HAL
open science

Svalabard: Une table à dessin virtuelle pour la modélisation 3D

Stéphane Huot, Cédric Dumas, Gérard Hégron

To cite this version:

Stéphane Huot, Cédric Dumas, Gérard Hégron. Svalabard: Une table à dessin virtuelle pour la modélisation 3D. Actes de la 16ème conférence francophone sur l'Interaction Homme-Machine, IHM 2004, Association Francophone d'Interaction Homme-Machine, 2004, Namur, Belgique. pp.85-92, 10.1145/1148613.1148626 . inria-00550691

HAL Id: inria-00550691

https://inria.hal.science/inria-00550691

Submitted on 29 Dec 2010

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.

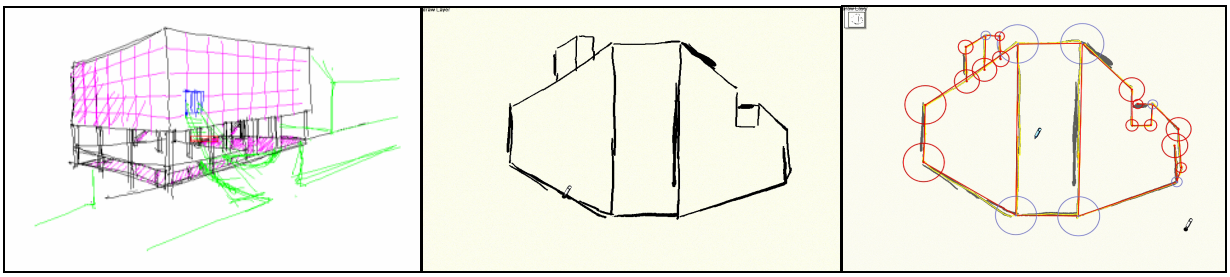
Svalabard: une table à dessin virtuelle pour la modélisation 3D

Stéphane Huot, Cédric Dumas

Ecole des Mines de Nantes
La Chantrerie - 4 rue Alfred Kastler
BP 20722
F-44307, Nantes cedex 3, France
Tel.: 1-33-2-51-85-82-41
{Stephane.Huot, Cédric.Dumas}@emn.fr

Gérard Hégron

CERMA UMR CNRS 1563
Ecole d'Architecture de Nantes
Rue Massenet, BP 81931
44319, Nantes Cedex 3, France
Tel.: 1-33-2-40-59-43-24
hegron@cerma.archi.fr



(a) **(b)** **(c)**

Figure 1: (a) Un dessin d'architecte (étude utilisateur), (b) un dessin réalisé avec notre système et (c) son interprétation 2D.

RESUME

Cet article présente une interface de dessin en perspective pour la modélisation 3D créative : Svalabard. Partant du constat d'inadéquation entre le savoir-faire des créateurs et les outils actuels, nous proposons un système de dessin libre, métaphore d'une table à dessin. L'interaction et les outils proposés sont dérivés des habitudes de travail des concepteurs (dessin multi calques, association des interactions avec des périphériques spécifiques). Svalabard intègre en particulier des filtres d'analyse sous-jacents, issus d'une étude utilisateur dans le domaine architectural. Ils permettent de détecter en temps réel le contexte courant du dessin afin d'adapter l'interaction et d'épurer le dessin. Ils produisent finalement une représentation structurée des volumes 3D dessinés.

MOTS CLES : modélisation 3D, interfaces de dessin, conception créative, interaction contextuelle, croquis.

ABSTRACT

This paper introduces a perspective sketching interface for creative 3D modeling: Svalabard. To close the gap between designers' abilities and 3D modeling tools, our proposal is a freehand sketching tool, metaphor of a drawing desk. Interactions and tools take advantage of designers work habits (multilayer drawing, association of interactions with input devices). Svalabard includes background drawing filters, developed from a user study in the architectural domain. Those filters detect the current drawing context in real-time to adapt the interface and clean the sketch. Finally, the result of the filters is a structured representation of drawn 3D volumes.

CATEGORIES AND SUBJECT DESCRIPTORS: H.5.2 [Information Interfaces and Presentation (e.g. HCI)]: User Interfaces --- Interaction styles, Graphical user interfaces (GUI), User-centered design; J.6 [Computer Applications]: Computer-Aided Engineering --- Computer-Aided Design (CAD).

GENERAL TERMS: Design, Experimentation, Human Factors.

KEYWORDS: 3D modeling, sketching interfaces, creative design, contextual interaction, perspective sketch.

INTRODUCTION

Un grand nombre d'outils informatiques pour la modélisation 3D ont été développés et utilisés depuis plusieurs années. La majorité de ces modeleurs, que nous appelleront « standard », sont efficaces et populaires, particulièrement dans des domaines comme la production multimédia ou la CAO (Conception Assistée par Ordinateur). Pourtant, leur utilisabilité dans un contexte créatif est encore ardue à cause de processus de modélisation inadaptés et d'interactions lourdes. De fait, la majorité des concepteurs sont réticents à l'utilisation de l'outil informatique dans les premières étapes de conception. La modélisation numérique 3D intervient souvent à la fin du processus de conception, produite par un spécialiste de ces outils (qui n'est donc pas le concepteur). Paradoxa-

lement, les concepteurs avec qui nous avons collaboré s'accordent à dire que l'apport d'un tel outil dans cette phase d'un projet serait un atout pour produire rapidement des modèles, exprimer des idées et les insérer dans des environnements visuels, ou des contextes d'utilisation (simulations, communication, travail collaboratif, etc.). *D'où vient donc cette inadéquation entre les outils existants et les besoins des utilisateurs « créatifs » ?*

Malgré la spécialisation des outils de modélisation 3D à des domaines ou tâches précises, ils restent difficiles à utiliser pour un non spécialiste et limitent en général la créativité de l'utilisateur. Les méthodes de modélisation et les paradigmes d'interaction utilisés pour concevoir ces systèmes sont trop éloignés des habitudes de conception des créateurs [14]. L'interaction dans les modeleurs 3D standard est souvent limitée à une multitude de menus, boutons et outils manipulant les paramètres mathématiques du modèle de données interne. C'est un principe bien adapté à l'ingénierie, mais dans un contexte créatif, l'utilisateur doit manipuler, assembler et raisonner en utilisant des primitives mathématiques plus ou moins complexes, très éloignées de sa représentation mentale du modèle qu'il conçoit [6]. Des améliorations notables ont tout de même été introduites au niveau de l'interaction pour rendre les outils proposés plus intuitifs, en utilisant des spécificités du domaine (ARCHICAD [1], pour l'architecture). Cela reste tout de même des systèmes d'ingénierie, non de création. Utilisés dans un contexte créatif, ils nécessitent toujours de gros efforts d'adaptation et d'apprentissage, leurs environnements étant trop éloignés des habitudes des créateurs*: « l'outil le plus raffiné reste au service de la main qu'il ne peut ni guider ni remplacer.* » H. Arendt cité dans [5].

C'est dans l'optique d'exploiter ces habitudes et capacités créatives que viennent se positionner les travaux présentés dans cet article. Nous proposons d'utiliser le dessin à main levée pour la modélisation 3D. En effet, le croquis, moyen simple et efficace de construire, communiquer et poser des idées, reste l'outil privilégié des créateurs dans les premières phases de travail [12]. Le système présenté est la partie interactive du projet **GINA** (**G**éométrie **I**nteractive et **NA**turelle) [10,18]. GINA propose une approche naturelle de la modélisation 3D. Le principe est de dessiner une vue perspective 2D de l'objet 3D à construire. Une seule vue perspective n'étant pas suffisante pour produire un modèle 3D, l'utilisateur doit aussi spécifier des contraintes géométriques (orthogonalité, parallélisme et incidence) ou structurelles sur les éléments du dessin. Le noyau mathématique [19] peut alors produire un modèle 3D, mais nécessite parfois la saisie de beaucoup de contraintes. C'est pourquoi il faut aussi assister l'utilisateur dans la saisie des contraintes afin de ne pas limiter, ou bloquer sa créativité. Le système doit analyser le dessin en temps réel, anticiper les actions et le comportement de l'utilisateur, sans interrompre sa tâche.

Le scénario d'utilisation est le suivant : l'utilisateur dessine sur une feuille de papier virtuelle à l'aide d'une tablette graphique. Le système interprète et analyse ses traits en temps réel. Dès qu'il y a assez d'informations ou lorsque l'utilisateur le souhaite, le noyau mathématique calcule et propose un modèle 3D qu'il est possible de manipuler et continuer à améliorer sous différents angles (détails, parties cachées, etc.). Si l'élévation 3D n'est pas possible, le système doit aussi être capable de demander des informations supplémentaires, des « conseils » (validation de choix), de manière non intrusive (sons, voix, suggestions en transparence).

Figure 2: Un exemple de configuration de notre système. Associations implicites avec différents périphériques d'entrée.

C'est dans ce sens que nous proposons, avec **Svalabard**, une métaphore de table à dessiner, ou l'utilisateur se retrouve dans un environnement familier (feuille, crayons et calques). L'interface est épurée de manière à ne pas le perturber et à fixer son attention sur son travail, comme lors d'un dessin avec une feuille et un crayon. Nous proposons des outils de dessin multicalques et des interactions simples, instrumentées et reconfigurables pour l'association implicite [2] avec les périphériques disponibles (figure 2). Le système est ainsi centré sur les aptitudes et les habitudes de l'utilisateur. Il est, par exemple, possible d'affecter un stylet de tablette à un outil de dessin particulier et la souris à un outil de manipulation. Le changement d'outil physique correspond alors à un changement d'outil logique. La manipulation des outils ou des objets graphiques (reconstruction 3D par exemple) est ainsi précise et efficace. Enfin, le traitement sous-jacent des données entrées par l'utilisateur est réalisé par des filtres modulaires, de manière transparente, dans l'esprit de l'interface effacée de P. Leclercq [13].

La première étape de notre travail a été de mener une étude sur des dessins d'architectes [7], plus précisément sur les traits composant ces dessins. Nous rappelons brièvement les résultats de cette étude dans la seconde partie de cet article, la première dressant un tour d'horizon des travaux importants dans le domaine de l'interaction en modélisation 3D. Nous présenterons dans la troisième partie les principes de Svalabard et les filtres d'interprétation du dessin à main levée du système. Finalement, nous discuterons de l'implémentation et de l'utilisation de la boîte à outil **MaggLite** [21] pour appliquer les principes de Svalabard.

ETENDRE LES INTERFACES DE MODELISATION 3D

Les interfaces WIMP sont le standard d'interaction des outils informatiques actuels. Pourtant, certaines tâches

(modélisation 3D, conception/production artistique, etc.) nécessitent des interactions améliorées, plus proches du mode de travail, des aptitudes et des méthodes de création. Beaucoup de spécialistes s'accordent sur le fait que le principe des interfaces Post-WIMP [23] est bien adapté aux applications et outils spécifiques à un domaine. D'ailleurs, les interfaces de modélisation 3D novatrices que nous allons présenter ici sont Post-WIMP.

Plusieurs approches ont été introduites pour rendre la modélisation 3D plus abordable aux débutants, mais aussi pour tirer partie des outils informatique dans les processus créatifs (pour les experts). Nous distinguons deux orientations dans les recherches sur la modélisation 3D. Une approche expérimentale, visant à introduire des paradigmes nouveaux et inhabituels. Une approche, plus scientifique, proposant des systèmes basés sur des études utilisateurs et leurs méthodes de travail.

Approches Expérimentales

En 1996, R. Zeleznik introduisit SKETCH [24], le premier outil de modélisation 3D utilisant la reconnaissance de geste. Ce système permet de construire des volumes polyédriques et de les manipuler, uniquement avec des gestes. Grâce à un ensemble de gestes appropriés, proches des primitives 3D et donc plus faciles à apprendre que des commandes arbitraires, SKETCH construit le premier pont entre le dessin à main levée et la CAO. Malgré un gain indéniable en facilité d'accès, cela reste éloigné d'un système de dessin. Il faut toujours créer des modèles 3D en assemblant des primitives prédéfinies, ce qui limite largement les possibilités créatives. TEDDY [8] réalise aussi se mélange entre le dessin et la modélisation 3D, dans un contexte bien particulier (dessin d'enfant). C'est un outil de modélisation regroupant deux modes d'interaction : dessin de formes et reconnaissance de gestes. Mais ces travaux, même si ils sont utilisables et intuitifs, sont toujours limités car produisant des modèles particuliers ou imprécis. La reconnaissance de gestes pose aussi le problème d'un ensemble limité de gestes (donc de commandes et d'objets) et la difficulté pour un débutant à les apprendre. T. Igarashi a résolu une partie de ce problème en proposant une interface suggestive dans un outil de modélisation 3D. Ce système, nommé Château [9], est un système de dessin « guidé » avec plusieurs agents de suggestion. Des propositions sont élégamment présentées à l'utilisateur qui peut les ignorer ou les valider.

Approches Scientifiques

Plus proches des méthodes « papier », les observations de tâches ou les études utilisateurs ont permis d'introduire des interfaces de dessin. D. Lamb, dans [11], a proposé une des premières approches efficaces de dessin 2D pour créer des modèles 3D. Le principe est d'interpréter automatiquement des dessins à main levée. Même si l'on trouve des limites évidentes à cause de la vue 2D imposée (axonométrie) et d'une interaction limitée, ces travaux marquent les premiers résultats significatifs issus du constat d'inadéquation entre la modélisation 3D, les périphériques 2D et les méthodes de conception. Le système Viking [15] se base sur ces mêmes méthodes de conception, mais d'une manière plus créative. Le principe est d'interpréter un dessin à base de lignes, tout en utilisant des contraintes descriptives ou structurelles spécifiées par l'utilisateur. D. Pugh qualifie alors son système de « *What you draw is what you get user-interface* ». L. Eggli propose dans [4] une méthode très proche de Viking. L'évolution majeure est le passage du dessin vectoriel au dessin à main levée. Ces deux systèmes extraient les propriétés structurelles des objets pour construire des modèles 3D. Afin de contourner les ambiguïtés dans la détection de propriétés et les problèmes de projection, la vue de l'objet dessiné doit être une axonométrie ou une coupe plane. Ces systèmes, plus naturels que les modeleurs 3D standard, restent toutefois limités en terme de créativité (spécialement pour la création architecturale, notre domaine d'application).

Un système de dessin à main levée en perspective a pourtant été proposé par O. Tolba et al. [22]. Conçu pour être utilisé dans les premières phases de la conception, cet outil de dessin projectif fournit des aides et des guides au créateur. Ce système parait facile à utiliser et efficace pour manipuler des vues en perspective des scènes dessinées. Mais le principe sous-jacent des points projectifs ne permet ni d'obtenir un modèle 3D, ni la structure spatiale des scènes et objets 3D.

Deux résultats majeurs émergent de ces travaux:

1. Les systèmes permettant d'obtenir des modèles 3D sont souvent trop contraignants pour être « créatifs » en comparaison du traditionnel « papier-crayon » (dessin vectoriel, reconnaissance de gestes).
2. Les systèmes de dessin à main levée permettent une meilleure expression des idées et des concepts, mais sont contraints par un choix de point de vue, ou ne produisent pas de modèle 3D.

Nous proposons donc de combiner les avantages créatifs du dessin à main levée en vue perspective avec une interaction instrumentale et contextuelle. Grâce à des outils avancés d'analyse du dessin, nous proposons un système sans contrainte de vue (perspective), ni de limitation (dessin « artistique »). L'ajout de modules de détection de propriétés et de reconstruction 3D permettra de rendre la modélisation 3D plus proche des méthodes de conception des utilisateurs « créatifs ».

ETUDE UTILISATEUR

Dans un premier temps, nous avons conduit une étude sur le dessin en perspective lors des premières phases de conception architecturale [7]. Nous voulions comprendre les gestes du créateur et trouver des invariants ou propriétés de cette tâche dans un contexte créatif. Le but n'était pas d'étudier la perception et les intentions de l'utilisateur, comme l'avait fait M. Suwa [20], mais de se concentrer sur la tâche de dessin au plus bas niveau, celui du trait. Nous avons introduit une taxinomie des traits architecturaux que nous avons appliquée à notre corpus de traits (les types sont : construction, principaux primaires et secondaires, détail, décor et style [7]).

Par l'analyse de la distribution temporelle de chaque catégorie de trait et de leurs transitions, nous avons identifiés trois phases dans la tâche de dessin en perspective:

- **phase constructive** (dessin de la forme principale).
- **phase de complétion et d'amélioration** (amélioration des contours et ajout de détails).
- **phase de style** (dessin de l'environnement).

Ces phases, pouvant être associées à des contextes d'interaction, se trouvent souvent ordonnées de la sorte. Même si nous n'avons pas pu en déduire un invariant chronologique, nous avons montré qu'un dessin architectural en perspective débute toujours par une phase constructive (30 premiers pourcents du temps du dessin). Cette première phase constructive est essentielle dans notre problématique. Elle comporte la plus grande partie des informations nécessaires à l'extraction des propriétés du dessin. Mais c'est aussi celle ou l'utilisateur est le plus créatif, et ne doit donc pas être perturbé. L'interaction doit donc être adaptée, les phases détectées.
Ces donnés nous ont aussi permis de calculer les probabilités de transition d'une phase à l'autre et de montrer, en analysant les temps de pause entre chaque trait, que les pauses sont plus longues lors d'un changement de phase.

La suite de cet article présente la détection de ces phases en temps réel et le « nettoyage » du dessin en structurant les entrées reçues de l'utilisateur. Mais dans un premier temps, nous exposerons les principes de Svalabard.

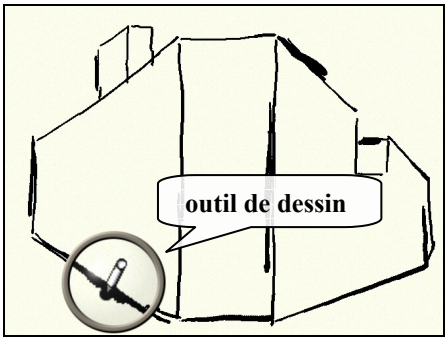


Figure 3: L'interface de Svalabard : vue du dessin.

SVALABARD

Svalabard est un système de dessin sur ordinateur qui d'analyse en temps réel le dessin produit dans le but de produire des modèles structurés. Ce système peut être décomposé en deux parties : interface et filtres d'analyse.

Les données et les analyses effectuées permettent :

1. d'adapter l'interface et les traitements à la tâche courante de l'utilisateur
2. d'utiliser le dessin comme entrée d'un noyau fonctionnel (reconstruction 3D pour le projet GINA).

Interface

L'interface et les interactions de Svalabard ont été conçues pour s'insérer naturellement dans un processus créatif, en tirant partie des aptitudes de l'utilisateur. Il s'agit d'une feuille de papier virtuelle sur laquelle le concepteur peut dessiner à l'aide d'outils. Il n'y a pas de contrainte quand à ce dessin à main levée, et c'est la différence majeure avec les systèmes de dessin pour la modélisation 3D. Aucune vue (axonométrie, plane, …) ou technique particulière (dessin vectoriel, gestes,…) n'est imposée. De plus, grâce au filtre de détection des phases de dessin que nous présenterons ci-après, l'utilisateur n'est pas limité au dessin de la forme à modéliser. Il peut embellir son dessin, dessiner l'environnement, se retrouver dans une démarche créative et artistique.

Pour parfaire la métaphore de la table à dessin, et permettre à l'utilisateur d'exprimer ses idées comme il le ferait avec ses techniques habituelles, nous avons inclus des *calques de dessin*. Des calques peuvent être ajoutés et ensuite manipulés (dessin, déplacement, masquage, suppression,…). Ils peuvent être branchés ou non sur les filtres de traitements. Enfin, il est possible de visualiser interactivement le résultat des filtres d'analyse connectés, par transparence sur le dessin original (figure 4).

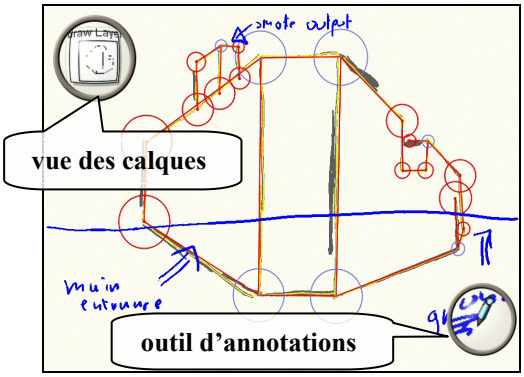


Figure 4: L'interface de Svalabard : résultat des filtres.

L'interface ne présente que la feuille de dessin et le retour graphique des outils de dessin disponibles (figure 3). Le comportement visible du système est simplement de laisser l'utilisateur dessiner. Toutes les interactions (manipulation des outils, ajout de calques, affichage des filtres, etc.) sont implicitement associées à des périphériques d'entrée. Ces associations, ainsi que les comportements de l'interface, sont entièrement reconfigurables graphiquement à l'aide d'ICon [3]. Il est ainsi possible d'adapter l'interaction aux périphériques disponibles, ou de définir des comportements particuliers du système. Nous verrons dans la partie implémentation que c'est une des propriétés de la boîte à outils MaggLite, développée comme support à Svalabard. On peut, par exemple, automatiser l'apparition du résultat visuel des filtres de traitement dès que l'utilisateur ne dessine plus (figure 4).

Analyse du dessin

Les filtres d'analyse de Svalabard permettent de structurer et nettoyer le dessin pour l'exploiter ensuite en entrée du système de reconstruction 3D. Structurer le dessin revient à transformer des entrées brutes (positions) en objets structurées et en relations entre ces objets (points, segments, segments concourants, …). Nettoyer le dessin consiste à simplifier les données et supprimer celles qui sont inutiles au noyau fonctionnel (élimination des traits des phases non constructives, fusion de points, etc.).

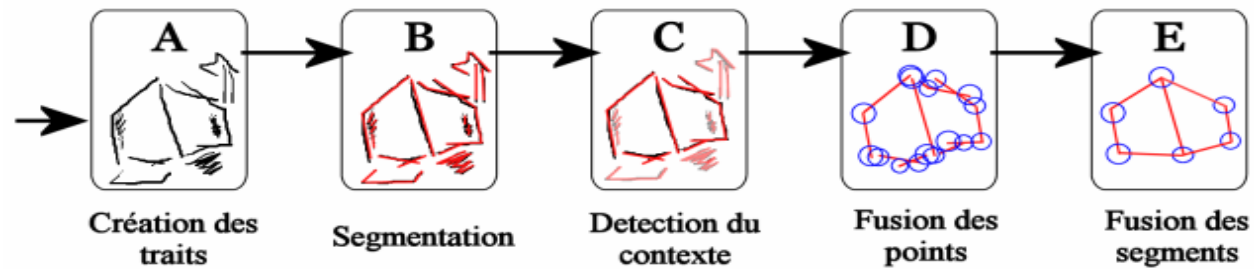


Figure 5: Filtres séquentiels de dessin.

Un filtre de détection du contexte est utilisé pour filtrer les traits des différentes phases, mais aussi pour adapter l'interface à la phase courante (affichage de résultats ou propositions, démarrage de traitements, etc.). Les filtres séquentiels suivent un paradigme de flux de données. Chaque filtre transmet au suivant le trait qu'il a reçu et les traitements qu'il a produit (figure 5).

Détection des traits (figure 5-A). Le premier filtre permet de construire les traits produits par l'utilisateur. Cet algorithme simple construit un trait à partir des points enregistrés lorsque l'outil de dessin est utilisé. Pour chaque point, le système enregistre la position 2D, le temps et la pression (si le dispositif d'entrée le permet).

Segmentation des traits (figure 5-B). Afin de structurer le dessin, il faut simplifier et décomposer les traits en un ou plusieurs segments formés uniquement de deux points. Le principe est d'éliminer tous les points « inutiles » situés entre deux points critiques (appelés coins) qui formeront les extrémités des segments (figure 6).

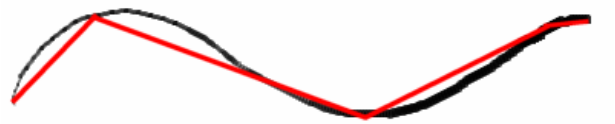


Figure 6: Segmentation de traits - un trait et son approximation.

Nous avons utilisé la méthode de T. Sezgin [16], qui combine les informations de vitesse et de courbure pour extraire les points critiques. L'algorithme calcule la courbure et la vitesse en chaque point du trait. Ensuite, des segmentations de plus en plus précises sont composées itérativement, en avec les points de plus petite vitesse et de plus grande courbure. La meilleure segmentation est gardée à chaque itération, jusqu'à ce qu'un taux de précision fixé soit atteint (distance entre l'approximation et le trait). Ce filtre reçoit donc un trait en entrée et produit en sortie un ou plusieurs segments, approximation du trait dessiné. Il reste tout de même à éliminer certains segments ainsi construits, de façon à épurer le dessin pour ne garder que les données utiles à la reconstruction 3D. C'est pourquoi, au vu de notre étude, nous introduisons un filtre de détection automatique du contexte. Cet algorithme va nous permettre de filtrer les segments des différentes phases, mais aussi d'adapter l'interaction à ces phases.

Détection du contexte (figure 5-C). Ce filtre identifie automatiquement le contexte (ou phase de dessin) à partir des segments reçus, de la phase courante et des données collectées dans [7]. Le processus de dessin architectural en perspective comporte 3 phases: construction, amélioration et style. Nous avons, pour les reconnaître en temps réel, introduit un algorithme d'évaluation basé sur les résultats de notre étude et des heuristiques simples.

Contexte	*Construction*	*Amélioration*	*Style*
Construction	95	3,5	1,5
Améliora-tion	2,5	96	1,5
Style	4	4	92

Table 1: Probabilités de transition entre contextes (%).

Lorsque le premier trait d'un dessin est reçu, le système est initialise en contexte "constructif" (un dessin débute toujours par cette phase). Lorsque un nouveau trait est reçu, le score de chaque contexte potentiel est initialisé à la probabilité de passer à ce contexte depuis le contexte courant. Ces probabilités ont été calculées d'après les données de notre étude (cf. table 1).

Trait courant	***Construction score***	***Amélioration score***	***Style score***
Pression<0,1	+80		
longueur<20	+70		
Trait dans la zone "constructive"	+100	+100	
gribouillage		+100	+100

Table 2: Pondération des scores selon le trait courant.

Contexte courant	***Construction score***	***Amélioration score***	***Style score***
Construction		+80	+15
Amélioration	+75		+25
Style	+50	+50	

Table 3: Pondération des scores pour un long temps de pause. (Ajout de *(temps de pause/20) + bonus précèdent* au score).

Ensuite, les scores sont pondérés par l'analyse des propriétés du trait reçu, du contexte courant et des propriétés du dessin. Nous avons par exemple constaté qu'il n'y avait pas de gribouillage en phase constructive. Ces traits sont donc détecté et le score des phases d'amélioration et de styles sont pondérées en conséquence. Nous analysons aussi la pression exercée sur le stylet (si disponible), la

longueur du trait, sa position par rapport aux autres traits de la phase de construction et le temps de pause lors d'un changement de phase (cf. tables 2 et 3).
Pour les temps de pauses, les pondérations sont proportionnelles aux moyennes calculées lors de notre étude. Ces valeurs expérimentales donnent des résultats satisfaisants lorsque nous les appliquons sur les dessins enregistrés lors de notre étude, ou sur de nouveaux enregistrement. Cela nous a permis d'ajuster les valeurs des paramètres. Toutefois, l'algorithme n'a pas encore été testé dans des conditions réelles d'utilisation.

Fusion des extrémités de segments (figure 5-D). Lors de dessin a main levé avec des entrées discrètes (tablette graphique, souris, …), les extrémités des segments ayant un point commun coïncident rarement. Nous devons donc, afin d'affiner la structure du dessin, détecter et regrouper ces points. Nous proposons une adaptation au traitement temps réel de la méthode de M. Shpitalni [17]. Le principe est de calculer, pour chaque extrémité de segment, une zone de tolérance dont la taille correspond à l'incertitude sur la position du point.

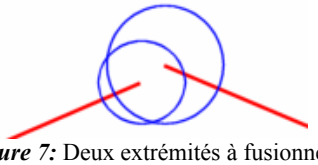


Figure 7: Deux extrémités à fusionner.

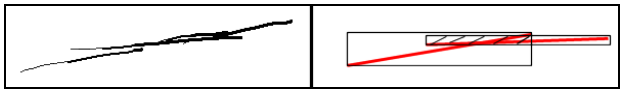
Cette zone est un cercle dont le rayon est fonction de la distance entre le point et tous les segments du dessin. Des extrémités doivent être fusionnées lorsque chacune est située dans la zone de tolérance de l'autre (figure 7). Le barycentre du groupe de points remplace donc les points et les segments sont mis à jour (cf. algorithme 1).

1. *calculer les zones de tolérance des extrémités d'un nouveau segment*
2. *former des groupes de points candidats à la fusion*
3. *remplacer chaque groupe par son barycentre*
4. *ajuster les segments et les zones de tolérance*

Algorithme 1: Algorithme de fusion des points.

L'efficacité de cette méthode est due en grande partie aux zones de tolérance adaptatives, qui prennent en compte la densité du voisinage de chaque point, et donc les aspects locaux du dessin. Originalement conçue pour un dessin terminé, notre adaptation pour le traitement temps réel se situe au niveau du calcul des zones, pondéré par la résolution (taille globale) du dessin, et donc évoluant au cours du temps. De plus, l'algorithme ignore les cas extrêmes (zones maximales) en début de dessin, et nécessite un nombre minimal de segments avant de regrouper des points pour ne pas occasionner de fusions indésirables.

Fusion de segments (figure 5-E). Nous trouvons aussi dans le dessin à main levée des cas où le dessinateur a tracé plusieurs traits pour ne former qu'un segment (figure 8a). Parfois, un trait en prolonge un autre, mais la plupart du temps les traits se recoupent ou se chevauchent (cas des traits « principaux secondaires » de notre taxinomie). Nous proposons donc un algorithme (cf. algorithme 2) permettant de fusionner de tels segments pour composer et nettoyer la représentation structurelle d'un dessin à main levée.
Les deux types de critères à retenir pour analyser de telles situations sont les critères positionnels et les critères relationnels.



(a) ***(b)***

Figure 8: Cas d'une fusion de segments.

Pour les critères positionnels, l'espace de dessin est tout d'abord divisé en zones rectangulaires de taille égales (4 ou 6, selon la taille de la zone de dessin). Deux segments doivent déjà appartenir (partiellement ou complètement) à la même zone pour être candidats à la fusion. L'algorithme procède alors au calcul de l'intersection des boîtes englobantes (figure 8b). Si l'aire de cette intersection est inférieure à un certain seuil, les segments sont potentiellement fusionnables. Dans le cas contraire, la distance entre les segments est prise en compte pour déterminer un éventuel prolongement (étape 3 de l'algorithme 2).

Ces étapes sont répétées jusqu'à ce qu'un nouveau segment soit ajouté ou qu'il n'y ai plus de fusions

1. *prendre deux segments non testés*
2. *si les 2 sont dans la même zone, continuer. Sinon retourner à l'étape 1*
3. *si les boîtes englobantes des segments ou la distance entre eux sont inférieurs aux seuils, continuer. Sinon, retourner à l'étape 1*
4. *si les segments sont concourants:*
   a. *si l'extrémité commune est aussi extrémité d'autres segments, et que l'angle entre les segments est inférieur au seuil approprié, les fusionner sinon retourner à l'étape 1*
   b. *Si l'angle entre les segments est inférieur au seuil approprié, les fusionner sinon retourner à l'étape 1*

   sinon, continuer
5. *si l'angle entre les segments est inférieur au seuil global les fusionner, sinon retourner à l'étape 1*

Algorithme 2: Fusion de segments.

Le critère relationnel est l'angle formé par les deux segments. Si cet angle est en dessous d'un seuil paramétrable, les segments remplissent alors une condition supplémentaire (étape 5 de l'algorithme 2).
Il faut tout de même considérer le cas particulier des segments concourants, tout spécialement lors de dessin en perspective:

- Si deux segments ont une extrémité commune, alors le seuil doit être plus bas pour éviter de fusionner des segments formant un angle écrasé par la projection perspective (segments AB et BC de la figure 9).

- Si 3 segments (ou plus) ont un point commun (sommet B de la figure 9), le seuil d'angle à considérer est encore plus petit. Ce cas évite la fusion de segments connectés par un point caractéristique du dessin perspective (jonction en T ou partie saillante d'un volume projeté). Si toutefois les segments sont fusionnés dans ce cas, la propriété d'incidence du point aux segments fusionnés est conservée.

Lorsque deux segments remplissent les critères ils sont alors fusionnés en un seul, selon les directions des segments originaux. Ce filtre est utilisé ici dans un but de « nettoyage » du dessin, sans retour pour l'utilisateur. Mais son comportement pourrait fournir des interactions intéressantes dans un système de dessin vectoriel (prolongement ou déplacement naturel de segments).

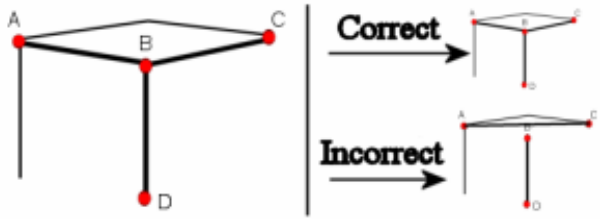


Figure 9: Fusion de segments: cas de segments concourants.

Ces filtres permettent donc de préparer, nettoyer et structurer le dessin pour des traitements de plus haut niveau. Les résultats sont très fiables pour les filtres de traitement des traits, excepté dans certains cas limites ou la perspective est très « écrasée ». La fusion des segments ou des points peut alors produire des résultats inattendus. Nous envisageons d'inclure à ces filtres un principe de temporisation (évaluation paresseuse) afin de prendre en compte plus de traits et l'aspect global du dessin.
Le résultat après traitement est un graphe composé de points et segments 2D (figure 1c) contenant des informations sur la structure implicite du dessin (segments concourants, points incidents à des segments). Nous finissons actuellement les filtres de reconnaissance de formes qui permettront de communiquer au noyau de reconstruction 3D les données du dessin et les contraintes géométriques.

IMPLEMENTATION

Svalabard est implémenté en Java et utilise la boîte à outil MaggLite [21]. MaggLite est une boîte à outil graphique basée sur ICon [3]. ICon (Input Configurator) délivre un haut niveau de configurabilité et d'adaptabilité aux applications qui l'utilisent (connexion graphique des périphériques d'entrée du système aux outils ou interactions de l'application). MaggLite fournit un modèle de composants graphique et d'abstractions d'outils. Chaque composant graphique et outil de MaggLite décrit ses points d'entrée et de sortie d'interaction sous forme de module ICon, et l'on peut ainsi les connecter directement à des périphériques ou des interactions génériques décrites dans la librairie (pointage, poignées, reconnaissance vocale et gestuelle, toolglass, fisheye, etc.).

Le premier intérêt pour Svalabard est donc la possibilité d'associer facilement des périphériques aux outils afin de proposer une interaction proche d'une table à dessiner, ou chaque outil physique a une fonction logique. Ensuite, en développant les filtres de traitement sous forme de modules ICon, nous avons pu exploiter la configurabilité et la modularité de cette boîte à outil. Configurabilité parce qu'il est possible de configurer graphiquement les filtres (paramètres, ordre si leurs entrées/sorties le permettent). Mais aussi modularité car il est simple d'étendre les filtres existants ou d'en ajouter de nouveaux afin d'utiliser et configurer interactivement l'interface pour d'autres applications (dessin technique, conception d'interfaces, dessin artistique, etc.).

TRAVAUX FUTURS

Nous travaillons actuellement sur les filtres de détection des propriétés géométriques du dessin (reconnaissance de formes). Le but est d'identifier au mieux les volumes dessinés pour en extraire les propriétés et contraintes nécessaires à la reconstruction 3D. Nous orientons nos travaux vers des méthodes de reconnaissance de formes structurelles, adaptées à notre traitement des données. Le but est de décharger le plus possible l'utilisateur de la saisie de contraintes géométriques. Dans les cas ambigus ou indéterminés, nous utiliserons la détection du contexte et les interactions avancées de MaggLite afin que l'utilisateur puisse guider et valider les choix du système de manière non intrusive.

Nous projetons ensuite de valider notre interface et ses filtres d'analyse par une étude dans des conditions réelles d'utilisation. Les données de cette étude permettront aussi d'ajuster plus finement les paramètres de traitement et de les améliorer en y incorporant des méthodes d'apprentissage adaptatives. Nous étudions d'une part la possibilité d'utiliser une architecture en réseau de neurones pour la détection du contexte, et d'autre part l'adaptation dynamique des seuils des filtres au fur et à mesure de la pratique du dessin avec Svalabard.

Finalement, la connexion de Svalabard au noyau mathématique de reconstruction 3D permettra de proposer un système de modélisation 3D par le dessin fondé sur les méthodes de conception créatives.

CONCLUSION

L'étude que nous avons conduite sur les méthodes de conception et le dessin architectural nous a permis de concevoir un système de dessin pour la modélisation 3D plus proche des habitudes des utilisateurs créatifs. Svalabard se présente comme une métaphore de table à dessin, ou le dessinateur retrouve un environnement connu (papier, crayon, calques). Les filtres transparents de traitement du dessin et de détection du contexte permettent d'adapter cette interface à la phase de dessin, et de nettoyer et structurer celui-ci en temps réel. Le résultat sera alors exploitable pour des traitements de plus haut niveau afin d'obtenir automatiquement une scène 3D.

Notre démarche itérative dans le domaine de la modélisation 3D en architecture permet de valider les concepts de notre approche au fur et à mesure. Mais nous pouvons déjà considérer son application à la modélisation 3D en général, ou à l'analyse de volumes dessinés grâce à la modularité et l'extensibilité du système.

REMERCIEMENTS

Les auteurs remercient Pierre Dragicevic, Jean-Daniel Fekete et Pierre Leclercq pour leurs aides et conseils.

BIBLIOGRAPHIE

1. ARCHICAD, Abvent, Disponible à l'adresse http://www.abvent.com/fr/logiciels/archicad/.
2. Beaudouin-Lafon, M. Interaction Instrumentale : de la manipulation directe à la réalité augmentée. *Actes des Neuvièmes Journées sur l'Interaction Homme Machine, IHM'97*, Poitiers, Septembre 1997, Cépaduès-Editions.
3. Dragicevic, P., Fekete, J.D. Input Device Selection and Interaction Configuration with ICon. *In Proceedings of IHM-HCI 2001*, Blandford, A., Vanderdonckt, J., Gray, P., (Eds.): People and Computers XV - Interaction without Frontiers, Lille, France, Springer Verlag, pp. 543—448.
4. Eggli, L., Hsu C., Brüderlin B.D., Elber, G. Inferring 3D models from freehand sketches and constraints. *Computer-aided Design*, vol. 29, n° 2, pp 101—112, 1997.
5. Estevez, D. *Dessin d'architecture et infographie, l'évolution contemporaine des pratiques graphiques.* Janvier, 2001, CNRS Editions.
6. Goel, V. *Sketches of Thought.* The MIT Press, 1995.
7. Huot, S., Dumas, C, Hégron G. Toward Creative 3D Modeling: an Architects' Sketches Study. Short Paper, *In Proceedings of 9th IFIP TC13 International Conference on Human-Computer Interaction, INTERACT'03* (September 1-5 2003, Zurich, Switzerland), IOS Press, pp 785—788.
8. Igarashi, T., Matsuoka, S., Tanaka, H. Teddy : A sketching interface for 3D freeform design. *In Proceedings of SIGGRAPH99 Conference,* (08-13 Août, 1999, Los Angeles), ACM Press, pp 409—416.
9. Igarashi, T., Hughes, J.F. A suggestive interface for 3D drawing. *In 14th Annual Symposium on User Interface Software and Technology* (11-14 Novembre, 2001, Orlando), ACM UIST'01, pp 173—181.
10. Kuzo, P.M., Macé, P.,Lhomme, O. A constraint-based System for 3D Geometric Modelling. *In Proceedings of 3rd International Conference "Computer Graphics and Artificial Intelligence"*, Limoges, France, Avril 28—29, 1998.
11. Lamb, D., Bandopadhay A. Interpreting a 3D Object From a Rough 2D Line Drawing. *In Proceedings of Visualization '90*, 1990, pages 59—66.
12. Lebahar, J.C. *Le dessin d'architecte. Simulation graphique et réduction d'incertitude.* Editions Parenthèses, Collection Architecture/Outils, Presses Universitaires de France, 1983.
13. Leclercq P., Juchmes R. The Absent Interface in Design Engineering. I*n AIEDAM Special Issue: Human-computer Interaction in Engineering Contexts*, Cambridge University Press, USA, 2002.
14. Mamykina, L., Candy, L., Edmonds, E. Collaborative Creativity. *Communications of the ACM*, Vol. 45, No 10, Octobre 2002, pp. 96—99.
15. Pugh, D. Designing solid objects using interactive sketch interpretation. *Computer Graphics*, vol. 25, March 1992, pp 117—126.
16. Sezgin, T., Stahovic T., Davis R. Sketch based interfaces: Early processing for sketch understanding. *In Proceedings of 2001 Perceptive User Interfaces Workshop (PUI'01),* (15-16 Novembre, 2001, Orlando), ACM Digital Library.
17. Shpitalni, M., Lipson, H. Classification of Sketch Strokes and Corner Detection Using Conic Sections and Adaptive Clustering, *Trans. of ASME J. of Mechanical Design*, Vol. 119, No. 2, 1997, pp. 131—135.
18. Sosnov, A., Macé, P., Hégron, G., Huot, S. Rapid incremental architectural modeling from imprecise sketches and geometric constraints. *In Proceedings of Graphicon Conference*, Nizhny Novgorod, Russie, 2002.
19. Sosnov, A., Macé, P., Hégron, G. Semi-metric Formal 3D Reconstruction from Perspective Sketches. *In Proceedings of ICCS 2002 Conference – Part 2, Computer Graphics and Geometric Modeling Workshop* (21-24 Avril, 2002, Amsterdam), Springer, pp 285—294.
20. Suwa, M., Tversky, B. What do architects and students perceive in their sketches? A protocol analysis. *Design Studies*, vol. 18, n° 4, Elsevier Science Ltd. Octobre 1997, pp 385—403.
21. The MaggLite UI Toolkit Page. Disponible à l'adresse http://www.emn.fr/x-info/magglite.
22. Tolba, O., Dorsey, J., McMillan, L. A Projective Drawing System. *In Proceedings of 2001 Symposium on Interactive 3D Graphics (I3D 2001)*, (March 19-21, Research Triangle Park, North Carolina), ACM Press, pp 25—34.
23. Van Dam, A. The Human Connection: Post-WIMP User Interfaces. *Communications of the ACM*, vol. 40, n° 2, ACM Press, Février 1997, pp 63—67.
24. Zeleznik, R.C., Herndon K.P., Hughes J.F. SKETCH: An interface for sketching 3D scenes. *In Proceedings of SIGGRAPH'96 Conference,* (04-09 Août, 1996, New Orleans), Addison Wesley, pp 163—170.