche consiste à associer à chaque symbole du langage visuel considéré, un tracé manuscrit permettant de le dessiner. Cette approche très simple est aujourd'hui encore très souvent utilisée.



Figure 0-4 Geste mono-stroke : le symbole musical est représenté par un seul stroke

Parmi les travaux de reconnaissance de formes basés sur les approches *mono-stroke*, citons : le système de conception de diagrammes UML [5] ; Le système de reconnaissance de formules mathématiques [6] ; le système permettant la conception d'interfaces graphiques SILK [7]. Dans le domaine de la composition musicale, nous pouvons citer le système de Ng et al, appelé *Presto* [8], et celui d'Anstice et al [9]. Ces systèmes consistent à définir des formes spécifiques pour chaque symbole reconnu à partir d'un seul trait.

Les approches *mono-stroke* restent encore loin de l'objectif initial qui est l'adaptation du système aux habitudes des utilisateurs. Celles-ci demandent une forte implication de la part de l'utilisateur [10], car celui-ci doit souvent apprendre une nouvelle façon de dessiner les symboles d'un domaine.

### 1.2.2. Geste multi-stroke

L'approche qui a suivi le *mono-stroke* et qui est actuellement beaucoup utilisée dans les systèmes de composition manuscrite sur surface tactile, est celle du *multi-stroke* (voir Figure 0-5). Contrairement à la première approche où un symbole doit être saisi par un seul trait, cette méthode

donne la possibilité de dessiner un symbole en plusieurs traits. Ces traits seront combinés pour former le symbole. Le regroupement des traits est un problème complexe.

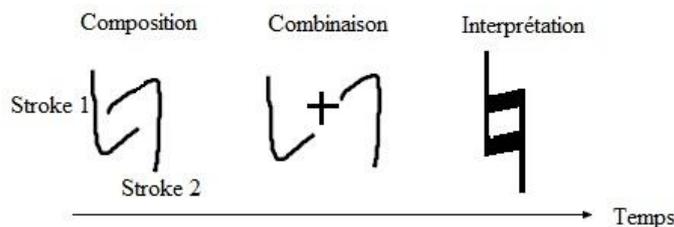


Figure 0-5 Geste multi-stroke : le symbole musical est représenté par la combinaison des deux strokes 1 et 2

Plusieurs approches de combinaison ont été utilisées dans la littérature. La première méthode demande une forte implication de la part de l'utilisateur, en lui demandant de préciser chaque début et fin de symbole en pressant un bouton par exemple : citons les travaux de Kurtoglu et al [11] sur la composition des circuits pour la physique. Une autre approche consiste à définir un délai minimum entre la saisie consécutive de deux symboles différents : la fin de la composition d'un symbole et le début de celle d'un autre sont espacés d'une durée dépassant un seuil prédéfini [4]. Cette approche est utilisée par Fonseca et al [12] pour leur système de reconnaissance de formes géométriques. La difficulté de cette méthode réside dans le choix du délai entre la saisie de deux symboles : trop petit, cela oblige l'utilisateur à saisir rapidement, ce qui peut générer des erreurs, trop long l'utilisateur est contraint d'attendre entre chaque symbole. Bien que ces approches soient moins contraignantes que les approches mono-strokes, elles nécessitent toujours une implication forte de la part de l'utilisateur [4].

Après avoir regroupé les traits qui forment un symbole, le problème est maintenant d'interpréter le symbole composé de ces traits. La troisième méthode de regroupement est une approche automatique qui combine à la fois regroupement et reconnaissance. Elle consiste à décomposer chaque forme saisie en un ensemble de primitives, cette phase nécessite un système de reconnaissance permettant de bien segmenter. Les primitives seront combinées par la suite pour former le symbole. Nous présentons l'approche en détail dans la section « Approches structurelles à base de grammaire ».

### 1.3. Interfaces de saisie multipoints

Les surfaces d'interaction multipoints autorisent contrairement à celle monopoints, plusieurs points de contacts simultanés (Voir Figure 0-6). Cette technologie est actuellement beaucoup utilisée dans les téléphones portables et sur les tablettes numériques. Leurs utilisations résident principalement dans des dispositifs de navigation et de commandes gestuelles très simples.

L'exploitation de la technologie d'acquisition multipoint dans le contexte des systèmes de reconnaissance de formes est au cœur de ce travail. Nous allons donc étudier de plus près la nature des tracés multipoints. Les gestes multipoints sont composés de plusieurs tracés, un pour chaque trajectoire induite par chacun des points de contact (doigt). Un geste multipoint se caractérise donc non seulement par la forme graphique de chacune de ses traces mais aussi par leur positionnement relatif, leur synchronisation ou encore leur dynamique relative. Les Figure 0-6 et 1-7 illustrent respectivement le tracé et le geste multipoint.



Figure 0-6 L'illustration du tracé multipoint

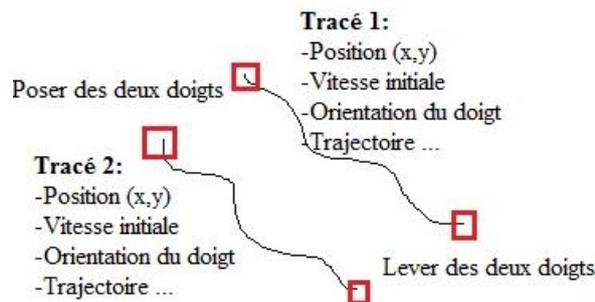


Figure 0-7 Caractéristiques du geste multipoint

Nous pouvons donc, dans un premier temps faire le rapprochement avec les gestes multi-strokes : un symbole est représenté par la combinaison de plusieurs traits. Le geste multipoint est cependant beaucoup plus riche car il ajoute à cette dimension, les notions liées à la synchronisation des différents tracés et à leur positionnement dans l'espace temporel.

Nous avons montré précédemment que l'encre électronique monopoint est la séquence de points ordonnés dans le temps. Ces points appartiennent à l'espace de deux dimensions ( $X, Y$ ). Pour l'encre électronique multipoint, les points appartenant aux différents tracés d'un geste, ne sont pas toujours ordonnés dans le temps, celle-ci peuvent se présenter en même temps. L'information temporelle est alors indispensable comme troisième coordonnée des points de l'encre électronique multipoint.

### 1.1. Manipulation tactile multipoint : état de l'art

Nous remarquons actuellement un nombre considérable de travaux sur l'interaction multipoint dans la communauté de l'interaction homme-machine (IHM): manipulation tactile multipoint (zoom, coverflow, rotation...). Citons par exemple les travaux de Chiu et al [13] qui proposent une interface multipoint permettant de plier/déplier de long documents.

Les interfaces de composition multipoint laissent la liberté aux utilisateurs dans la manière de saisir. Wu et al [14] proposent un ensemble de principes pour la conception de gestes. Leur système offre la possibilité aux utilisateurs de définir les gestes qu'il souhaite employer, pour l'interaction. Ainsi que la possibilité de réutiliser ce geste pour un autre type d'interaction.

Wang et al [15] suggère la nécessité de définir pour chaque domaine d'application l'ensemble des gestes à utiliser. Dans ce contexte, Frisch et al [16] font une étude sur les préférences des gestes qu'utilisent des personnes dans la création de diagramme UML sur une surface tactile multipoint (l'utilisation d'une seule main, du stylo, l'interaction des deux mains, l'interaction entre la main et le stylo, etc.).

Ces publications ne proposent pas de solution générique pour reconnaître des gestes multipoints, mais se focalise sur leur utilisation dans le domaine de l'IHM (Interaction Homme Machine)

## Conclusion

Nous avons présenté dans ce chapitre le contexte de la composition sur surfaces sensibles. Nous avons introduit les deux types de surfaces : les monopoints et les multipoints, tout en décrivant pour

chaque type de surface la nature des gestes pour l'interaction. Le chapitre suivant abordera le domaine de la reconnaissance de formes, qui est au cœur de notre travail de recherche.

## 2. Systèmes de reconnaissance en ligne

### Introduction

Ce chapitre aborde la problématique de la reconnaissance de gestes multipoints que nous cherchons à soulever. Nous commençons par exposer les travaux actuels dans le domaine. Nous présentons ensuite, les différents systèmes de reconnaissance de formes existants. Ces systèmes sont utilisés actuellement pour la reconnaissance de symboles monopoints. Notre objectif est d'utiliser l'un de ces systèmes pour la reconnaissance de gestes multipoints. Nous étudions en détail chacun de ces systèmes, en s'appuyant sur les points forts et les points faibles de chacun, ce qui nous permettra de sélectionner le système le plus adapté à notre problématique.

### 2.1. Reconnaissance de gestes multipoints : état de l'art

La reconnaissance de signaux (formes) complexes multipoints, représente un domaine encore assez peu exploré à l'heure actuelle. La plupart des travaux se basent sur l'étude des gestes<sup>1</sup>.

Wang et al [15] définissent le geste par le nombre de points de contact (en générale les doigts), la position de chaque doigt sur l'interface et la distance entre chaque position. Les deux dernières caractéristiques doivent être mises à jour lors du mouvement. Damaraju et Kerne [17] ont exploité ces mêmes caractéristiques pour reconnaître un ensemble de 40 gestes prédéfinis. Les taux de classifications pour des gestes utilisant deux, trois, quatre et cinq doigts sont respectivement 93.8%, 91.4%, 92.1% et 88.8% en utilisant un système de reconnaissance par HMM (voir Reconnaissance de symboles par alignement de séquences).

Wang et al [18] ont montré que l'orientation des doigts pour faire un geste est une propriété très utile pour améliorer l'interaction. Plusieurs travaux se sont donc concentrés sur cette description. Marquardt et al [19] et Dang et al [20] proposent des approches permettant de reconnaître chaque doigt de la main ainsi que leurs orientations. Ils considèrent dans leurs études, des gestes dédiés à la manipulation (sélection, rotation d'objet, zoom, etc.).

Un système permettant la reconnaissance automatique et la localisation de poser de main, et des classes d'objets (Main, Stylo, etc.) en temps réel, a été proposé par Deselaers et al [21]. Un algorithme de forêts aléatoires est employé qui sélectionne de façon incrémentale l'ensemble minimal de caractéristiques permettant d'atteindre la meilleure discrimination de classe pour une image données. L'algorithme testé sur une base de 45 classes, donne des taux de classification variés ; selon le type de caractéristiques utilisées.

Les systèmes de reconnaissance de gestes que nous avons cités, utilisent des caméras pour l'extraction de caractéristiques à partir des gestes à reconnaître. Ces caméras enregistrent le mouvement de la main saisissant le geste, et extrait des informations sur le nombre de doigts, la direction de chaque doigt, la vitesse de saisie ...

La plupart des interfaces multipoints qui sont actuellement commercialisés, sont équipés de systèmes de reconnaissance qui se basent sur la définition des gestes multipoints introduite par Apple

---

<sup>1</sup> Le geste fait référence ici au mouvement physique de la main lors de l'interaction (position initiale, nombre de doigt utilisé, etc.)

[22]. Ces gestes sont des configurations prédéterminés, qui comprennent un nombre prédéterminé de points de contact. A chaque point de contact est associé un mouvement prédéfini. De cette manière tous les gestes considérés par le système sont au préalable enregistrés, et pour chaque geste, un reconnaiseur dédié est implémenté. Les développeurs enregistrent manuellement les différents gestes saisis et doivent gérer les conflits entre les différents gestes. Par conséquent, l'ajout d'un nouveau geste nécessite une connaissance approfondie sur les gestes existants. Ces systèmes sont performants pour la reconnaissance d'un ensemble réduit de gestes, mais sont moins généralisable, et reste assez difficiles à mettre en place. D'autres systèmes ont tenté de soulever cette difficulté en définissant des langages plus haut niveau [23, 24, 25]. La difficulté de ces langages accroît avec la complexité des gestes considérés, de plus apprendre un nouveau langage est contraignant pour les développeurs.

Hao et al [26] ont mis en place *Gesture-Coder*, un système qui génère des reconnaiseur de geste en apprenant à partir des gestes multipoints saisi. Les auteurs ont remarqué qu'un geste multipoint est une séquence d'évènements (ces évènements sont les positionnements des doigts). De cette manière chaque geste est représenté par une machine d'états, l'ensemble des transitions entre les états représente le mouvement au sein du geste. *Gesture-Coder* apprend à partir des machines d'états de chaque geste multipoint.

Ideum [27] a introduit le langage de balisage "*GestureML*" pour le développement de ces applications multipoints. Un fichier *GestureML*<sup>2</sup> prédéfini est associé à chaque geste, ce fichier contient toutes les informations permettant de caractériser le geste (le nombre de tracés, leur formes géométriques, leurs vitesses de saisie...). Ces informations sont ensuite utilisées pour la reconnaissance des gestes en question.

Ces systèmes capturent les gestes directement de la surface, mais restent moins généralisables du fait que, pour chaque geste un reconnaiseur dédié est développé.

## 2.2. Problématique

Les travaux que nous avons cités se basent sur une définition complète du geste, et un reconnaiseur dédié à chaque geste, ce qui les rend difficilement adaptable à tous types de gestes. D'autres types de travaux que nous avons également cités sont basés sur l'extraction de caractéristiques à partir d'enregistrement vidéo. A notre connaissance, il n'existe pas de système qui exploite la richesse de l'encre électronique pour l'extraction de caractéristiques.

Les gestes multipoints se rapprochent des gestes multi-strokes : un symbole est représenté par la combinaison de plusieurs traits. Le geste multipoint est cependant beaucoup plus riche car il ajoute à cette dimension, les notions liées à la synchronisation des différents tracés et à leur positionnement dans l'espace temporel. Nous pensons donc qu'il serait intéressant d'étudier les techniques de reconnaissance de geste multi-strokes qui utilisent directement l'encre électronique pour l'extraction de caractéristiques. L'idée est de réfléchir à comment nous pouvons utiliser l'une de ces méthodes pour soulever notre problématique, qui est la reconnaissance de gestes multipoints.

---

<sup>2</sup> *GestureML* est basé sur une représentation en *XML*

## 2.3. Systèmes de reconnaissance en-ligne de formes monopoints : état de l'art

La reconnaissance en-ligne de forme à laquelle nous nous intéressons dans ce travail, consiste à identifier chaque classe de symbole considéré saisie manuellement sur une surface tactile. On distingue généralement deux phases : la phase d'*apprentissage* et la phase de *classification* ou *généralisation*. La première phase d'apprentissage consiste à modéliser (apprendre) toutes les classes de symbole d'un domaine précis. Ces modèles sont ensuite exploités en phase de reconnaissance (généralisation) pour affecter une forme inconnue à une classe apprise.

On distingue souvent deux grandes approches de reconnaissance de symboles : les approches dites « *statistiques* » où les formes sont représentées par des caractéristiques globales, et les approches dites « *structurelles* » où cette fois les formes sont décomposées en primitives graphiques qui les constituent. Nous présentons dans les paragraphes suivant, les deux approches de reconnaissance.

### 2.3.1. Approche « statistique » de reconnaissance de symboles

L'approche dite « statistique » consiste à représenter chaque symbole considéré par un vecteur de caractéristiques qui sera donné en entrée à un classifieur. Les caractéristiques considérées ici sont des informations sur la forme globale du symbole. Plusieurs types de classifieur peuvent être utilisés dans ce type d'approche SVM (Séparateur à vaste marge) [28], arbre de décision, réseau de neurones [3].... Cette approche est beaucoup utilisée dans les systèmes de composition mono-stroke, car elle est bien adaptée à la reconnaissance de formes manuscrites isolées [4]. Les classifieurs statistiques donnent en générale les meilleurs résultats de classification.

L'approche peut aussi être utilisée pour la reconnaissance des gestes multi-strokes : citons les travaux de Miyao et Maruyama [29] pour la reconnaissance de symbole musicaux. Le système reconnaît chaque tracé séparément, puis combine les résultats de ces interprétations pour en déduire la classe d'un symbole.

Cette approche pourrait être utilisée pour la reconnaissance multipoint. Mais les descripteurs à considérer pour décrire un geste multipoint (morphologie, structure, dynamique, etc.) devront être plus riches que ceux utilisés pour décrire un geste monopoint. Cela aurait pour effet d'augmenter la taille du vecteur de caractéristiques à considérer, et par conséquent rendra la tâche de classification potentiellement plus complexe.

### 2.3.2. Approche « structurelle » de reconnaissance de symboles

Le principe ici est de représenter les symboles à reconnaître par l'ensemble des primitives graphiques qui les constituent (voir Figure 2-1). Ces primitives sont ensuite utilisées de différentes façons pour la reconnaissance. Chan et al. [30] proposent de modéliser les symboles en fonction d'un ensemble de primitives structurelles prédéfinies (ligne, courbe, petit trait, boucle etc). Ils comparent ensuite ces modèles avec les modèles de la base du système pour essayer de trouver le symbole correspondant.

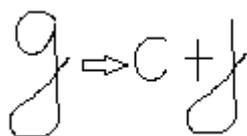


Figure 2-1 Représentation de la lettre manuscrite 'g' par deux primitives graphiques

Les méthodes de reconnaissance automatique de gestes multi-strokes, se composent principalement d'une étape de prétraitement suivie d'une étape de regroupement automatique. La première consiste à décomposer chaque tracé manuscrit en un ensemble de primitives<sup>3</sup> graphiques. La segmentation se fait en des points bien particuliers ; un système de reconnaissance bas-niveau est alors utilisé pour identifier chacun des sous-tracés obtenus comme une primitive graphique [4]. La difficulté réside donc dans la détection des points de segmentations, se sont en générale des points de courbure du tracé [31] associé à des informations de vitesse de ce tracé [32].

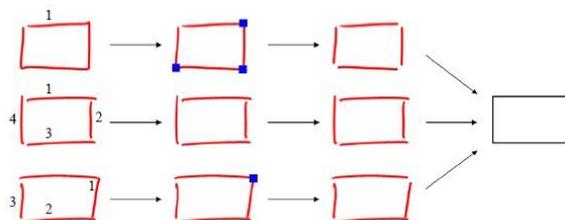


Figure 2-2 Un rectangle peut être dessiné de plusieurs manières différentes et être reconnu à l'aide du même système de reconnaissance (les numéros indiquent l'ordre dans lequel les tracés ont été dessinés, les points bleu ceux de la segmentation lors de la phase de prétraitement) [4]

La deuxième étape consiste à regrouper ces primitives en symboles. Plusieurs méthodes ont été utilisées dans la littérature. Certaines sont basées sur le contexte structurel (proximité graphique) : citons les travaux de Miyao et al [29] pour la composition de partitions musicales où le système combine les tracés séparés par une distance minimum. Ou citons également les travaux de Liao et al [33] utilisant le même principe. D'autres méthodes de regroupement automatique ajoutent le contexte temporel (proximité temporelle) au structurel. Nous citons les travaux de Garain et al [34] pour le regroupement automatique des tracés en symbole au sein des formules de mathématiques. Le système impose à l'utilisateur de terminer la composition d'un symbole avant de passer à un autre.

### 2.3.2.1. Approches structurelles à base de grammaire

Parmi les méthodes de reconnaissance structurelle nous trouvons celles qui sont liées à l'exploitation de formalismes grammaticaux et d'analyseurs syntaxiques associés [4]. Une grammaire est un formalisme qui permet de définir un langage, et donc un ensemble de phrases sur un alphabet donné. L'analyse syntaxique d'une phrase  $X$ , associée à une grammaire  $G$ , consiste à déterminer si  $x$  fait partie du langage de  $G$ . Les symboles sont modélisés en termes de primitives graphiques structurelles prédéfinies, telles que le segment, l'arc, le cercle, etc. Cette méthode a beaucoup été utilisée dans la reconnaissance de symboles séparés [4], et de documents structurés. Plusieurs types de grammaires existent dans la littérature, un état de l'art complet sur les grammaires est disponible dans [4].

Les grammaires bi-dimensionnelles définissent chaque symbole par un ensemble de primitives graphiques, et un ensemble de contraintes de positionnement spatial qui doivent être satisfaites par ces primitives. Nous citons la méthode DALI [35] qui utilise une grammaire bi-dimensionnelle basée sur le contexte structurel du document pour l'interprétation des tracés manuscrits. La première difficulté de la méthode est la nécessité de décrire tout symbole comme une combinaison de telles primitives. L'approche est très adaptée à la modélisation de symboles structurés. Cette approche nous semble plus

<sup>3</sup> Une primitive graphique peut être un segment de droite, un arc, un cercle, etc.

laborieuse lorsqu'il s'agit de décrire des symboles complexes. Ce qui est le cas pour les gestes multipoints que nous cherchons à reconnaître.

La seconde difficulté est liée à la nature manuscrite des tracés qu'il faut segmenter. La Figure 2-3 illustre quelques problèmes liés à la segmentation. Dans (a) la manière de saisie de l'utilisateur produit cinq segments, cela aura un impact sur le système qui utilise une grammaire ne supportant que quatre segments. Dans (b) le point de segmentation est difficile à détecter on parle de sous-segmentation. On parlera aussi de sur-segmentation pour le carré (c). Un autre problème de reconnaissance peut aussi être rencontré, dans la figure 2-3 (d) le segment 2 sera reconnu comme un arc plutôt que droite. Nous concluons qu'une erreur de segmentation peut entraîner l'échec de tout le processus d'analyse.

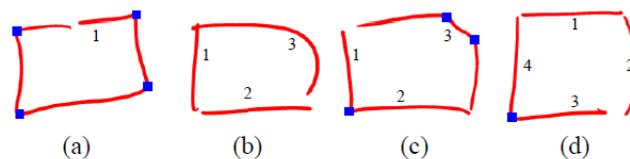


Figure 2-3 Difficultés liées à la segmentation des tracés manuscrits en primitives graphiques

L'utilisation des grammaires pour la reconnaissance de gestes multipoints nous semble assez contraignante et difficile à mettre en place. Nous souhaitons mettre en place un système qui soit capable de reconnaître tous types de gestes et qui soit capable de s'adapter à tout type d'utilisateur.

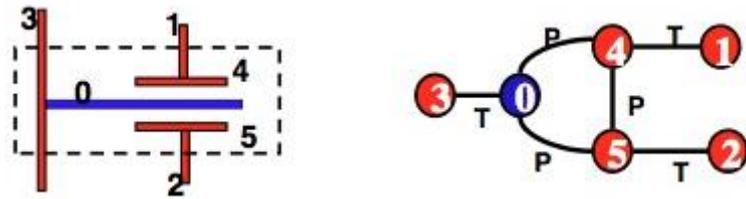
### 2.3.2.2. Approche structurelle à base de graphes

Une autre manière de représenter un symbole à reconnaître en fonction de ses primitives, est la représentation en graphes. La classification consiste à trouver le graphe modèle le plus proche du graphe du symbole à reconnaître, en utilisant généralement une distance inter-graphes [3].

#### 2.3.2.2.1. Représentation structurelle à base de graphe

Généralement, les primitives correspondent aux sommets du graphe et les relations topologiques entre primitives correspondent aux arcs entre ces sommets. [36] propose de représenter les caractères chinois par des graphes dans lesquels les sommets représentent les traits et les arêtes les relations topologiques comme P, X, T ou  $L^4$  dépendant du type d'intersection entre les traits. Dans [37] les quadrilatères et leurs relations de voisinage sont utilisés pour construire un graphe structurel étiqueté. Dans cette représentation structurelle, les primitives constituent les sommets du graphe tandis que les arêtes décrivent la nature de la relation topologique (Figure 2-4) entre primitives (comme parallèle, intersection, jonction, etc.). Xu et al [38] proposent de représenter les primitives structurelles d'un symbole graphique par un graphe de relations spatiales.

<sup>4</sup> P : pour parallèle, X, L et T : pour une jonction en X, L ou T



(a) Image d'un symbole

(b) Son graphe de primitives

Figure 2-4 Une image d'un symbole et sa représentation par un graphe de primitives [37].

La représentation en graphe, nous semble intéressante pour la reconnaissance de gestes multipoints. Les nœuds du graphe pourraient représenter les différents tracés du geste, les arcs décriraient les différentes natures de relations (position, vitesse, trajectoire, etc.) entre les tracés du geste. La description des graphes utilisés sera faite dans la section Contribution.

### 2.3.2.2.2. Reconnaissance des formes à base de graphes

En utilisant cette représentations, le verrou est de déterminer si deux graphes représentent chacun un objet de même type ou de deux types différents ; plus formellement, si deux graphes décrivent des objets appartenant à la même classe. Les premières méthodes de comparaison de graphes s'appuient sur des appariements de graphes. Ces techniques permettent de mettre en évidence que deux graphes sont identiques ou non. Par conséquent, deux graphes présentant certaines différences, suffisamment pour ne pas être appariés, sont jugés dissimilaires. La similarité entre ces deux graphes sera jugée nulle. En reconnaissance de formes, la comparaison se fait de manière moins booléenne. De ce point de vue, les méthodes d'appariement ne répondent donc pas toujours efficacement aux problématiques de la reconnaissance des formes à cause de la variabilité des gestes naturels.

Pour pallier cette lacune, des méthodes proposent la mise en place de distance entre graphes. L'idée générale est de définir la dissimilarité de deux graphes par la distorsion nécessaire pour transformer un graphe en un autre. Le calcul de la distance d'édition est basé sur un arbre de recherche représentant toutes les solutions d'appariements possibles ; chaque nœud de l'arbre représente un appariement (entre nœuds et arcs du premier graphe au deuxième graphe) et chaque chemin de la racine à une feuille, correspond à un appariement possible entre les deux graphes considérés. La complexité calculatoire de cette méthode est exponentielle.

D'autres approches permettant l'utilisation d'outils statistiques pour mesurer une similarité de graphes, ont succédé l'approche de calcul de distance, ce sont les méthodes de projection de graphes<sup>5</sup>. Ces techniques cherchent à encapsuler les informations portées par un graphe dans un vecteur numérique. Ce vecteur numérique peut ensuite être utilisé pour la classification avec le formalisme des techniques de la reconnaissance des formes statistique. Il existe différentes approches pour réaliser cette projection. Nous présentons dans cette section les principales approches.

#### **Les méthodes spectrales**

De nombreux travaux pour la projection de graphes exploitent la théorie spectrale des graphes [39] et caractérisent la topologie des graphes en utilisant les valeurs et les vecteurs propres des matrices d'adjacence ou laplaciennes. Les méthodes spectrales sont sensibles aux bruits ; la

<sup>5</sup> Projection de graphe : en anglais Graph embedding, Le terme anglais est le plus souvent utilisé.

suppression d'un nœud, par exemple, modifie les matrices (d'adjacence ou laplaciennes) et ces erreurs affectent les fonctions de projection. De plus, leur utilisations spécifiques aux graphes non-étiquetés rendent ces approches difficiles à exploiter dans la majorité des applications de reconnaissance de formes. Cette méthode est expliquée en détail dans [40]

### ***Les représentations par dissimilarités***

Le principe est de construire un vecteur de caractéristiques en comparant un graphe à une sélection de graphes dits prototypes. L'idée principale est d'utiliser la dissimilarité pour permettre de séparer une classe d'objets d'une autre, un objet est donc caractérisé par un vecteur de différence aux objets d'autres classes. Riesen et al [41] propose d'utiliser comme mesure de dissimilarité, la distance entre graphes. L'auteur calcule des distances d'édition de graphes entre le graphe à projeter et un ensemble de ' $k$ ' graphes prototypes sélectionnés dans une base. La projection du graphe est donc un vecteur de ' $k$ ' distances. Cette catégorie de projection est très performante, mais la difficulté liée au calcul de distances évoquée précédemment se retrouve dans cette méthode. De plus, le choix des graphes comme prototypes est aussi un paramètre non négligeable puisqu'il détermine la taille du vecteur et sa qualité à bien représenter le graphe dans l'espace vectoriel.

### ***Les sondages de graphes***

Cette technique est basée sur l'extraction de caractéristiques relatives aux données structurées. Ces caractéristiques numériques sont liées aussi bien aux informations sur la topologie (nombre de nœuds, d'arcs, degré des nœuds...) qu'au contenu du graphe (informations liées aux étiquettes. . .). De cette façon, le graphe initial est projeté dans un espace euclidien. Dans [42], sur des graphes non-orientés et non étiquetés, les auteurs calculent le degré de chaque nœud pour ensuite constituer un vecteur trié de degrés. Les attributs de ce vecteur sont le nombre de nœuds de degré 1 pour la première caractéristique, le nombre de nœuds de degré 2 pour la deuxième, etc. [43] proposent une projection dans le cas où les graphes sont orientés, la représentation distingue les demi-degrés intérieur et extérieur. Les auteurs adaptent aussi cette représentation pour prendre en considération les graphes étiquetés, en considérant, en plus du degré des nœuds, les étiquettes des arcs connectés. Récemment, dans [44] l'auteur propose une méthode de projection basé sur l'extraction de sous-structures de 2 nœuds, 3 nœuds... à partir d'un graphe. Le vecteur de caractéristiques est obtenu en comptant les fréquences de ces sous-structures dans le graphe à projeter.

## **2.3.3. Reconnaissance de symboles par alignement de séquences**

Plusieurs systèmes de reconnaissance de symboles utilisent la méthode de mise en correspondance de motifs. Elle consiste à comparer l'objet à reconnaître avec l'ensemble des références des classes de symboles présents dans la base de données. La reconnaissance se fait généralement par calcul de distances [14]. La distance élastique, Dynamique Time Warping (DTW), considère l'aspect temporel dans la comparaison des séquences. C'est une approche simple, qui pourrait s'avérer intéressante dans le cadre de la reconnaissance des gestes multipoints. Une autre méthode, basée sur les modèles de markov caché (Hidden Markov Model, HMM) peut aussi être adoptée pour la reconnaissance de séquences. Pour chaque classe de geste, un HMM peut être appris. Par exemple, un geste peut être décomposé en une séquence de primitives définie par le poser d'un doigt puis la composition d'un cercle au tour de ce doigt, par un autre doigt.

## Conclusion

Nous avons exploré dans ce chapitre le domaine de la reconnaissance de formes en présentant les différentes approches de l'état de l'art. Les travaux que nous avons étudiés dans cette section sont tous basés sur une interaction monopoint. En effet le domaine est bien avancé sur ce type de surface. Nous avons aussi évoqué les travaux actuels sur la reconnaissance multipoint. L'approche qui nous semble la plus intéressante pour soulever notre problématique est celle de la représentation par graphe qui paraît bien adaptée pour supporter la richesse et la nature multicritère du geste multipoint. Pour pouvoir profiter pleinement des performances des méthodes statistiques, nous avons exploré la méthodologie qui consiste à représenter un graphe sous forme de vecteurs de caractéristiques, ce qui nous permettra de tester plusieurs algorithmes de classification par la suite.

## 3. Contribution

### Introduction

Nous présentons dans ce chapitre le système de reconnaissance de gestes multipoints que nous avons proposé. Nous détaillons les différentes étapes du processus pour l'apprentissage et la reconnaissance de ce type de geste.

#### 3.1. Architecture du système proposé

Un geste multipoint se caractérise non seulement par la forme graphique de chacune de ses traces mais aussi par leur positionnement relatif, leur synchronisation ou encore leur dynamique relative. Nous avons jugé que la représentation par graphe est la plus adaptée à ce type de geste. En effet, la puissance de la représentation par graphe réside non seulement dans son pouvoir de modélisation de chaque composante du geste, mais aussi dans le fait de pouvoir exprimer la dynamique relative entre ces composantes.

A travers notre étude bibliographique sur l'utilisation des graphes dans le domaine de la reconnaissance de formes, nous avons retenu que leur utilisation entraîne une problématique de comparaison de graphe. Plusieurs méthodes ont été proposées afin de soulever cette problématique. Une méthodologie qui a attiré notre attention est celle de la projection des graphes, cette méthode permet aussi l'utilisation de classifieurs statistiques qui fournissent en générale les meilleurs résultats de reconnaissance. Nous exploitons cette méthodologie pour le système de reconnaissance que nous proposons. La Figure 3-2 illustre l'architecture globale de notre système.

La reconnaissance en-ligne de formes consiste à identifier chaque classe de symbole considéré saisi manuellement sur une surface tactile. On distingue généralement deux phases : la phase d'*apprentissage* et la phase de *classification* ou *généralisation*. La première phase d'apprentissage consiste à modéliser (apprendre) toutes les classes de symbole d'un domaine précis. Ces modèles sont ensuite exploités en phase de reconnaissance (généralisation) pour affecter une forme inconnue à une classe apprise. Deux étapes précèdent ces deux principales phases, dans notre système : l'étape de construction des graphes et l'étape de projection de ces graphes (voir Figure 3-2). Chaque geste saisi est représenté sous forme de graphes, où les sommets modélisent les différents tracés du geste, et les arêtes, les relations entre tracés (voir Figure 3-1). Les sommets portent l'information de la forme du tracé comme étiquette, les arêtes véhiculent les relations entre tracés. Chaque graphe est ensuite transformé en un vecteur numérique dans la phase de projection. Cette étape du processus nécessite l'utilisation d'un lexique de sous-graphes, la projection consiste à compter le nombre d'occurrence de ces sous-graphes dans le graphe à projeter. L'apprentissage et la reconnaissance des gestes, se font à l'aide de classifieurs statistiques. La Figure 3-1 illustre de façon globale le processus de traitement pour l'apprentissage et la reconnaissance des gestes.

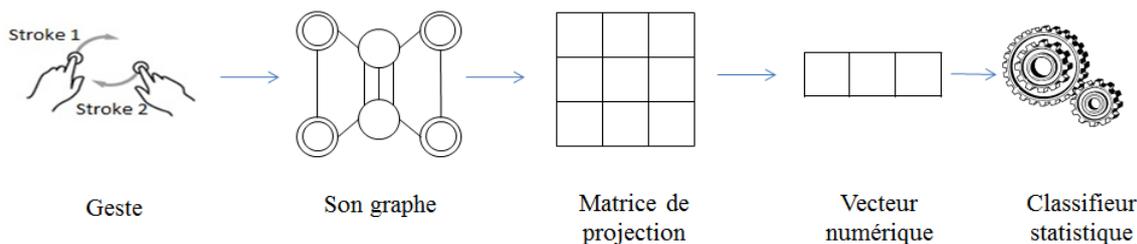


Figure 3-1 Processus d'apprentissage et de reconnaissance des gestes multipoints

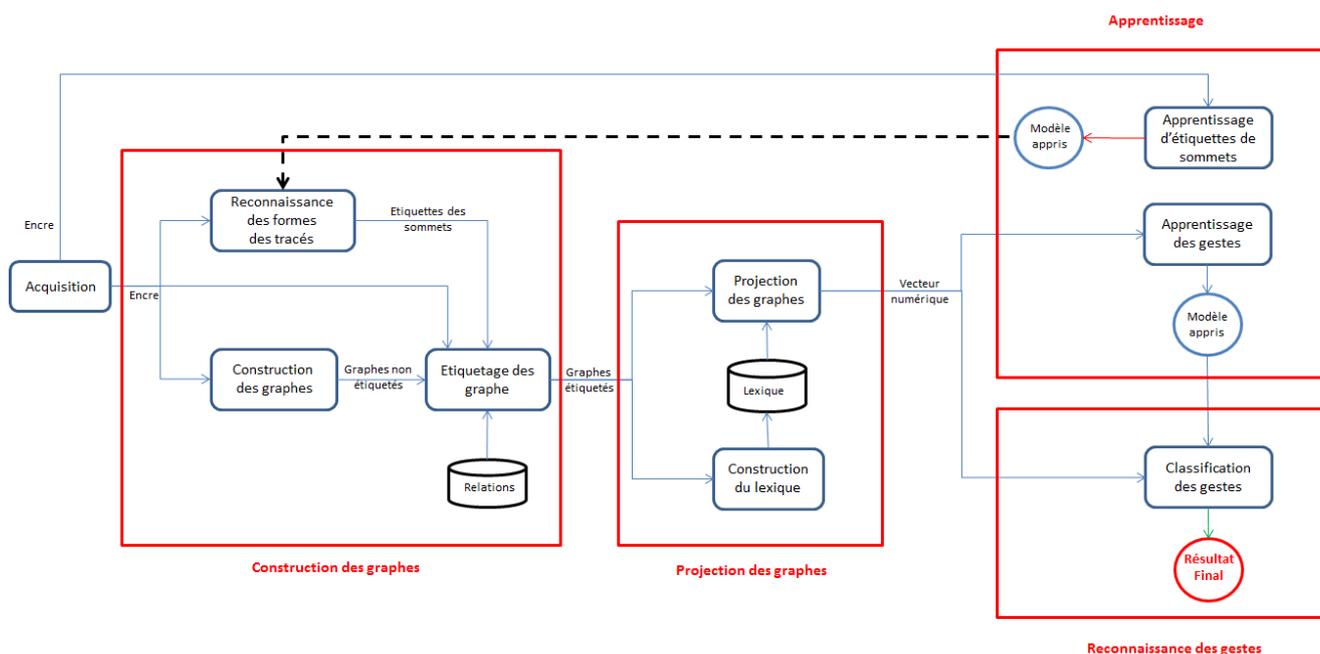


Figure 3-2 Architecture du système proposé

## 3.2. Description des graphes utilisés

### 3.2.1. Construction des graphes

Nous rapprochons les gestes multipoints que nous manipulons aux gestes multi-strokes<sup>6</sup>. Chaque doigt génère un tracé. Nous présentons dans la figure 1 quelques exemples de geste multipoints.

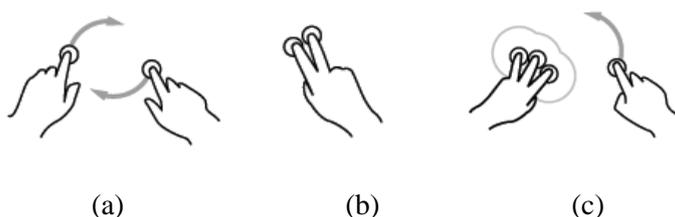


Figure 3-3 Exemples de gestes multipoints. (a)(b) sont des gestes composés de deux traits. (c) est un geste est composé de quatre traits<sup>7</sup>

<sup>6</sup> Les gestes multi-strokes sont expliqués en détail dans 1.2.2 Geste multi-stroke

<sup>7</sup> Figures prises de : <http://gestureworks.com/features/open-source-gestures/> : Open source gesturelibrary

Dans la plupart des techniques de représentation par graphes de l'état de l'art que nous avons citée, chaque sommet désigne une primitive<sup>8</sup> et chaque arête une relation structurelle entre deux primitives. Pour notre approche nous représentons les tracés par deux types de sommets : un type modélisant la primitive et un deuxième type permettant de représenter les extrémités de cette primitive (voir Figure 3-4). Pour chaque tracé trois sommets sont représentés : Deux sommets de type *Extrémité* modélisant le début et la fin du tracé, et un sommet de type *Stroke* pour symboliser le trait dans sa globalité. Cette représentation nous permet d'exprimer les relations spatiales et temporelles entre deux tracés du même geste ainsi qu'entre leurs extrémités. En effet, un geste multipoint est défini par la position de chaque doigt sur la surface et par leur mouvement respectif, au sein du même geste. Le format des graphes que nous proposons permet une représentation complète de la dynamique des gestes multipoints. Nous montrons à travers la Figure 3-4 un exemple de représentation par graphe.

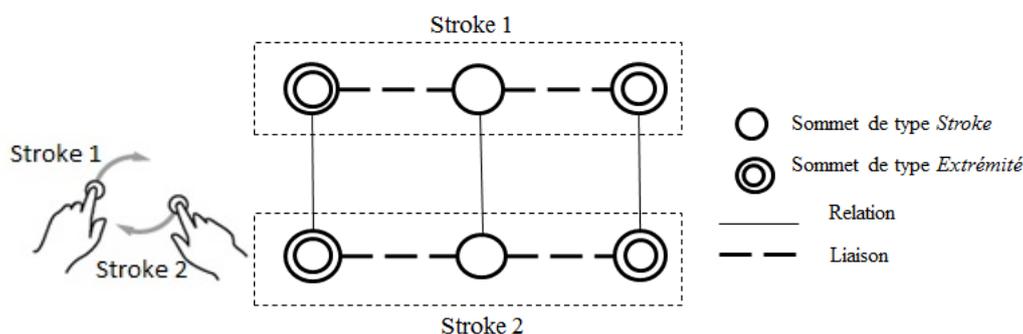


Figure 3-4 Représentation du geste 'rotation' par graphe. Les arêtes de liaison ne portent aucune information. Les relations spatiales et temporelles seront véhiculées par les étiquettes d'arrêtes modélisant des relations.

La structure des graphes change selon la dynamique du geste. Le nombre d'arêtes n'est pas fixé, contrairement au nombre de sommets qui est toujours de trois nœuds par trace. Les arêtes de liaison sont présentes dans tous les graphes. Celles exprimant les relations ne sont pas toujours toutes représentées : les relations entre les sommets de type *Stroke* sont toujours représentées, celle entre les sommets modélisant les extrémités des tracés, ne sont représentées que dans le cas d'une synchronisation. Nous détaillons cela dans la suite de ce document.

### 3.2.2. Etiquetage des graphes

L'information que doit véhiculer le graphe va au-delà de ce que permet de représenter sa topologie et il devient alors nécessaire d'ajouter des données supplémentaires pour caractériser plus finement les objets. Cela est possible en associant des étiquettes aux arêtes et aux sommets du graphe.

#### 3.2.2.1. Etiquetage des sommets

Les tracés des gestes multipoints peuvent avoir différentes formes. Un geste peut être la combinaison de plusieurs segments de droites, de plusieurs cercles, d'un segment de droite et d'un cercle, et plein d'autres formes. Il est donc nécessaire d'identifier la forme des traces pour la reconnaissance du geste. Nous proposons de véhiculer cette information par les étiquettes des sommets de type *Stroke*. Afin de pouvoir identifier les différentes formes des tracés une phase de reconnaissance sur les tracés mono stroke s'impose.

<sup>8</sup> Nous utilisons les termes : primitive, trait et tracé pour désigner le résultat d'interaction d'un doigt au sein du geste.

### **Apprentissage d'étiquettes de sommets**

Cette étape se fait de manière indépendante à note système. Un système statistique<sup>9</sup> est utilisé pour l'apprentissage des étiquettes des tracés composant les gestes. Un ensemble de caractéristiques structurelles est extrait de chaque tracé, et un classifieur statistique est entraîné pour l'apprentissage. Le modèle appris sera ensuite utilisé dans notre système de reconnaissance de gestes multipoints, pour reconnaître les formes des tracés. (Voir Figure 3-2 : partie Apprentissage)

Les sommets modélisant les extrémités ne véhiculent aucune information de forme, les étiquettes portées sont respectivement "Début" pour le début du tracé et "Fin" pour la fin du tracé.

#### **3.2.2.2. Etiquetage des arêtes**

Les arêtes portent l'information du positionnement des tracés du geste. Cette information est mesurée de façon relative entre les différentes primitives. Nous considérons dans notre système deux types de positionnements ; le positionnement temporelle et le positionnement spatial.

##### **Informations temporelles**

Les modèles existants permettant d'exprimer les relations temporelles peuvent être divisés en deux catégories : les modèles basés sur la notion d'intervalle temporel [45] et ceux basés sur la notion de point temporel [46]. Nous distinguons ces deux types d'évènements dans les graphes des gestes manipulés : le tracé dans sa globalité est un intervalle de temps, les extrémités du tracé sont des points temporels.

Le type basé sur la notion d'intervalle temporel, considère les entités élémentaires comme des intervalles de temps qui peuvent être ordonnés selon différentes relations. Les modèles existants sont principalement basés sur les relations définies par Allen dans [45]. Pour notre système, nous considérons un évènement (un tracé du geste) comme un intervalle de temps, qui exprime la durée de vie d'un stroke. Nous exploitons les relations d'Allen, pour le positionnement temporel de deux tracés d'un même geste. Ces relations sont portées par les arêtes reliant les sommets de type *Stroke*. Nous avons mentionné auparavant, que la structure des graphes n'est pas la même pour tous les gestes, certaines informations sont importantes et donc présentes dans tous les graphes, d'autres informations dépendent de la dynamique du geste. L'information de positionnement temporel des tracés est nécessaire pour la reconnaissance du geste, elle est donc présente dans tous les graphes.

---

<sup>9</sup> Les systèmes statistiques de reconnaissance de formes, sont présentés dans : 2.3.1 Approche « statistique » de reconnaissance de symboles

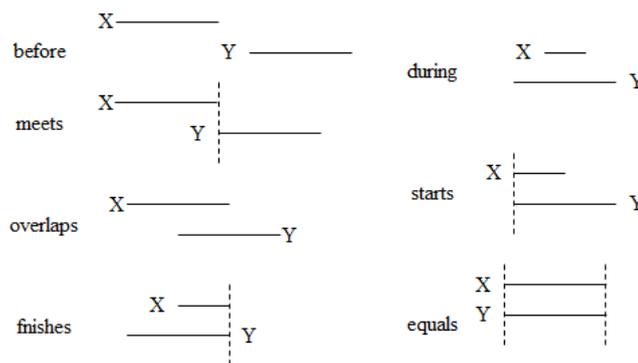


Figure 3-5- Représentation graphique des relations d'Allen pour le positionnement de deux intervalles de temps<sup>10</sup>

Pour le second type qui est basé sur la notion de point temporel, les points sont des unités élémentaires répartis le long de l'axe du temps. Chaque événement est associé à un point temporel. Pour les deux événements  $e_1$  et  $e_2$ , trois relations temporelles peuvent être déterminées. Un événement peut être avant (<), après (>) ou simultané (=) à un deuxième événement. Les extrémités des tracés sont des événements de ce type.

Pour certains types de gestes multipoints, le début et la fin d'un même tracé peuvent être synchronisés temporellement (simultanés), nous prenons exemple du geste "Tap". Pour permettre la reconnaissance de ce type de geste, nous ajoutons une arête entre le début et la fin du même tracé, pour porter l'information de synchronisation temporelle. Cette information est représentée par ce type de relation (l'étiquette "="). L'information de synchronisation temporelle n'est jamais représentée entre les extrémités de deux traces différentes. Ça serait de la redondance, comme cette information est véhiculée par les relations d'Allen définits précédemment.

### Informations spatiales

La majorité des techniques de positionnement décrivent les positions des primitives en termes de directions, d'angles et de distances [47]. Nous exploitons pour notre système la notion de boîte englobantes<sup>11</sup>. Nous positionnons les tracés d'un même geste en positionnant leurs boîtes englobantes. Nous projetons ces dernières sur l'axe des 'x' et sur l'axe des 'y' et nous utilisons les relations d'Allen dans le domaine spatial pour le positionnement des projections. Nous considérons alors deux types de relations spatiales : *les relations en X* et *les relations en Y*. Ces informations sont nécessaires pour discriminer les gestes, elles sont donc exprimées pour tous types de geste.

<sup>10</sup> Illustration prise de [http://interstices.info/encart.jsp?id=c\\_38911&encart=2&size=650,550](http://interstices.info/encart.jsp?id=c_38911&encart=2&size=650,550)

<sup>11</sup> Boîte englobante : Une boîte qui englobe l'objet à décrire et dont ses axes sont alignés avec les axes (X et Y) de l'espace de représentation.

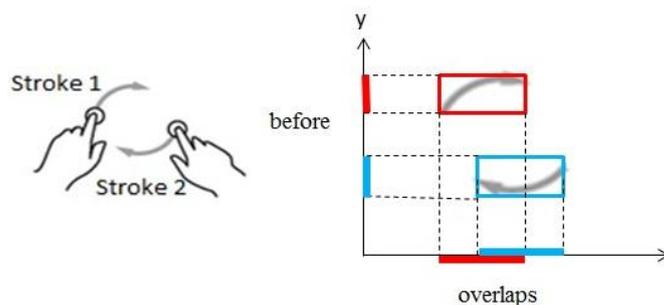


Figure 3-6- Représentation des relations spatiales pour le geste 'rotation' : relation en X "overlaps", relation en Y "before"

Le positionnement spatial entre les extrémités des tracés, est aussi une information importante qui permettra de distinguer entre les objets. Nous exprimons cette information pour les tracés qui commencent ou finissent dans la même zone spatiale, en ajoutant une arête au graphe portant cette information.

### 3.2.2.3. Exemple de représentation par graphes

Nous avons choisi pour notre exemple de représenter le geste 'scroll' à deux doigt sous forme de graphe de relation spatiales et temporelles. (Voir Figure 3-7)

#### 1. Construction du graphe

Le geste est défini par deux traces (voir Figure 3-7), le graphe représentant le geste sera alors composé de 6 sommets : quatre sommets de type *Extrémité* et deux sommets de type *Stroke*. Les arêtes reliant les sommets de type *Extrémité* au sommet de type *Stroke* du même tracé, sont présentes quelque soit le geste considéré. D'autres arêtes toujours présentes, sont celles qui expriment les relations liées aux positionnements spatiaux et temporels des tracés (les arêtes reliant les sommets de type *Stroke* des deux tracés). Pour les arêtes reliant les extrémités qui ne sont présentes que dans le cas d'une synchronisation, une analyse du geste s'impose avant leur création. Les synchronisations observées pour le geste considéré sont :

- Entre le début du tracé 1 et le début du tracé 2: synchronisation spatiale en Y
- Entre la fin du tracé 1 et la fin du tracé 2: synchronisation spatiale en Y
- Entre le début du tracé 1 et la fin du tracé 1: synchronisation spatiale en X
- Entre le début du tracé 2 et la fin du tracé 2: synchronisation spatiale en X
- Entre le début du tracé 1 et le début du tracé 2: synchronisation temporelle
- Entre la fin du tracé 1 et la fin du tracé 2: synchronisation temporelle

Nous ne représentons pas les synchronisations temporelles entre les extrémités de deux traces différentes. En effet, cette information sera représentée par les arêtes reliant les nœuds de types *Stroke*, en positionnant les intervalles de temps des deux tracés par les relations d'Allen [45].

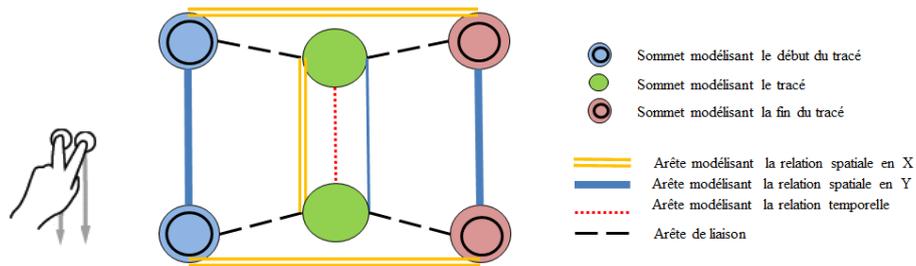


Figure 3-7- Le geste 'scroll' et son graphe avant étiquetage

### 2. Etiquetage du graphe

Les étiquettes des sommets de type *Extrémité* symbolisant le début et la fin du trait sont respectivement "début" et "fin". Les étiquettes des sommets de type *Stroke* sont la forme géométrique du tracé. Dans notre exemple les étiquettes des sommets sont "droite" pour les deux tracés.

Les relations spatiales et temporelles entre les deux traits du geste, sont représentées par les étiquettes des arêtes reliant les deux sommets de type *Stroke* de chaque tracé.

Les relations entre les nœuds des extrémités ne sont représentées que dans le cas d'une synchronisation, spatiale ou temporelle. Celles-ci portent l'étiquette "="

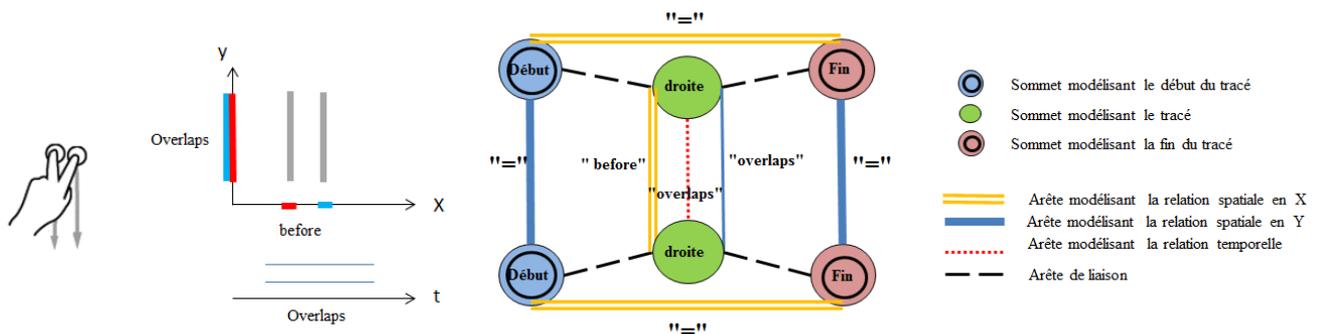


Figure 3-8-représentation du geste "scroll" par graphe de relations spatiales et temporelles

## 3.3. Projection des graphes

Le principe est de repositionner les graphes dans un espace vectoriel et permettre ainsi l'utilisation d'outils statistiques éprouvés.

### 3.3.1.1. Idée directrice

Appliqué au contexte de reconnaissance de formes, le graphe permet une représentation en sous-parties et une description des relations entre ces sous-parties grâce aux deux informations véhiculées au sein de la même structure :

- l'information topologique, portée par les arêtes, qui définit les relations structurelles entre les parties de formes représentées par les sommets.
- l'information étiquetée qui peut correspondre soit aux caractéristiques de la forme (ou d'une partie de forme) si elle est portée par les étiquettes de sommets, soit à exprimer la relation entre deux sommets quand elle est véhiculée par les étiquettes d'arêtes.

Ces deux informations sont complémentaires, et permettent une caractérisation complète du graphe. Il est donc nécessaire de les exploiter pour la discrimination des gestes que nous manipulons.

N. Sidère [44] propose une approche pour la projection de graphe en considérant ces deux informations. Le principe est de chercher des structures topologiques dans le graphe à projeter et de compter leur nombre d'occurrence. Ces structures sont issues du réseau des sous-graphes non isomorphes présenté par [48] (voir Figure 3-9). Ce réseau présente l'ensemble des graphes composés de 'n'arêtes jusqu'au rang de taille N (N étant le nombre d'arcs maximum) de manière totalement exhaustive. Ce réseau est construit de façon itérative à partir du graphe constitué d'un unique sommet. À chaque itération, il est possible de construire un graphe de taille 'n' en ajoutant une arête à un graphe de taille 'n-1' avec au besoin l'ajout d'un sommet supplémentaire. La Figure 3-9 illustre le réseau des graphes non-isomorphes jusqu'au rang de taille 4. Les flèches en pointillé indiquent les chemins de construction du réseau.

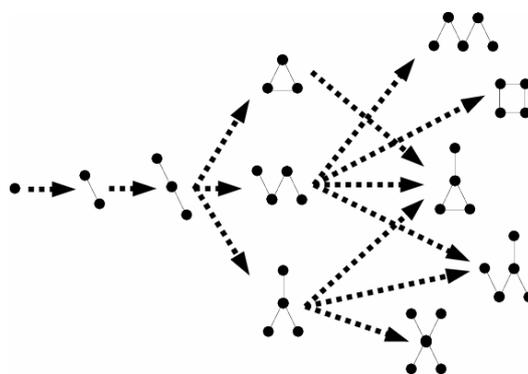


Figure 3-9-Réseau des sous-graphes non isomorphes [44]

La projection de graphe s'appuie sur le dénombrement, au sein du graphe à représenter, de sous-graphes, appelés motifs. L'ensemble de ces sous-graphes est appelé lexique. Le lexique utilisé ici est extrait du réseau des sous-graphes non isomorphes. Le résultat de la projection est une matrice de fréquence d'apparition d'étiquettes de sommets et d'arêtes, dans chaque motif.

Le format de graphes que nous avons proposé fait la distinction entre deux types de sommets (un type modélisant le tracé, et un autre type modélisant les extrémités des tracés), pour ne pas perdre cette information, nous proposons d'étendre cette approche en définissant un lexique de sous-graphes en considérant les types des sommets.

Le deuxième point que nous proposons d'étendre, concerne le processeur d'extraction de motifs : l'approche de Sidère [44] construit le lexique en extrayant des sous-structures de 2 nœuds, 3 nœuds etc, à partir du réseau de sous graphe non-isomorphes. Ces motifs sont définis indépendamment du contexte pour lequel ils sont utilisés. Pour notre approche nous proposons de construire le lexique en extrayant les motifs directement de l'ensemble des graphes manipulés. Ce qui permet un lien plus fort avec le type de graphe. Notre approche s'inspire des méthodes de fouille de graphes.

### 3.3.1.2. Construction du lexique

Nous procédons par décomposition de tous les graphes de la base d'apprentissage : Nous extrayons de chaque graphe tous les sous graphes possibles. Les plus fréquents parmi tous ces sous graphes, seront les motifs que nous utiliserons pour la projection. De cette manière nous aurons gardé dans chaque motif la double information lié à la structure et au type de nœuds.

### 3.3.1.3. Projection

La projection repose sur la mise en place d'une matrice numérique. Les colonnes de la matrice correspondent aux motifs du lexique défini tandis que les lignes sont dédiées à la description de l'information liée aux étiquettes. Un élément de cette matrice donne alors le nombre d'occurrences du motif portant l'étiquetage spécifié sur la ligne correspondante. La Figure 3-10 illustre la structure de la matrice ainsi obtenue.

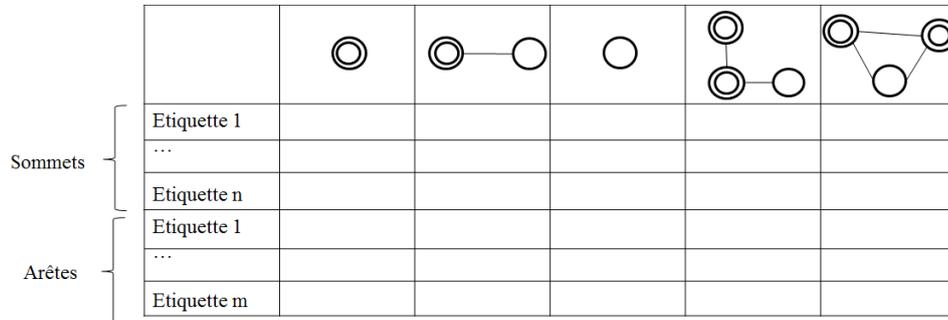


Figure 3-10- Exemple de projection d'un graphe, avec un lexique de taille 5 et un nombre d'étiquettes de taille "n" pour les sommets et de taille "m" pour les arêtes. L'approche de Sidère [44] considère d'autres motifs

## 3.4. Phase d'apprentissage

Après construction et étiquetage des graphes représentant les gestes de la base d'apprentissage, le lexique pour la projection, est construit à partir de tous ces graphes. La phase d'apprentissage est illustrée dans la Figure 3-11. La projection permet de passer d'une représentation des gestes sous forme de graphe, à leur représentation sous forme de vecteur numérique. La phase d'apprentissage consiste à entraîner des classifieurs, pour l'apprentissage de toutes les classes des gestes multipoints. Nous pouvons ainsi comparer les performances de plusieurs classifieurs pour l'apprentissage des gestes considérés.

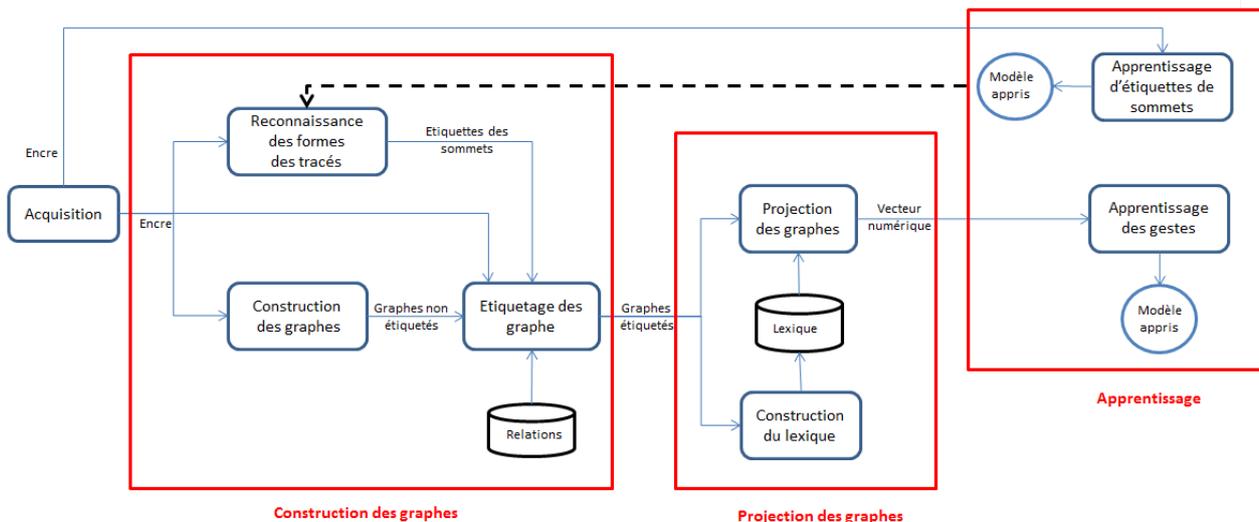


Figure 3-11 Processus d'apprentissage

### 3.5. Phase de reconnaissance

Pour reconnaître un nouveau geste (Voir Figure 3-12), nous procédons de la même manière que pour l'apprentissage : Nous commençons par construire et étiqueter son graphe, puis nous projetons le graphe du geste en utilisant le lexique défini en phase d'apprentissage, le modèle appris servira à déterminer la classe de ce geste.

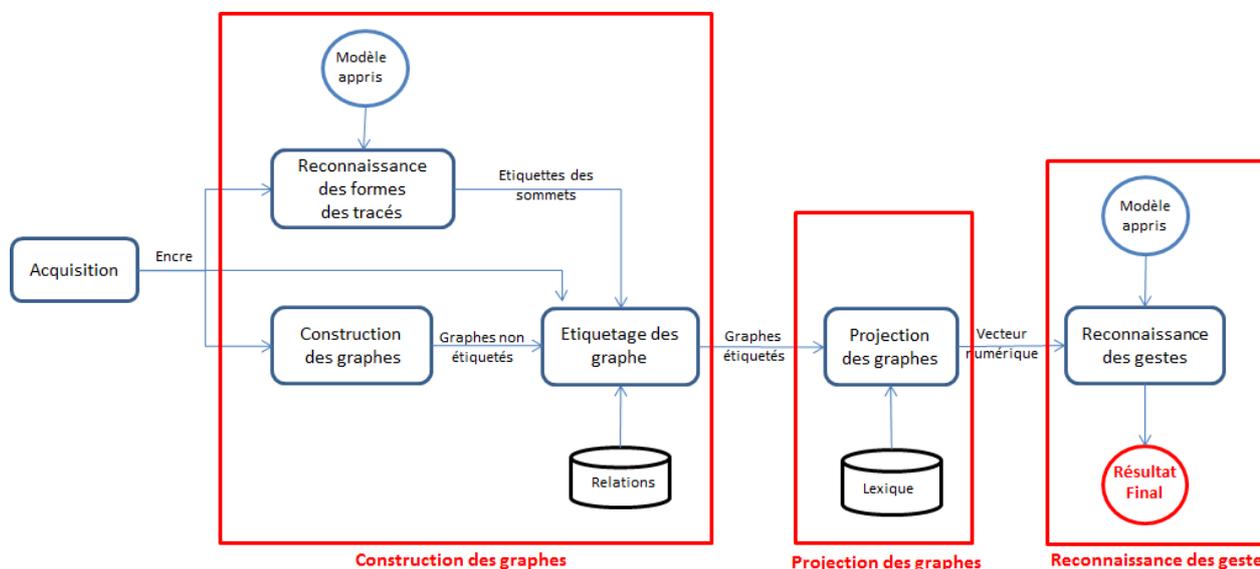


Figure 3-12 Processus de reconnaissance

## Conclusion

Nous avons présenté dans ce chapitre le système de reconnaissance de gestes multipoints que nous avons mis en place. Nous représentons les gestes sous forme de graphes de relations spatiales et temporelles. Afin d'éviter la complexité calculatoire liée à la comparaison des graphes, et pour profiter pleinement des performances des classifieurs statistiques, nous avons proposé une approche qui se base sur les méthodes de sondage de graphes. Nous nous sommes inspirés dans notre approche d'une méthode de projection qui recherche des motifs de sous-graphes dans le graphe à projeter [44].

Ces motifs ne véhiculent que l'information topologique, et sont construits de manière indépendante au système. Nous avons proposé d'améliorer cette méthode en recherchant les motifs les plus informatifs (portant des informations liées au type de nœud considérés), et extraits directement de l'ensemble des graphes manipulés. Ces motifs sont les sous-graphes les plus fréquents dans la base de graphes manipulée.

## 4. Expérimentations

### 4.1. Objectifs

Le graphe par sa nature, véhicule deux types d'informations : l'information topologique qui décrit la structure du graphe, et l'information liée à l'étiquetage, qui correspond aux informations portés par les sommets et les arêtes du graphe. Le système que nous avons proposé utilise ces deux informations véhiculées par la matrice de projection extraite à partir de chaque graphe d'un geste. Les matrices de projection des gestes, contiennent le nombre d'occurrence de motifs présélectionnés. Ces motifs sont des sous-structures de graphes. Les relations spatiales et temporelles entre les tracés au sein d'un même geste, sont exprimées dans le graphe du geste, par des étiquettes. Les matrices de projections des gestes véhiculent également cette information en comptant le nombre d'occurrence d'étiquettes présentes dans le graphe du geste pour chaque motif. La présence ou non d'une étiquettes dans le graphe d'un geste permet sa caractérisation. Nous souhaitons dans un premier temps évaluer à travers nos expérimentations, l'influence de ces deux informations (l'information topologique et l'information liée à l'étiquetage), pour la reconnaissance des gestes.

Pour nos premières expérimentations, nous nous arrêtons à ces deux évaluations. Pour une évaluation complète du système, nous pourrions envisager de tester l'importance de l'utilisation des graphes en comparant plusieurs méthodes de représentation des gestes. Il serait intéressant aussi de mesurer l'efficacité du format de graphe que nous avons mis en place, en comparant avec les méthodes de représentation par graphe de l'état de l'art.

### 4.2. Base de gestes

Pour tester notre système, nous avons sélectionné un ensemble de dix gestes multipoints que nous illustrons dans la Figure 4-1.

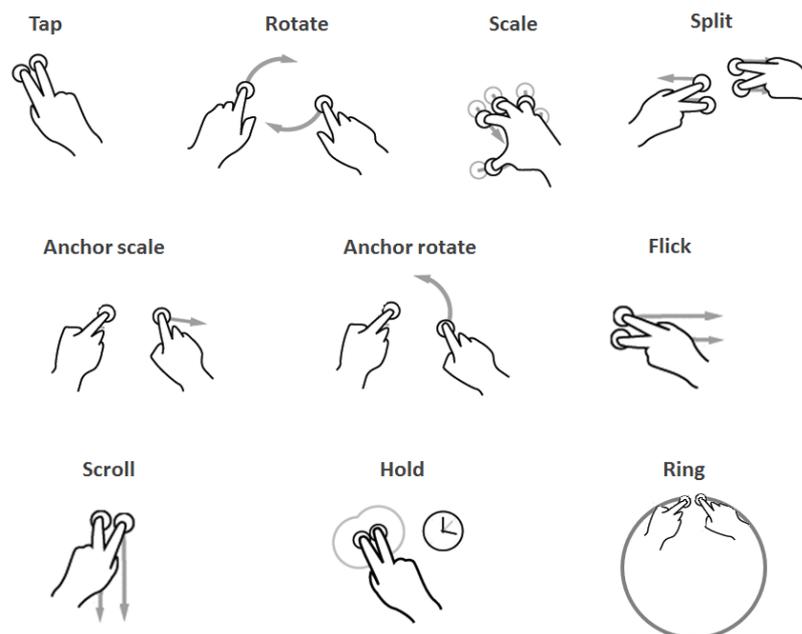


Figure 4-1 L'ensemble des gestes multipoints pour les expérimentations

La plupart de ces gestes sont issus du domaine de l'interaction multipoint, nous avons sélectionné quelques-uns de "Open Source Gesture Library"<sup>12</sup> et nous avons ajouté un geste permettant de dessiner une forme géométrique (Ring). Notre base de gestes contient 3 types de gestes : des gestes à deux traces, à 4 traces et à 5 traces. Les gestes que nous manipulons ne sont associés à aucune commande, nous considérons uniquement l'encre électronique de ces gestes, qui nous permettra de distinguer chaque classe.

Pour la collecte des données, nous avons utilisé l'interface de saisie multipoint de l'équipe *INTUDOC* (voir Figure 4-2), les fichiers de stockage sont au format standard *InkML*<sup>13</sup>. Quatre personnes de l'équipe *INTUDOC* ont participé à la collecte de données, ces personnes sont des habitués des interfaces tactiles. Nous avons demandé à chacun d'eux de saisir 15 gestes par classe en utilisant les doigts. Nous laissons la liberté quant à la taille des gestes saisis.

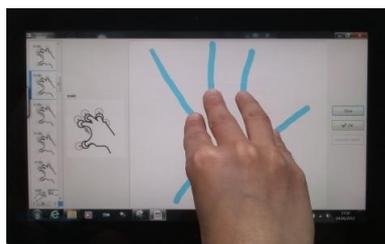


Figure 4-2 Saisie des gestes sur interface tactile multipoint : geste "Scale"

### 4.3. Protocole

Nous représentons les gestes à reconnaître sous forme de graphes de relations spatiales et temporelles, les sommets du graphe modélisent les tracés, les arêtes représentent les relations entre les tracés. Dans les premiers tests, nous ne considérons pas la partie apprentissage d'étiquettes de sommets de type *stroke* (voir Apprentissage d'étiquettes de sommets). Les étiquettes véhiculées par les sommets du graphe sont donc uniquement "début" "stroke" et "fin", pour les nœuds modélisant respectivement, l'extrémité initiale du tracé, le tracé, et son extrémité finale. Les étiquettes portées par les arêtes sont les relations d'Allen [45] (voir Etiquetage des arêtes).

Chaque graphe est transformé sous forme de matrice numérique, cette matrice contient le nombre d'occurrence des motifs du lexique que nous construisons (voir Projection des graphes). La sélection des motifs fréquents se base sur un seuil défini empiriquement. Nous précisons le nombre de motifs fréquents avant l'exécution du système. Cependant, cette étape du processus pourrait être automatisée en sélectionnant un nombre de motifs de référence en fonction de la taille de la base de graphe manipulé.

Pour évaluer l'importance des deux informations : topologique, et celle lié à l'étiquetage, nous mettons en place les deux expériences décrites dans les paragraphes suivants.

---

<sup>12</sup> <http://gestureworks.com/features/open-source-gestures/> : Bibliothèque de gestes multipoints défini par *Ideum* [27]

<sup>13</sup> <http://www.w3.org/TR/InkML/> *Inkml* est un langage de balisage, c'est un sous type de fichier *xml*

### 4.3.1. Evaluation de l'importance de l'information topologique

Dans cette première expérience nous projetons les graphes de la base en comptant le nombre d'occurrence des motifs sans considérer ni les étiquettes des nœuds ni celles des arêtes, nous obtenons un vecteur d'occurrences au lieu d'une matrice :

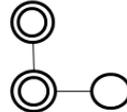
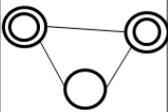
					
Topologie	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$

Figure 4-3 Projection de l'information topologique d'un graphe

Cette expérience permettra d'étudier l'influence de l'information topologique portée par la structure du graphe, sur la reconnaissance des gestes. la présence ou non d'un motif, ainsi que sa fréquence d'apparition permet de discriminer sa classe. Nous faisons également varier le nombre de motifs à chaque expérience, pour observer l'évolution du système.

### 4.3.2. Evaluation de l'importance de l'information liée à l'étiquetage

Nous considérons ici les fréquences des étiquettes d'arêtes et de sommets pour la projection (les lignes de la matrice). Nous projetons en gardant les mêmes motifs de l'expérience précédente, afin de pouvoir comparer les résultats de reconnaissance. Cette expérience nous permettra d'évaluer l'influence des étiquettes modélisant les relations spatiales et temporelles entre les tracés d'un geste. Nous faisons également varier le nombre de motifs à chaque expérience

	Motif 1	Motif 2	....	Motif n
Etiquette 1	$F_{11}$	$F_{12}$	....	$F_{1n}$
Etiquette 2	$F_{21}$	$F_{22}$	....	$F_{2n}$
...	....	....	....	....
Etiquette m	$F_{m1}$	$F_{m2}$	....	$F_{mn}$

Figure 4-4 Projection de l'information topologique, et celle lié à l'étiquetage d'un graphe

### 4.3.3. Découpage de l'ensemble des données

Nous divisons la base des gestes que nous avons collectés en deux parties ; apprentissage et test. La première partie des données servira pour définir les motifs de références et entrainer le classifieur, le modèle appris sera ensuite utilisé pour reconnaître les gestes de la seconde partie. Il existe trois manières classiques de découper une base de données:

- *Découpage en Multi scripteur* : Plusieurs personnes participent à la construction de l'ensemble d'apprentissage, ces mêmes scripteurs participent également dans la construction de l'ensemble de test.
- *Découpage en Omni-Scripteur* : Le système est entraîné sur un ensemble de gestes collectés par des personnes, et testé sur des gestes collectés par d'autres personnes. Le découpage en Omni-Scripteur est le plus proche d'un cas réel d'utilisation. En effet, un système de

reconnaissance de forme doit être capable de reconnaître les saisies de n'importe quel scripteur.

- *Découpage en Mono-Scripteur* : le système est entraîné et testé sur les données saisies par le même scripteur. Cette manière de découper une base de données est la plus basique, et donne en générale les meilleurs résultats de reconnaissance, en effet, quand un système est entraîné sur les symboles saisies pas une personne, il est plus simple pour le système de reconnaître d'autres symboles saisies par cette même personne.

Pour nos premières expérimentations, nous faisons les tests en mono-scripteur, ce qui nous permettra d'avoir les premiers résultats que nous améliorerons par la suite. Nous divisons la base de gestes que nous avons collectés en 60% pour le fichier d'apprentissage, et 40% pour le fichier de teste.

### 4.4. Expériences

Nous avons implémenté notre système en langage C++ et nous avons utilisé la bibliothèque *LEDA*<sup>14</sup> pour la création et la manipulation de graphes, le choix de cette bibliothèque porte sur la richesse de ces fonctionnalités et sur sa simplicité d'utilisation. Pour la manipulation des fichiers *inkml*, nous avons utilisé la bibliothèque *Xerces*<sup>15</sup>. Nous utilisons le logiciel *Weka*<sup>16</sup> pour l'apprentissage et la classification, ceci pour la richesse de sa bibliothèque de calcul. Nous testerons plusieurs classifieurs ce qui nous permettra de comparer les différentes performances.

## Conclusion

Nous avons présenté dans cette section le protocole qui va nous permettre de tester notre système de reconnaissance de gestes multipoints, nous ne disposons pas de résultats à interpréter dans ce rapport. Notre système est presque prêt à être testé. Nous discutons les résultats que nous obtiendrons dans les jours à suivre, lors de la soutenance du stage.

---

<sup>14</sup> <http://www.algorithmic-solutions.com/leda/>

<sup>15</sup> <http://xerces.apache.org/xerces-c/>

<sup>16</sup> <http://www.cs.waikato.ac.nz/ml/weka/>

## Conclusion générale

Durant notre stage, nous nous sommes intéressés à la problématique de reconnaissance de geste graphique sur surfaces sensibles multipoints. Ces interfaces sont actuellement beaucoup utilisées dans le domaine de l'interaction homme machine, et permettent la reconnaissance de quelques gestes basiques, comme par exemple agrandir une image en utilisant deux doigts ou encore passer d'une page à l'autre en glissant le doigt sur la surface de l'écran. Ces systèmes se basent sur une définition complète des gestes considérés dans le système, et un reconnaiseur dédié à chaque geste, ce qui les rend difficilement adaptable à tous types de gestes.

Le but de notre travail est l'élaboration d'un système qui soit capable d'apprendre à partir des gestes saisis, et de les reconnaître par la suite. Nous avons étudié les différentes approches de reconnaissance de formes monopoints, de l'état de l'art. En effet le domaine est bien avancé sur ce type de surface. Le but de cette étude était de pouvoir utiliser l'une de ces approches pour la reconnaissance de geste multipoints. Le geste multipoint est beaucoup plus riche qu'un symbole monopoint, celui-ci est défini par la synchronisation des différents tracés (contact sur la surface) et à leur positionnement dans l'espace temporel. Parmi les approches de reconnaissance de formes de l'état de l'art, nous avons dégagé l'approche de reconnaissance structurelle à base de graphes, qui permet de représenter les sous parties du geste ainsi que les relations entre ses parties. Nous représentons les gestes multipoints par un graphe de relations spatiales et temporelles. Ces relations permettent de positionner les différents traits du geste. Nous avons utilisé les relations définies par Allen [45]

La comparaison des graphes dans le domaine de la reconnaissance de forme est un problème complexe. Plusieurs méthodes permettant de soulever cette complexité existent dans la littérature, parmi ces méthodes, la projection de graphe est une alternative intéressante qui cherche à repositionner les graphes dans un espace vectoriel, et permet ainsi l'utilisation de classificateur statistique. Les outils statistiques fournissent en générale les meilleurs résultats. Le système que nous avons proposé se base sur une approche de projection de graphe.

Nous avons proposé d'étendre une méthode de l'état de l'art [44] qui s'appuie sur le dénombrement, au sein du graphe à projeter, de sous-graphes, appelés motifs. L'ensemble des motifs est appelé lexique. Le lexique utilisé dans cette approche est défini de manière indépendante au système. Pour notre approche, nous avons proposé d'extraire les motifs du lexique directement de la base de graphe utilisée, ce qui permet un lien plus fort avec le type des graphes manipulés. Le lexique que nous définissons est l'ensemble des sous graphes fréquents de la base d'apprentissage.

Pour l'évaluation de notre système nous avons mis en place deux expériences : la première expérience permettant de mesurer l'importance de l'information topologique véhiculée par la structure du graphe. La deuxième expérience permet de mesurer l'influence des informations de positionnement spatial et temporel des tracés. Nous ne disposons pas de résultats à discuter dans ce rapport. Les résultats des deux expériences envisagées seront interprétés lors de la soutenance du stage.

Dans notre système, Les motifs fréquents sélectionnés pour la construction du lexique sont définis empiriquement. Pour la continuité de notre travail, il serait intéressant d'utiliser les approches de fouille de graphe pour la sélection des motifs fréquents.

# Bibliographie

1. A. Bulis, R. Almog, M. Gerner and U. Shimony, "Computerized Recognition of Hand-Written Musical Notes". *Proceedings of International Computer Music Conference*. p110-112 (1992)
2. O. Yadid-Pecht, M. Gerner, L Dvir, E Brutman and U. Shimony, "Recognition of handwritten musical notes by a modified Neocognitron". *Machine Vision and Applications* 9 (2). p 65-72 (1996)
3. Ahmad Montaser AWAL. "Reconnaissance de structures bidimensionnelles : Application aux expressions mathématiques manuscrites en-ligne". *Thèse de doctorat. Université de Nantes, (2010)*
4. Sébastien Macé. Composition manuscrite interactive et interprétation à la volée de documents structurés en-ligne. *thèse de doctorat de l'Institut Nationale des Sciences Appliqués de Rennes, (2007)*
5. Christian Heide Damm, Klaus Marius Hansen, and Michael Thomsen. "Tool support for cooperative object-oriented design: gesture based modelling on an electronic whiteboard". *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '00)*, p 518–525, (2000)
6. Edward Lank. "A retargetable framework for interactive diagram recognition". *Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR'03)*, p 185–189, (2003).
7. James A. Landay and Brad A. Myers. "Interactive sketching for the early stages of user interface design". *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI'95)*, p 43–50, (1995).
8. E. Ng, T. Bell and A. Cockburn, "Improvements to a Pen-Based Musical Input System", *Proceeding of The Australian Conference on Computer-Human Interaction (OzCHI'98)*, p.239-252 (1998)
9. Jamie Anstice, Tim Bell, Andy Cockburn, and Martin Setchell. "The design of a penbased musical input system". *Proceedings of the Sixth Australian Conference on Computer-Human Interaction (OzChi 96)*, p 260–267, (1996).
10. Chris Calhoun, Thomas F. Stahovich, Tolga Kurtoglu, and Levent Burak Kara. "Recognizing multi-stroke symbols". *Proceedings of the AAAI Spring Symposium on Sketch Understanding (SSSU'02)*, p 15–23, (2002).
11. Tolga Kurtoglu and Thomas F. Stahovich. "Interpreting schematic sketches using physical reasoning". *Proceedings of the AAAI Spring Symposium on Sketch Understanding (SSSU'02)*, p 78–85 (2002).

12. Manuel J. Fonseca, César Pimentel, and Joaquim A. Jorge. "Cali: An online scribble recognizer for calligraphic interfaces". *Proceedings of the AAAI Spring Symposium on Sketch Understanding (SSSU'02)*, p 51–58 (2002).
13. Patrick Chiu, Chunyuan Liao, and Francine Chen "Multi-touch Document Folding: Gesture Models, Fold Directions and Symmetries". *Proceedings of the 2011 annual conference on Human factors in computing systems*. p 1591-1600 (2011)
14. Mike Wu, Chia Shen, Kathy Ryall, Clifton Forlines, and Ravin Balakrishnan "Gesture Registration, Relaxation, and Reuse for Multi-Point Direct-Touch Surfaces", *Horizontal Interactive Human-Computer Systems. First IEEE International Workshop*. P 8, (2006)
15. Changbo Wang, Jinqiu Dai, Long Wang "3D Multi-touch Recognition Based Virtual Interaction". *3rd International Congress on Image and Signal Processing (CISP2010)*,p 1478-1481, (2010)
16. Mathias Frisch, Jens Heydekorn, Raimund Dachsel "Investigating Multi-Touch and Pen Gestures for Diagram Editing on Interactive Surfaces", *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, p 634-640 (2009)
17. Sashikanth Damaraju, Andruid Kerne "Multitouch Gesture Learning and Recognition System", *IEEE Workshop on Tabletops and Interactive Surfaces*, p 102-104 (2008)
18. Wang, F, Cao, X., Ren, X., Irani, P.: "Detecting and leveraging finger orientation for interaction with direct-touch surfaces". *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology*, p. 23–32. (2009)
19. Marquardt, N., Kiemer, J., Greenberg, S: "What caused that touch? Expressive interaction with a surface through fiduciary-tagged gloves". *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces ITS 2010*, p. 139–142. (2010)
20. Chi Tai Dang and Elisabeth André "Usage and Recognition of Finger Orientation for Multi-Touch Tabletop Interaction", *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction*, I p 409-426 (2011)
21. T. Deselaers, A. Criminisi, J. Winn, A. Agarwal "Incorporating On-demand Stereo for Real Time Recognition", *Computer Vision and Pattern Recognition*, p 1-8 (2007)
22. Elias, J.G, Westerman, W.C, Haggerty M.M. "Multi-Touch Gesture Dictionary" US Patent, 2007.
23. Hoste L. "Software engineering abstractions for the multi-touch revolution". *Proceeding ICSE*, p 509-510 (2010)
24. Kin K, Hartmann B, DeRose T, Agrawala M Proton: "Multitouch Gestures as Regular Expressions". *Proceeding CHI*, (2012).

25. Scholliers, C., Hoste, L., Signer, B., and De Meuter, W. "Midas: a declarative multi-touch interaction framework". *Proceeding TEI*, p 49-56 (2011)
26. Hao lu, Yang Li "Gesture Coder: A Tool for Programming Multi-Touch Gestures by Demonstration". *Proceeding CHI*, (2012)
27. <http://www.gestureml.org/> [Online]
28. Don Willems, Ralph Niels, Marcelvan Gerven, Louis Vuurpijl "Iconic and multi-stroke gesture recognition". *Pattern Recognition*, vol 42, p 3303-3312, (2009)
29. Miyao, H., Maruyama, M., "An Online Handwritten Music Score Recognition System", *Proceedings of the 17th International Conference on Pattern Recognition*, p. 461-464, (2004).
30. Kam-Fai Chan and Dit-Yan Yeung, "An efficient syntactic approach to structural analysis of on-line handwritten mathematical expressions," *Pattern recognition*, vol. 33, p.375-384, (2000).
31. Bo Yu and Shijie Cai. "A domain-independent system for sketch recognition". *Proceedings of the First international conference on Computer graphics and interactive techniques (GRAPHITE'03)*, p 141-146, (2003).
32. Tefvik Metin Sezgin, Thomas Stahovich, and Randall Davis. "Sketch based interfaces: Early processing for sketch understanding". *Proceedings of International Conference on Computer Graphics and Interactive Techniques (ICCGIT'01)*, p 37-44, (2001).
33. Shi-Zhong Liao, Wen-Gang Liu, and Wei Guo. "Composite sketch shape recognition based on dagsvm and decision tree". *Proceedings of Fifth International Conference on Machine Learning and Cybernetics (ICMLC'06)*, p 3254-3259, (2006).
34. Utpal Garain and Bidyut Baran Chaudhuri. "Recognition of online handwritten mathematical expressions". *IEEE Transaction on Systems, Man and Cybernetics - Part. B: Cybernetics*, 34(6), p 2366-2376, (2004).
35. Sébastien Macé, Eric Anquetil, "Eager Interpretation of On-Line Hand-Drawn Structured Documents: The DALI Methodology", *Pattern Recognition*, vol 42, P 3202-3214, (2009).
36. Liu CL, Jaeger S, Nakagawa M. "Online recognition of chinese characters: The state-of-the-art". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. (2004)
37. Ramel, J.-Y. "Graphes en analyse d'images : un problème fortement combinatoire". *Séminaire dans le cadre du groupe "Architecture et Images" du GDR ISIS*. (2009)

38. Xiaogang Xu, Zhengxing Sun, Binbin Peng, Wangyu Jin, and Wenyin Liu.s., "An online composite graphics recognition approach based on matching of spatial relation graph," *International Journal on Document Analysis and Recognition*, vol. 7, p 44–55, (2004).
39. Chung Fan R. K. "Spectral Graph Theory". *American Mathematical Society*. (1997)
40. Muhammad Muzzamil Luqman "Fuzzy Multilevel Graph Embedding for Recognition, Indexing and Retrieval of Graphic Document Images" *Thèse de doctorat. Université François Rabelais, Tours (2012)*.
41. Kaspar Riesen, Host Bunke "Graph classification and clustering based on vector space embeddin" *World scientific, London p 331(2010)*
42. Papadopoulos A. N, Manolopoulos Y. "Structure-based similarity search with graph histograms". *In International Workshop on Database and Expert Systems Applications*. (1999).
43. Lopresti D, Wilfong G. "A fast technique for comparing graph representations with applications to performance evaluation". *International Journal of Document Analysis and Recognition*. (2003)
44. Nicolas Sidère, Pierre Héroux, Jean-Yves Ramel "Vector Representation of Graphs: Application to the Classification of Symbols and Letters" *10th International Conference on Document Analysis and Recognition, ICDAR '09. P 681-685, (2009)*
45. Allen, F J. "Maintaining Knowledge about Temporal Intervals." *ACM, Tome 26 (11), p 832 – 843 (1983)*
46. Vilain M., and Kautz H. A. "Constraint propagation algorithms for temporal reasoning." *AAAI-86, p. 132-144 (1986)*
47. Qureshi, Rashid Jalal, and Jean-Yves Ramel. "Combination of symbolic and statistical features for symbols recognition." *IEEE-ICSCN, p 477-482 (2007)*
48. Jaromczyk, J, and G Toussaint. "Relative neighborhood graphs and their relatives." *Proceedings of the IEEE, (1992)*
49. Bill Buxton. "Multi-Touch Systems that I Have Known and Loved". [Online]. <http://www.billbuxton.com/multitouchOverview.html>