Un éditeur interactif de tableaux dessinés à main levée

An Interactive Editor for Hand–Sketched Tables

par Luc JULIA, Claudie FAURE
ENST Dpt Signal — CNRS URA 820, 46, rue Barrault, F-75634 Paris cedex 13

Résumé

Dans le cadre des interfaces multimodales d'aide à la conception incrémentale de documents graphiques, un prototype (TAPAGE) a été développé pour l'édition interactive de tableaux par la parole et le geste. Cet article présente les traitements qui permettent d'obtenir un tableau normalisé à partir d'un tracé à main levée et de le modifier par des commandes gestuelles. Les traitements et les représentations des données sont élaborés en fonction des différentes contraintes qu'imposent la qualité variable des tracés manuscrits, la situation d'interaction utilisateur–machine et l'objectif de coopération entre le dessinateur et le système d'interprétation des tracés.

Mots clés : Reconnaissance de tracés à main levée, Interaction au stylo, Structures spatiales.

Abstract

The studies on the incremental graphic design using multimodal interfaces led us to realize an initial prototype : a speech end gestures table editor, TAPAGE. This paper presents the beautification of hand–sketched tables and the correction with gestural commands. Data processing and data structures were chosen according to several constraints : the variable quality of pen entries, the human–computer interaction context and the possible collaboration between the user and the system.

Key words : *Hand–sketched stroke recognition, Pen–based interfaces, Spatial structures.*

1. Introduction

Le travail présenté concerne essentiellement le noyau applicatif d'un prototype d'interface multimodale pour la conception incrémentale de tableaux par la parole et le geste. L'architecture et les protocoles d'interaction multimodaux du prototype TAPAGE ont été décrit dans : [Faure 1993, 1994]. On insistera donc plus particulièrement sur la reconnaissance et l'interprétation de formes produites à l'aide d'un stylo sur une surface qui permet la numérisation du tracé.

La reconnaissance de tracés graphiques saisis sur des tablettes à numériser a fait l'objet de nombreuses études, en particulier pour la reconnaissance de l'écriture, qu'il s'agisse de reconnaître des caractères isolés ou de l'écriture cursive. On trouvera dans [Tappert 1990] un état de l'art sur la question. La reconnaissance du dessin concerne des domaines comme les schémas électriques ou les diagrammes de type organigrammes ou réseaux de Pétri [Murase 1986; Machii 1993; Zhao 1993a]. L'arrivée des ordinateurs à stylo ouvre une nouvelle voie de recherche qui aborde la reconnaissance des tracés graphiques dans un contexte d'interaction homme–machine [Rubine 1990; Apte 1993; Endo 1994; Nakagawa 1993b; Stolpmann 1993; Zhao 1993b]. Notre étude se rattache directement à ce type de démarche.

Les ordinateurs à stylo ouvrent de nouvelles perspectives pour les protocoles d'interaction utilisateur–machine. Ils permettent de recréer la situation papier–stylo où l'utilisateur exprime rapidement ses idées sous forme visuelle. L'interaction directe au stylo évite l'apprentissage de langages ou de logiciels graphiques mais surtout libère l'utilisateur qui n'a plus à recourir à un outil d'expression intermédiaire qui rompt l'intégration naturelle de la gestuelle graphique et de la pensée. Le succès de ce type d'interaction implique que la saisie directe sur papier électronique soit augmentée de procédures de traitements qui interprètent la version encre, c'est–à–dire le brouillon produit au stylo, et reconstruisent une version « normalisée »[1] qui servira à sa publication, son archivage ou sa diffusion à d'autres utilisateurs dans le cadre d'une conception collective du document. On parle alors de coopération entre dessinateur et machine qui travaillent sur le même document afin d'atteindre la version finale désirée.

La conception de documents graphiques dans le contexte d'une coopération utilisateur–machine impose des contraintes sur les

[1] Nous adoptons ici « normalisation » pour traduire *beautification*, le terme consacré en anglais pour identifier le traitement qui permet de passer d'une version à main levée à une version « propre ».

traitements. On décrira d'abord la nature de ces contraintes. La présentation de l'interprétation automatique des tableaux fera apparaître les choix qui ont été faits au niveau des procédures et des représentations pour permettre une très grande rapidité d'interprétation et un partage des tâches de correction entre utilisateur et machine au cours de la conception incrémentale des tableaux.

2. Interprétation dans un contexte contraint

Pour comprendre les problèmes rencontrés pour la réalisation d'un éditeur de tableaux interactif, nous donnons le scénario d'une interaction entre un utilisateur et la machine : un tableau est produit au stylo sur l'écran de l'ordinateur et immédiatement interprété par la machine, la version normalisée est affichée sur ce même écran. Cette version peut ne pas satisfaire l'utilisateur, soit parce qu'une erreur d'interprétation a été faite, soit parce que l'utilisateur a produit un tableau qui ne correspond pas exactement à ses intentions qui évoluent au cours de la conception incrémentale. L'utilisateur a alors la possibilité de corriger son tableau à l'aide d'outils disponibles au niveau de l'interface, les modifications apportées sont interprétées et le tableau mis à jour est affiché instantanément. Une fois la structure satisfaisante, le remplissage et l'impression du tableau peuvent être effectués.

L'interprétation fait intervenir des procédures de traitement et de reconnaissance des données saisies en ligne ainsi que des connaissances liées au domaine d'application. Dans un contexte de coopération homme–machine, l'interprétation de dessins est soumise à plusieurs contraintes.

La première contrainte est celle du style de production dont la qualité est très variable d'un dessinateur à l'autre et à l'intérieur d'un même dessin. Cette qualité semble liée à la vitesse d'exécution du dessin. Quand cela est possible, les seuils de tolérance qui sont forcément impliqués dans les décisions de l'interpréteur, seront adaptés à la qualité du dessin.

La seconde contrainte est caractéristique d'une situation d'interaction utilisateur–machine, c'est la contrainte de temps : que ce soit lors de la création, ou lors des modifications, l'interprétation doit être immédiate, les structures de représentation et les calculs doivent être optimisés. La négociation entre la qualité et le temps de l'interprétation devra être menée au plus juste, avec un impératif : l'utilisateur ne doit pas attendre plus d'une seconde un résultat.

A cela il faut ajouter la contrainte de coopération qui impose un partage des tâches entre machine et dessinateur lors de la modification du dessin. La machine doit propager automatiquement des modifications locales dans la structure du tableau et effectuer les modifications qui sont suggérées par celles de l'utilisateur pour maintenir une cohérence globale du tableau.

Une dernière contrainte, d'avantage liée à l'interface qu'au noyau applicatif, est à signaler. Le stylo, utilisé à la fois pour produire les données (les dessins et le texte) et pour effectuer des commandes gestuelles, fait apparaître l'ambiguïté des tracés graphiques propre aux interactions au stylo. Cette ambiguïté est gérée dans ce prototype au niveau de l'interface : l'utilisateur informe le système du statut des tracés graphiques qui vont être produits. Cette solution nous semble plus fiable qu'une discrimination automatique des commandes et des données dans la mesure où le tracé ne contient pas souvent l'information discriminante (un *O* ou un encerclement, par exemple sont identiques). Le recours à une décision contextuelle serait souvent souhaitable mais pose les problèmes de la définition de ces contextes et de leur manipulation qui risque de ralentir l'interaction sans garantir une parfaite fiabilité. Les commandes gestuelles font l'objet de procédures spécialisées pour leur reconnaissance.

3. Le traitement automatique des dessins de tableaux

Nous présentons le traitement automatique des tracés de tableaux qui a pour but de donner une forme normalisée au dessin à main levée. Les tableaux traités se présentent comme des ensembles de cellules rectangulaires connexes, ne comportant pas de lignes diagonales. Le stylo permet de saisir le dessin sous forme de suites de points correspondant aux divers tracés [fig. 1]. Ces données ne seront interprétées qu'à la fin du dessin afin d'exploiter une vue globale du tableau et de ne pas perturber le dessinateur en lui retournant des informations visuelles pendant son activité de production de dessin, voir à ce propos une argumentation sur la reconnaissance différée dans [Nakagawa1993a].

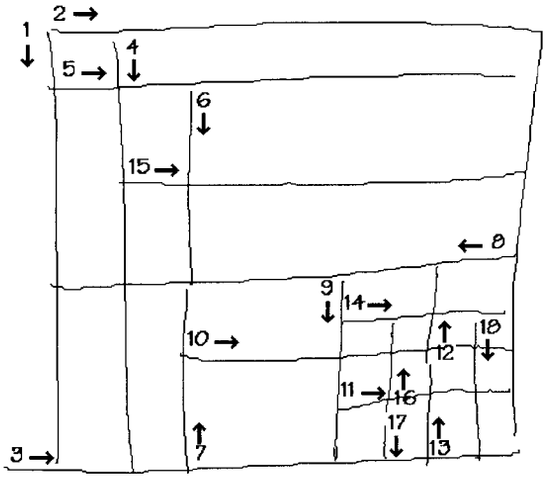


Fig. 1. – Dessin original et numérotation des tracés dans l'ordre de production.

3.1. INTERPRÉTATION DES TRACÉS

L'interprétation automatique est présentée par étapes, leur description souligne que les choix effectués pour les traitements et les représentations sont liés aux contraintes énoncées ci–dessus.

La première étape de la remise en forme automatique dans TAPAGE [fig. 2] consiste à polygonaliser chaque tracé en n'autorisant que des segments verticaux (V) et horizontaux (H). La suite des segments de la ligne polygonale est obtenue par comparaison des abscisses et ordonnées d'un ensemble de points repères du tracé. Cette opération utilise un seuil fixé *a priori*, qui a été déterminé en fonction de la résolution de l'écran. L'algorithme de polygonalisation fonctionne par dichotomie récursive, il est assimilable à un algorithme de la corde ou à l'étape *split* d'une méthode *split and merge* [Pavlidis 1977]. Pour chaque tracé, le critère d'alignement de points repères est évalué. Les points repères d'un tracé de longueur L sont situés aux positions : 0, $L/4$, $L/2$, $3L/4$, L. Quand ce critère n'est pas vérifié (c'est le cas du tracé n°2 qui correspond à une verticale et une horizontale du cadre du tableau), le tracé est découpé en deux, au niveau d'un point repère, et l'algorithme est appliqué sur les tracés résultants. La figure 2 montre les segments H et V obtenus après cette étape, l'allure du tableau original est rappelée en pointillé.

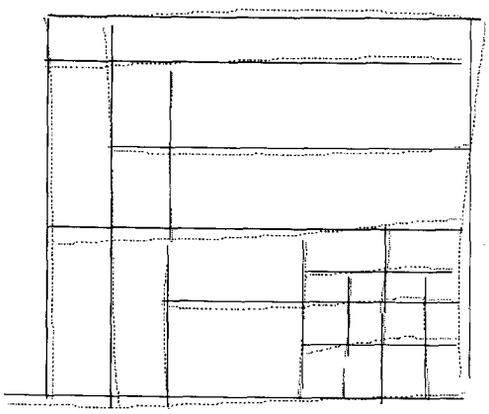


Fig. 2. – Recherche des segments localement verticaux ou horizontaux.

Le choix de points repères pré–définis rend l'algorithme de la corde sous–optimal mais évite de calculer les distances à la droite d'approximation pour chaque point et par suite permet un gain de temps. Ce choix n'affecte pas de manière visible les performances de la segmentation en V et H. De plus, les informations relatives aux sens des tracés sont perdues intentionnellement afin d'uniformiser la notion de verticale (segment allant de haut en bas) et celle d'horizontale (segment allant de gauche à droite), ce qui permet d'utiliser une structure de données plus légère dans les étapes suivantes.

Lors de la seconde étape il y a recherche d'alignements des segments construits à l'étape un, c'est le *merge* de la méthode *split and merge*. Les segments obtenus ne sont pas forcément verticaux ou horizontaux, c'est l'étape trois qui va les redresser. C'est au cours des étapes deux et trois que l'on va commencer à maintenir des listes de coordonnées correspondant à des grilles d'aimantation définies localement.

Une grille d'aimantation [fig. 3] englobe chaque tracé interprété H ou V. Sa taille est calculée pour chacun des tracés en fonction de la dispersion des points initiaux du tracé brouillon autour du segment H ou V reconstruit par l'algorithme de segmentation. Plus la dispersion de ces points est grande (plus le dessin est brouillon) plus la taille de la grille sera importante. Cette grille définit des seuils locaux qui seront utilisés pour la suite de l'interprétation.

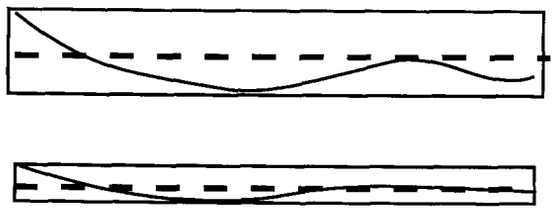


Fig. 3. – Exemples de grilles d'aimantation. En trait plein, le tracé initial. En pointillé le segment reconstruit.

La quatrième étape de correction automatique permet d'éliminer les petits dépassements et de remplir les espaces au niveau des jonctions en T ou en L, [fig. 4]. Les coordonnées des segments V et H reconstruits aux étapes précédentes, associées aux tailles des grilles d'aimantation locales, vont permettre la détermination des références (notées *Réf*) et la reconstruction des jonctions T et L.

Algorithme de recherche des références :

– Soit X_1, la première abscisse traitée, alors $RéfX_1 = X_1$ est la première référence pour les abscisses.

– Soient X_i une autre abscisse et j le nombre de références d'abscisses définies à cet instant.

– Si $X_i \approx RéfX_k$ avec $k \in [1, j]$ et $\approx$ qui signifie « est proche de » et utilise les seuils locaux définis à partir les grilles d'aimantation.

Alors $RéfX_k = \frac{(RéfX_k + X_i)}{2}$

Sinon $RéfX_{j+1} = X_i$ est une nouvelle référence pour les abscisses.

De même pour les ordonnées.

Tous les segments sont alors définis grâce à ces références. L'effet d'aimantation est une égalisation de la valeur des abscisses (resp. ordonnées) des points qui sont à l'intérieur d'une même grille d'aimantation définie pour un segment V (resp. H). La conséquence sera une reconstruction de la plupart des jonctions T et L, comme on peut le constater [fig. 4].

La structure de représentation est illustrée [fig. 5]. Chaque segment est représenté par un entier indiquant sa classe de direction, H ou V, et des pointeurs sur des listes de références qui spécifient la position du segment (pointage sur une référence X, resp. Y,

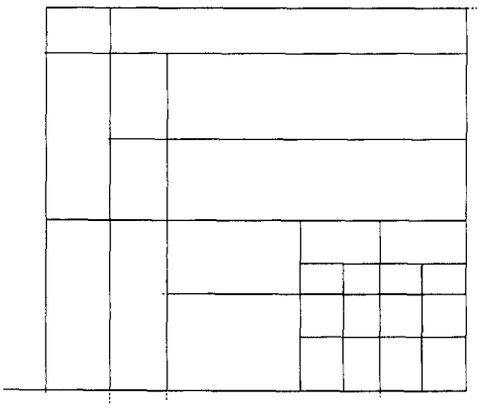


Fig. 4. – Reconstruction des jonctions par le calcul des références.

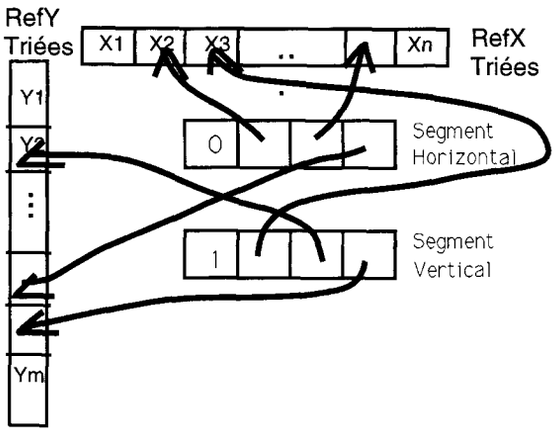


Fig. 5. – Structure des données.

pour un segment V, resp. H) et des points extrémités. Par exemple, quand un segment H est en contact avec deux segments V à chacune de ses extrémités, ce qui est toujours vrai dans le cas des tableaux à cellules fermées, les références X pointées par H sont aussi les références de position de ces segments V.

Cette organisation des données permet, comme on le verra plus loin (section 4) de propager rapidement des modifications locales dans l'ensemble du tableau par simple changement d'une valeur de référence et héritage de cette valeur pour les segments qui pointent sur elle.

L'étape précédente a éliminé la plupart des extrémités libres des segments uniquement par un choix de structure de données, il peut toutefois en rester d'autres, le seuil d'aimantation s'avérant trop faible. La cinquième étape du traitement consiste à rechercher les extrémités libres puis à déterminer le segment perpendiculaire le plus proche de chaque extrémité libre afin de reconstruire une jonction en T ou en L. Ces opérations de recherche sont particulièrement coûteuses en temps de calcul. L'intérêt de l'étape 4 est très visible sur cet exemple : il ne reste à l'étape 5 qu'une seule jonction à traiter (jonction L en bas à gauche sur la figure 4) alors que l'étape 4 a déjà traité 21 jonctions T ou L.

3.2. INTERPRÉTATION DES STRUCTURES

Après avoir reconstruit les tracés en termes de segments V et H et de jonctions T et L, le tableau n'a pas encore la structure exacte que souhaitait lui donner son concepteur : certaines différences de tailles entre des lignes et des colonnes ne sont pas significatives, elles résultent de l'imprécision introduite par la production manuscrite. Une analyse globale de la structure du dessin obtenu est nécessaire pour éliminer ces différences qui introduisent des informations parasites. Les travaux de [Bertin 1977], en sémiotique graphique, ont conduit à des règles de présentation pour l'expression graphique qui ont été reprises, entre autres, pour la construction de schémas à partir de spécifications formelles [Marks 1990] et la programmation visuelle [Ladret et Rucher 1991]. Bertin souligne que toute différence (de taille, de forme, d'épaisseur de trait ...) produit forcément chez le lecteur une tentative d'interprétation. La structure du tableau sera donc traitée en suivant la consigne de ne conserver que les différences significatives. Un critère numérique est utilisé pour décider si une différence entre les tailles de lignes et de colonnes correspond à une intention du dessinateur ou relève de l'imprécision du tracé à main levée. La mise en œuvre de ce critère de normalisation impose une propagation de décisions locales dans l'ensemble de la structure. La figure 6 illustre la différence entre le tableau avant et après l'analyse de la structure.

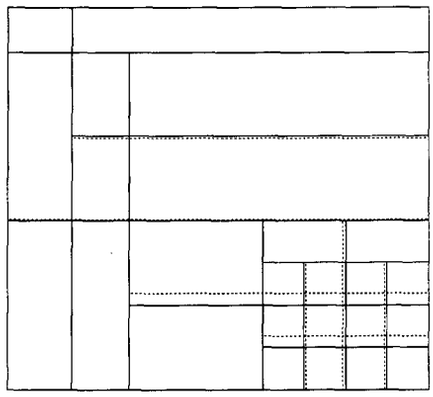


Fig. 6. – Égalisation des motifs.

La normalisation de la structure est réalisée par un module de traitement distinct de celui qui traite les tracés. Elle est faite dynamiquement par un algorithme récursif qui recherche les *motifs* qui permettent la remise en forme finale du tableau. Un motif est défini comme une zone entre deux segments H ou V successifs, il est caractérisé par sa largeur s'il s'agit d'un motif horizontal (dans l'exemple de la figure 7, ce sont des motifs horizontaux qui sont recherchés) et par sa hauteur pour un motif vertical. Les motifs horizontaux sont d'abord recherchés, puis les motifs verticaux. On introduit la notion de *segment fixe* et de *segment mobile*. Au départ seuls les segments les plus à l'extérieur (c'est–à–dire le cadre du tableau) sont fixes. Les motifs H

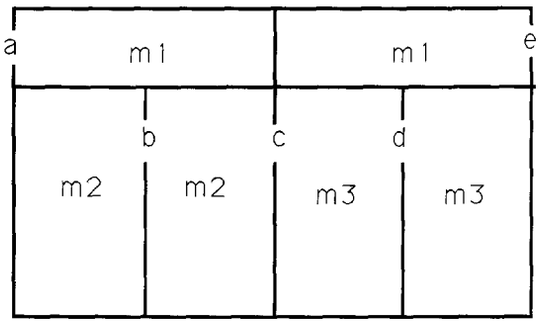


Fig. 7. – Exemple : l'algorithme récursif de recherche de motifs identiques détermine les espaces de recherche (a,e) (a,c) (c,e).

(respectivement V) sont recherchés entre deux segments V (resp. H) successifs fixes qui forment *l'espace de recherche* le long d'un segment H (resp. V) qui constitue *la droite de recherche*. L'ordre de recherche est fixé d'une part par les positions des segments dans le tableau, de gauche à droite pour les horizontaux, de haut en bas pour les verticaux, et d'autre part dynamiquement à mesure que les segments sont fixés. Sur la figure 7, la suite des espaces de recherche des motifs verticaux est (a, e) (a, c) (c, e). Les segments sont fixés dans l'ordre : a, e, c (motifs $m1$), b (motifs $m2$), d (motifs $m3$).

Deux motifs sont « identiques » quand les grandeurs qui les caractérisent (largeur ou hauteur) sont égales, à la tolérance prés du seuil d'égalisation qui est choisi de manière empirique. La normalisation fixe les segments mobiles qui délimitent les suites de motifs « identiques » dans des positions où l'égalité des grandeurs est stricte. On note que l'identité des motifs $m2$ et $m3$ résulte d'un héritage de propriétés et non d'une connaissance explicitement représentée dans la structure de données. C'est parce que le motif $m1$ a permis de séparer le tableau en deux parties égales que $m2$ et $m3$ se retrouvent être égaux. L'algorithme exploite des propriétés structurelles pour limiter le calcul à ce qui est strictement nécessaire. Le tableau résultant des traitements des tracés et de sa structure est affiché sur l'écran, moins d'une demi seconde après la fin de la saisie du dessin.

4. La coopération dessinateur–machine

L'interprétation des dessins est effectuée à chaque étape de la conception incrémentale. L'utilisateur peut faire évoluer son dessin par des modifications : il peut ajouter, déplacer, dupliquer ou effacer des composantes graphiques. La structure du tableau est définie comme un ensemble de cellules fermées contiguës. Elle doit être conservées quand l'utilisateur provoque des modifications locales. Le système prend alors en charge la mise à jour du tableau et reconstruit une structure valide en tenant compte de la structure préexistante et des changements apportés. Il y a donc partage de la tâche entre l'utilisateur et la machine qui propage la modification locale provoquée par l'utilisateur à l'ensemble du tableau en réalisant des modifications supplémentaires que l'utilisateur n'a pas besoin de préciser. Par exemple, si un segment est déplacé, les segments en contact avec lui suivent le mouvement, les jonctions entre segments sont automatiquement mises à jour grâce à la structure de donnée adoptée. Sur la figure 8, le déplacement de la verticale entraîne les horizontales dont les abscisses des extrémités droites pointent sur la même valeur de référence que l'abscisse de la verticale. Dans le cas d'un effacement, figure 9, les extrémités libres sont éliminées par la disparition de la valeur de référence correspondant au segment effacé.

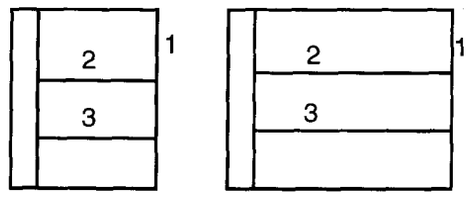


Fig. 8. – Entraînement des segments 2 et 3 par déplacement du segment 1.

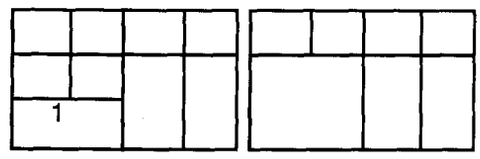


Fig. 9. – Élimination de l'extrémité libre après effacement du segment 1.

Le tableau affiché peut ne pas correspondre aux intentions de l'utilisateur soit parce que la normalisation ne conduit pas au tableau souhaité par l'utilisateur ou parce que bien les intentions de l'utilisateur ont changés, ce qui est caractéristique de la conception incrémentale. Dans ces deux cas, l'utilisateur dispose d'outils pour modifier son tableau. Ces outils peuvent être appelés à l'aide de commandes multimodales qui exploitent la tendance humaine naturelle à combiner geste et parole [Hauptmann 1993], ou l'utilisation de pointages au stylo qui spécifient l'action par une sélection sur des menus, et les variables nécessaires à l'action en pointant des objets et des positions sur le dessins, ou encore des commandes purement gestuelles. Ces dernières correspondent aux actions de sélection, effacement et déplacement [fig. 10].

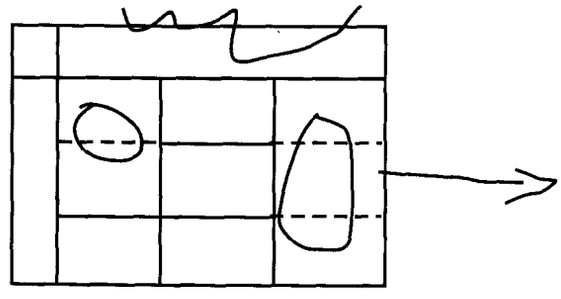


Fig. 10. – Les commandes gestuelles de déplacement, sélection et effacement.

La commande de sélection par entourage est plus expressive que le pointage d'un segment, il devient possible de sélectionner une partie de segment ou plusieurs segments. Les commandes gestuelles d'effacement et de déplacement sont concises, elles permettent de spécifier simultanément la nature de l'action, associées à la forme des tracés, et les objets et positions sur lesquelles portent l'action.

L'interprétation des commandes gestuelles implique d'abord une reconnaissance des tracés pour déterminer l'action et ensuite une analyse du contexte graphique pour savoir à quels objets (et à quelle position dans le cas du déplacement) l'action s'applique.

Les critères utilisés pour la reconnaissance des tracés des commandes sont : la fermeture, le nombre de directions principales du tracés estimées à partir d'un histogramme des directions, l'existence de changements brusques de direction au cours du tracé, le nombre de tracés de la commande. La sélection et l'effacement présentent au moins trois directions principales. La fermeture ne suffit pas à les discriminer car il arrive que le geste d'effacement revienne sur lui-même et conduise à la détection d'un tracé fermé. Dans ce cas, l'existence d'un changement brusque de direction est un critère de discrimination entre ces deux commandes. Pour pouvoir fusionner les deux tracés de la flèche en une même commande, il doit y avoir proximité temporelle entre les événements correspondant à la fin du premier tracé et au début du second.

Les rectangles englobant les tracés permettent de localiser les objets sur lesquels les actions doivent porter en examinant la superposition spatiale des englobants et des objets graphiques affichés. La position relative des centres de gravités des englobants du corps et de la tête de la flèche donne la direction du déplacement.

5. Tests d'utilisation

L'éditeur interactif de tableau a été testé sur des sept utilisateurs (un expert, trois initiés et trois novices) qui devaient recopier sept tableaux plus ou moins complexes, soit 49 sessions d'interaction. Les tableaux imprimés donnés comme modèles constituent des cibles que les tous les utilisateurs ont atteint après des temps d'interaction et des recours aux corrections variables (en moyenne 1mn 3s d'interaction). Les durées des sessions de recopie ne font pas apparaître de différences significatives entre les initiés et les novices. On note quatre abandons suivis de reprise à zéro par certains utilisateurs. Ces abandons sont dus soit à un tracé de tableau qui dépasse les limites de la surface de l'écran, soit à des corrections trop nombreuses. Si les réactions de la machine sont immédiates, le résultat de ces réactions n'est pas toujours conforme aux intentions de l'utilisateur qui doit alors corriger le tableau affiché. Quand les différences de taille des lignes ou des colonnes sont faibles, la système égalise des motifs que l'utilisateur veut différencier. Les références créées dans la structure interne du tableau peuvent conduire à de nombreuses corrections que l'utilisateur préfère éviter en recommençant le dessin.

Cette expérimentation montre que les différences entre le tableau obtenu et le tableau que l'utilisateur veut réaliser proviennent des seuils d'égalisation qui sont néanmoins nécessaires. En effet, la résolution des surfaces réactives des ordinateurs à stylo est assez grossière, de l'ordre de 30 points par cm, et la parallaxe rend difficile une localisation exacte du stylo sur l'écran. En l'absence d'égalisation automatique des motifs par l'analyse des structures (cf. section 2.2), il serait impossible d'obtenir des lignes et des colonnes égales compte tenu de la résolution et de la parallaxe. Par contre, cette normalisation de structure empêche le dessinateur de créer intentionnellement des lignes ou des colonnes adjacentes de taille légèrement différentes. La séparation, telle qu'elle existe dans ce prototype, entre le module de traitement des tracés et celui de l'analyse des structures devra donc être exploitée pour empêcher des égalisations et permettre des différences minimes entre tailles de motifs en désactivant ce dernier module à la demande de l'utilisateur.

Les commandes gestuelles ont fait l'objet de tests spécifiques en situation d'interaction utilisateur-machine. La tâche consistait à effacer des objets et à les sélectionner par entourage, à se déplacer dans l'espace graphique en dessinant une flèche et à mesurer des distances entre objets en traçant une ligne droite. Les 19 utilisateurs étaient tous novices : ils n'avaient jamais, ou une seule fois, utilisé un ordinateur à stylo. Ces tests ont été menés dans le but de mesurer l'efficacité des commandes. Une commande est efficace quand elle est exécutée. De nombreux facteurs conduisent à des commandes gestuelles inefficaces : le tracé n'est pas, ou seulement partiellement, enregistré (ce qui se produit lorsque l'utilisateur appuie par inadvertance sur le bouton situé sur le coté du stylo ou lorsque le dispositif de saisie n'est pas disponible pour recevoir une commande), le tracé n'est pas reconnu, le tracé est reconnu mais confondu avec une autre commande, le tracé est reconnu mais les variables nécessaires à l'action ne sont pas identifiées. Les tests concernent donc l'ensemble des facteurs qui conditionnent l'exécution d'une commande gestuelle. Pour vérifier le « naturel » de ces gestes, nous n'en avons pas donné d'exemples graphiques. L'utilisateur reçoit seulement des consignes orales. En particulier, le geste d'effacement est décrit comme étant un zigzag superposé à l'objet à effacer et le geste de déplacement comme une flèche dessinée en deux morceaux. Ces deux commandes gestuelles sont réalisées de manière très variable (différence de forme, de taille, de vitesse) par les différents utilisateurs. Les résultats obtenus pour ces commandes sont détaillés.

Sur 216 intentions d'effacement, 156 ont demandé un seul geste, 40 deux gestes, 10 trois gestes et 10 plus de trois gestes. Les intentions nécessitant plusieurs gestes sont le plus souvent au début de la session quand l'utilisateur est peu habitué à ce type de commande. Dans les cas difficiles, en début de session, l'expérimentateur guide verbalement l'utilisateur en lui recommandant, par exemple, de ne pas appuyer sur le bouton du stylo, de raturer les objets à effacer d'un geste rapide et ample ou de dépasser le cadre qui entoure l'espace où sont disposés les objets pour effacer

ceux du bords qui sont partiellement visibles. Dans le cas des déplacements, sur 318 flèches tracées, 253 flèches ont conduits à un déplacement ou un message indiquant que le déplacement n'était plus possible, la limite étant atteinte. Vingt six flèches, sur les 65 qui n'ont pas pas conduit à une commande effective, sont dessinées en un ou trois tracés.

On peut noter que dans une expérimentation où les préférences des utilisateurs étaient testées [Faure 1995], la commande gestuelle d'effacement a toujours été choisie, malgré une efficacité de l'ordre de 75%, de préférence à l'utilisation du menu et pointage d'objet, beaucoup plus efficace.

6. Conclusion

Un prototype pour l'édition interactive de tableau a été conçu avec pour objectif de réaliser une étude de cas dans le cadre de la conception incrémentale de documents graphiques où l'utilisateur dispose d'un stylo pour produire les données graphiques et des commandes. Cette étude à permis de définir des traitements et des structures de données particulièrement efficaces pour obtenir une réaction rapide lors de la normalisation et de la propagation des modifications locales dans la structure. A propos de la normalisation des structures, ce prototype a fait apparaître le problème de la collaboration d'un dessinateur et d'une machine qui dispose d'une connaissance pour normaliser le dessin suivant un modèle standard. L'acceptation de ce type de systèmes interactifs implique que l'utilisateur puisse produire des cas spécifiques, rares, avec l'intention de ne pas se conformer au modèle le plus général du domaine. Ceci devra être pris en compte pour définir, au niveau de l'interaction, les moyens de spécifier un usage personnalisé des traitements.

Les applications actuellement en cours de développement s'appuient sur les acquis de ce premier prototype qui a permis la réutilisation de certains composants logiciels, tout d'abord l'interface et en particulier la reconnaissance des commandes gestuelles, les procédures de traitement des tracés qui ont été généralisées par la suite à des lignes de direction quelconque. Les structures de données sous forme de listes de références ont aussi été retenues pour d'autres applications. On s'intéresse plus particulièrement à des systèmes interactifs pour l'aide à l'aménagement de pièces d'habitation et à l'édition de diagrammes en réseaux [Julia 1995] où la reconnaissance des figures géométriques combine une généralisation aux obliques du traitement des tracés de tableaux et une méthode de reconnaissance proche de celle des commandes gestuelles.

BIBLIOGRAPHIE

[Apte] A. Apte, V. Vo, T.D. Kimura, « Recognizing Multistroke Geometric Shapes : An experimental Evaluation », *Proc. of the ACM Symposiom on User Interface Software and Technology*, 1993, pp. 121-128.

[Bertin] J. Bertin, *La graphique*, Flammarion, 1977.

[Endo] Y. Endo, S. Akimichi, M. Milne, « The Context–Based Graphic Input System : T–Board », In *Proc. HCI International'93*, Orlando, Elsevier, 1993, pp. 1004-1009.

[Faure] C. Faure , L. Julia, « Interaction homme–machine par la parole et le geste pour l'édition de documents : TAPAGE », *L'interface des mondes réels & virtuels : Montpellier*, 1993, p. 171-180.

[Faure] C. Faure, L. Julia , « An agent based architecture for a multimodal interface », *Working notes. AAAI Spring Symposium*, 1994, pp. 82-86.

[Faure] C. Faure, S. Hervet, « Pen–based human–computer interaction », *7th Int. Graphonomics Conference*, London (Ontario), 1995.

[Hauptmann] A.G. Hauptmann, P. McAvinney, « Gestures with speech for graphic manipulation », *International Journal of Man–Machine Studies*, 38, 1993, pp. 231-249.

[Julia] L. Julia, C. Faure, « Pattern recognition and beautification for pen–based interfaces », *Actes Int. Conf. Document Analysis and Recognition*, ICDAR'95, Montréal, 1995.

[Ladret] D. Ladret, M. Rucher, « VLP : a visual logic programming language », *Journal of Visual Languages and Computing*, Vol. 2, n°2, 1991, pp. 163-188.

[Machii] K. Machii, H. Fukushima, M. Nakagawa, « On–line Text/Drawings Segmentation of Handwritten Patterns », *Proc. 2nd Int. Conf. on Document Analysis and Recognition (ICDAR'93)*, 1993, pp. 710-713.

[Marks] J. Marks, E. Reiter, « Avoiding Unwanted Conversational Implicatures in Text and Graphics », *Proc. Eight National Conference on Artificial Intelligence*, The MIT Press, 1990, pp. 450-456.

[Murase] H. Murase, T. Wakahara, « On–line Hand–sketched Figure Recognition », *Pattern Recognition*, Vol. 19, n°2, 1986, pp. 147-160.

[Nakagawa,a] M. Nakagawa, N. Kato, K. Machii, T. Souza, « Principles of Pen Interface Design for Creative Work », *Proc. 2nd ICDAR*, 1993, pp. 718-721.

[Nakagawa,b] M. Nakagawa, S. Kazaa, T. Satou, N. Fukuda, « Pen–based interfaces for drawing fiures with 'Stationary Metaphors' », *Proc. HCI International'93 : Orlando*, 1994, p. 1046-1051.

[Pavlidis] T. Pavlidis, *Structural Pattern Recognition*, Springer Verlag, 1977.

[Rubine] D. H. Rubine, « The Automatic Recognition of Gestures », *Thesis summary, Carnegie Mellon University*, 1990. [Stolpmann] M. Stolpmann, D. Roller, « Sketching Editor for Engineering Design », *Proc. HCI International'93 : Orlando*, 1993, pp. 609-614.

[Tappert] C.C. Tappert, C.Y. Suen, T. Wakahara, « The state of the art in on–line handwriting recognition », *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 12, n°8, 1990, pp. 787-808.

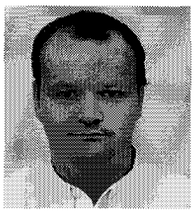
[Zhao, a] R. Zhao, « Incremental Recognition in Gesture–Based and Syntax–Directed Diagram », *Proc. of the ACM conf. InterCHI'93*, Amsterdam, 1993, pp. 95-100.

[Zhao, b] R. Zhao, « Gesture Specification and Structure Recognition in Handsketch–Based Diagram Editors », In *Proc. HCI International'93* (Orlando), Elsevier, 1993, pp. 1052-1057.

Manuscrit reçu le 23 Janvier 1995.

LES AUTEURS

Luc JULIA



Luc Julia a obtenu une maîtrise d'informatique et le DEA Informatique Théorique, Calcul et Programmation à l'université P. et M. Curie (Paris VI). Il a ensuite été doctorant à l'École Nationale Supérieure des Télécommunication. Sa thèse porte sur les interfaces multimodales intégrant le geste graphique et la parole. Il et actuellement chercheur invité au SRI International, Stanford, où il s'intéresse plus particulièrement à l'interaction multimodale et aux architectures à base d'agents pour les systèmes distribués et collaboratifs.

Claudie FAURE

Après une maîtrise de physique à l'Université de Nice, Claudie Faure s'est orientée vers le traitement du signal et la reconnaissance des formes, d'abord à l'université d'Orsay puis à l'université de technologie de Compiègne où elle a obtenu une thèse d'État en 1982. Elle est chargée de recherche au CNRS et travaille depuis 1985 dans l'URA CNRS 820 à l'ENST. Sa recherche actuelle concerne les modes d'expression graphique du point de vue de la perception, de la production, de leur traitement informatique et de leur usage dans des systèmes interactifs.