
L'interprétation de tableaux dans une situation d'interaction utilisateur-machine

Interpretation of tables in a Human-Computer interaction context

par Luc JULIA & Claudie FAURE

ENST Dpt SIGNAL - CNRS URA 820
46 rue Barrault, 75634 Paris Cedex 13 - FRANCE
e_mail : julia@sig.enst.fr ou cfaure@enst.fr

Résumé

Dans le cadre des interfaces multimodales d'aide la conception incrémentale de documents graphiques, nous avons développé un prototype pour l'édition de tableaux par la parole et le geste : TAPAGE. La partie relative à l'interprétation des tableaux à main levée est présentée ici. 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 dessins.

Abstract

The studies on the incremental graphic design using multimodal interfaces led us to realize an initial prototype : a speech and gestures table editor, TAPAGE. 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 drawing interpreter.

1. Introduction

Le travail présenté concerne 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 décrits dans : [Faure93], [Faure94], [Julia]. 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 digitaliser 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] 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], [Machii], [Zhao93a]. 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], [Apte], [Endo], [Nakagawa93b], [Stolpmann], [Zhao93b]. 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 "idéale" 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 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 idéalisée est retournée. 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

même à l'intérieur d'un même dessin. Cette qualité semble liée à la vitesse d'exécution du dessin. Les seuils de tolérance qui sont forcément impliqués dans les décisions de l'interpréteur ne doivent pas être figés, ils 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 signalée. Le stylo utilisé à la fois pour produire les données (les dessins et du texte) et pour effectuer des commandes gestuelles fait apparaître la contrainte d'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 l'interprétation automatique de tableaux qui a pour but de donner une forme idéale au dessin à main levée. Les données fournies par le stylo sont des suites

de points correspondant aux divers tracés [figure 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 [Nakagawa93a].

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 [figure 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 un algorithme de la corde ou à l'étape *split* d'une méthode *split and merge* [Pavlidis]. 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), 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 pointillés.

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 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 [figure 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 pointillés le segment reconstruit.

Comme on le voit sur la figure 4, 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. 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 Ref) et à la reconstruction des jonctions T et L.

Algorithme de recherche des références :

- Soit X_1 , la première abscisse traitée, alors $RefX_1=X_1$ 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 $X_i \approx RefX_k$ avec $k \in [1, j]$ et \approx qui signifie "est proche de" et utilise les seuils locaux définis à partir des grilles d'aimantation.

Alors $RefX_k = (RefX_k + X_i) / 2$ Sinon $RefX_{j+1} = X_i$ 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 (respectivement ordonnées) des points qui sont à l'intérieur d'une même grille d'aimantation définie pour un segment V (respectivement H). La conséquence sera une reconstruction de la plupart des jonctions T et L, comme on peut le constater sur la figure 4.

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

La structure de représentation est illustrée sur la figure 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, 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 cellules fermées, les références X pointées par H sont aussi les références de position de ces segments V.

Fig. .5– Structure des données.

Cette organisation des données permettra 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 se divise en deux parties : la recherche d'une extrémité libre puis la détermination du segment perpendiculaire le plus proche pour effectuer une connexion en T ou en L. Ces opérations sont particulièrement coûteuses en temps de calcul (surtout la seconde). 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 gauche sur la figure 4) alors que l'étape 4 a déjà traité 21 jonctions T ou L.

3.2. Interprétation des structures

Le tableau a désormais une allure correcte, il n'a toutefois 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 sémantique du dessin obtenu est nécessaire pour éliminer ces différences. En effet, toute différence (de taille, de forme, d'épaisseur de trait...) produira forcément chez le lecteur une tentative d'interprétation [Bertin], [Marks].

Fig. .6– égalisation des motifs.

La figure 6 illustre la différence entre le tableau avant et après cette analyse. Nous sortons ici du module d'interprétation des tracés en termes de segments H et V et de jonctions T et L pour rentrer dans le module d'interprétation de la structure qui constitue l'étape

sémantique, nous verrons plus loin le pourquoi de cette séparation.

Cette correction sémantique 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 (respectivement V) sont recherchés entre deux segments V (respectivement H) successifs fixes qui forment l'espace de recherche le long d'un segment H (respectivement 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).

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

Deux motifs sont "identiques" quand les grandeurs qui les caractérisent (largeur ou hauteur) sont égales, à la tolérance près du seuil d'égalisation. La remise en forme 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 m_2 et m_3 se retrouvent être identiques. L'algorithme exploite des propriétés structurelles pour limiter le calcul à ce qui est strictement nécessaire.

Le tableau définitif 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 conservée 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– Elimination de l'extrémité libre après effacement du segment 1.

Le tableau affiché peut ne pas correspondre aux intentions de l'utilisateur : une erreur d'interprétation s'est produite ou bien les intentions de l'utilisateur ont changées, 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], ou l'utilisation de pointage au stylo sur des menus, des objets et des positions dans le dessin, ou encore des commandes purement gestuelles. Ces dernières correspondent aux actions de sélection, effacement et déplacement [figure 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. La détection de la fermeture du tracé permet de reconnaître cette commande. Pour les autres, les tracés sont polygonalisés à l'aide du même algorithme que celui utilisé pour les tableaux, ici les alignements des points de référence se font sur des directions quelconques (et non sur des verticales et des horizontales comme précédemment). Les segments obtenus par polygonalisation des courbes sont codés sur huit directions et le nombre de changements de direction dans un tracé est utilisé pour les discriminer : le corps de la flèche ne tolère aucun changement de direction, la pointe correspond à deux changements de direction et la commande d'effacement doit avoir au moins trois change-

ments de direction. Il faut aussi ajouter que la flèche est composée de deux parties dont les tracés sont séparés par un court intervalle de temps prédéfini, la position de la tête de la flèche devant être localisée à une extrémité du tracé du corps de la flèche.

La reconnaissance de la forme des tracés des commandes gestuelles permet de connaître l'action souhaitée par l'utilisateur. Il faut encore analyser la relation du tracé au contexte graphique affiché pour savoir à quels objets l'action doit être appliquée. Le rectangle englobant une commande permet de localiser les objets sur lesquels elle porte en examinant la superposition spatiale de ce rectangle et des objets graphiques affichés.

La résolution des surfaces réactives des ordinateurs 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 de l'égalisation automatique, qui correspond à l'interprétation de structure (cf.3.2), il serait impossible d'obtenir des lignes et des colonnes égales compte tenu de la résolution et de la parallaxe. Par contre, si le dessinateur désire créer intentionnellement des lignes ou des colonnes adjacentes de taille légèrement différentes, il verra le système égaliser à chaque interprétation. La séparation entre le module d'interprétation des données et celui de l'interprétation des structures peut être exploitée pour empêcher des égalisations et permettre des différences minimales entre tailles de motifs en désactivant ce dernier module à la demande de l'utilisateur.

5. Conclusion

L'interaction utilisateur-machine fait apparaître la nécessité de traiter rapidement les données produites au stylo par le dessinateur. Les traitements et surtout la représentation interne du tableau ont été choisis pour minimiser les temps de calcul. La représentation choisie est basée sur la notion de références numériques pour organiser les données. La structure n'est pas explicite par opposition à une représentation par graphe (par exemple) qui serait plus analogue aux données visualisées. L'économie réalisée par cette forme de représentation est particulièrement visible quand le dessin subit des modifications locales qui sont propagées automatiquement dans la structure du tableau.

Pour s'affranchir de la contrainte de style, les seuils sont définis en fonction d'une qualité estimée du tracé et d'un seuil de base de faible valeur. La combinaison des grilles

d'aimantation et des représentations par références permet de reconstruire la plupart des jonctions par des ajustement numériques aveugles, au sens où l'événement graphique trait (la jonction) n'est ni recherché ni spécifiquement traité par le système.

La séparation du module d'interprétation des tracés et du module d'interprétation des structures sera maintenu pour les futures applications. Le module sémantique pose le problème fondamental de la collaboration d'un dessinateur et d'une machine "intelligente" qui n'a à sa disposition que les règles et les conventions générales du domaine. La question de la personnalisation des interfaces implique que cette intelligence standardisée puisse aussi accepter des cas spécifiques, rares, que l'utilisateur veut produire avec l'intention de ne pas se conformer au modèle le plus général du domaine.

Des tests de validation ont été faits avec des utilisateurs experts, initiés et novices qui devaient recopier des tableaux plus ou moins complexes. La structure idéale des tableaux a toujours été obtenue après des temps d'interaction et des recours aux corrections variables (en moyenne 1mn 3s d'interaction) mais ne faisant pas apparaître de différence significative entre les initiés et les novices. Les abandons de certains sujets étaient dus un tracé de tableau qui dépassait les limites de la surface de l'écran, ou à un cycle de correction-mise à jour qui durait trop longtemps. Dans ces deux cas, le sujet recommençait son dessin. Dans les cas de tableaux complexes, certains sujets demandaient l'affichage du tableau interprété en cours de dessin pour continuer à travailler sur un tableau propre.

Les applications graphiques pouvant tirer parti des ordinateurs à stylo et des interfaces multimodales sont très nombreuses. Nous pensons en particulier à des outils pour les architectes et nous avons commencé à élaborer un éditeur de diagrammes en réseaux (DERAPAGE : Dessin RAPide par la PArole et le GEste) qui, pour la reconnaissance des objets graphiques, combine les méthodes d'interprétation et de représentation des tableaux et celles de la reconnaissance des commandes gestuelles.

BIBLIOGRAPHIE

- [Apte] A. APTE, V. VO, T.D. KIMURA, "Recognizing Multi-stroke Geometric Shapes : An experimental Evaluation", Proc. of the ACM Symposium on User Interface Software and Technology, 1993, p. 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, p. 1004-1009.
- [Faure93] 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.
- [Faure94] C. FAURE, L. JULIA, "An agent based architecture for a multimodal interface". Working notes. AAAI Spring Symposium, 1994, p. 82-86.
- [Hauptmann] A.G. HAUPTMANN, P. McAVINNEY, "Gestures with speech for graphic manipulation", International Journal of Man-Machine Studies, 38, 1993, p. 231-249.
- [Julia] L. JULIA, C. FAURE, "A Multimodal Interface for Incremental Graphic Document Design", HCI International'93 : Orlando, 1993, p. 237.
- [Machii] K. MACHII, H. FUKUSHIMA, M. NAKAGAWA, "Online Text/Drawings Segmentation of Handwritten Patterns", Proc. 2nd Int. Conf. on Document Analysis and Recognition (ICDAR'93), 1993, p. 710-713.
- [Marks] J. MARKS, E. REITER, "Avoiding Unwanted Conversational Implications in Text and Graphics", Proc. Eight National Conference on Artificial Intelligence, The MIT Press, 1990, p. 450-456.
- [Murase] H. MURASE, T. WAKAHARA, "Online Hand-drawn Figure Recognition", Pattern Recognition, vol. 19, n 2, 1986, p. 147- 160.
- [Nakagawa93a] M. NAKAGAWA, N. KATO, K. MACHII, T. SOUZA, "Principles of Pen Interface Design for Creative Work", Proc. 2nd ICDAR, 1993, p.718-721.
- [Nakagawa93b] M. NAKAGAWA, S. KAZAMA, T. SATOU, N. FUKUDA, "Pen-based interfaces for drawing figures with Stationary Metaphors", Proc. HCI International'93 : Orlando, 1993, p. 1046-1051.
- [Pavlidis] T. PAVLIDIS, "Structural Pattern Recognition", Springer Verlag, 1977.
- [Rubine] D.H. RUBINE D.H., "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, p. 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, p. 787-808.
- [Zhao93a] R. ZHAO R., "Incremental Recognition in Gesture-Based and Syntax-Directed Diagram Editors", Proc. of the ACM conf. InterCHI'93, Amsterdam, 1993, p. 95-100.
- [Zhao93b] R. ZHAO, "Gesture Specification and Structure Recognition in Hand-drawn Based Diagram Editors", In Proc. HCI International'93 (Orlando), Elsevier, 1993, p. 1052-1057.