

**THÈSE**

*Pour obtenir le grade de*

**DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE**

*Spécialité Informatique, Arrêté ministériel du 7 août 2006*

*Présentée par*

**Élisabeth ROUSSET**

*sous la direction de* François Bérard, Michaël Ortega, Laurence Nigay  
*préparée au sein de* Laboratoire d'Informatique de Grenoble  
*dans* École Doctorale Mathématiques, Sciences et  
Technologies de l'Information, Informatique

**PEINTURE NUMÉRIQUE SUR  
MODÈLES 3D EN USAGE RÉCRÉATIF**

*Thèse soutenue publiquement le 11 février 2016  
devant le jury composé de*

*Présidente :* **Pr. Marie-Paule Cani**  
*Professeur à l'Université de Grenoble*

*Rapporteurs :* **Pr. Géry Casiez**  
*Professeur à l'Université Lille 1*  
**Pr. Pascal Guitton**  
*Professeur à l'Université Bordeaux 1*

*Examineur :* **Mr. Jean-Baptiste de la Rivière**  
*Directeur R&D à Immersion SAS*

*Directeurs de thèse :* **Dr. François Bérard**  
*Maître de conférence à l'Université de Grenoble*  
**Dr. Michaël Ortega**  
*Ingénieur de recherche au CNRS*  
**Pr. Laurence Nigay**  
*Professeur à l'Université de Grenoble*





# Table des matières

<b>Table des Figures</b>	<b>v</b>
<b>Liste des Tableaux</b>	<b>vii</b>
<b>Introduction</b>	<b>1</b>
Motivations . . . . .	1
Contributions . . . . .	2
<b>I État de l’art</b>	<b>3</b>
<b><i>Introduction</i></b>	<b>5</b>
<b>1 Peinture numérique</b>	<b>7</b>
<i>Introduction</i> . . . . .	8
1.1 Origines . . . . .	10
1.2 Interaction . . . . .	11
1.3 Peinture 3D . . . . .	18
<i>Bilan</i> . . . . .	31
<b>2 Navigation 3D</b>	<b>33</b>
<i>Introduction</i> . . . . .	33
2.1 Navigation libre . . . . .	35
2.2 Navigation assistée . . . . .	45
<i>Bilan</i> . . . . .	49
<b>3 Peinture et Navigation</b>	<b>51</b>
<i>Introduction</i> . . . . .	51
3.1 Interaction bi-manuelle . . . . .	52
3.2 Choix des dispositifs d’entrée . . . . .	58
3.3 Optimisations . . . . .	64
<i>Bilan</i> . . . . .	67
<b><i>Bilan de l’état de l’Art</i></b>	<b>69</b>
<b>II Contributions</b>	<b>71</b>
<b><i>Introduction</i></b>	<b>73</b>

---

<b>4</b>	<b>Influence de l'aspect direct de l'interaction sur la peinture numérique</b>	<b>75</b>
	<i>Introduction</i> . . . . .	76
4.1	Évaluation expérimentale . . . . .	77
4.2	Résultats . . . . .	81
4.3	Discussion . . . . .	85
	<i>Bilan</i> . . . . .	87
<b>5</b>	<b>Interactions subjectives pour les rotations 3D</b>	<b>89</b>
	<i>Introduction</i> . . . . .	90
5.1	Le critère de surjection . . . . .	91
5.2	Nouvelles techniques de rotation 3D . . . . .	93
5.3	Analyse de critères . . . . .	98
5.4	Évaluation expérimentale . . . . .	99
5.5	Résultats . . . . .	103
5.6	Discussion . . . . .	107
	<i>Bilan</i> . . . . .	108
<b>6</b>	<b>Interactions bimanuelles et unimanuelles pour la peinture 3D</b>	<b>109</b>
	<i>Introduction</i> . . . . .	110
6.1	Choix des interactions . . . . .	110
6.2	Choix de la tâche . . . . .	113
6.3	Contraintes d'implémentation . . . . .	116
6.4	Évaluation expérimentale . . . . .	118
6.5	Résultats . . . . .	121
6.6	Discussion . . . . .	126
	<i>Bilan</i> . . . . .	129
	<b><i>Bilan des contributions</i></b>	<b>131</b>
	<b>Conclusion</b>	<b>133</b>
	Bilan général . . . . .	133
	Pistes de recherche . . . . .	134
<b>A</b>	<b>Revue technologique</b>	<b>137</b>
	<i>Introduction</i> . . . . .	138
A.1	Technologies tactiles . . . . .	138
A.2	Stylets numériques . . . . .	149
	<i>Bilan</i> . . . . .	155
	<b>Bibliographie</b>	<b>157</b>

# Table des figures

1.1	Exemples d'oeuvres numériques . . . . .	8
1.2	Exemples de brosses numériques . . . . .	9
1.3	Tracé numérique à la souris et au stylet . . . . .	12
1.4	Interaction au crayon et interaction au pinceau . . . . .	13
1.5	Interaction directe et indirecte au stylet . . . . .	14
1.6	Mesures de MacKenzie . . . . .	18
1.7	Structure standard d'un objet 3D . . . . .	19
1.8	Paramétrage d'un matériau à l'aide de textures . . . . .	21
1.9	Exemples de textures procédurales . . . . .	22
1.10	Déformation de la texture lors d'un dépliage automatique . . . . .	23
1.11	Dépliage manuel . . . . .	23
1.12	Processus de peinture 3D . . . . .	24
1.13	Projection painting . . . . .	29
1.14	Empreinte dynamique pour compenser la distorsion . . . . .	30
1.15	Simplification du workflow grâce à la peinture par projection . . . . .	30
2.1	Les degrés de liberté du contrôle de la caméra . . . . .	35
2.2	Translations 3D et gestes RST en tactile multipoint . . . . .	37
2.3	Arcball . . . . .	39
2.4	Sticky Tools . . . . .	41
2.5	RST et Two Axis Valuator . . . . .	42
2.6	Screenspace . . . . .	43
2.7	Point d'intérêt . . . . .	45
2.8	Point d'intérêt . . . . .	46
2.9	Scruticam . . . . .	46
2.10	HoverCam . . . . .	47
2.11	Automated Camera Control for Drawing . . . . .	48
2.12	Smooth HoverCam . . . . .	48
3.1	Rôles des cadres de référence kinesthésiques . . . . .	55
3.2	Dispositifs d'entrée à 6 degrés de liberté . . . . .	59
3.3	LaserPaint . . . . .	65
3.4	LayerPaint . . . . .	66
3.5	Automated Camera Control for Drawing . . . . .	66
4.1	Interaction directe vs. interaction indirecte . . . . .	77
4.2	La tâche . . . . .	78
4.3	Calcul de OE et OV . . . . .	81

4.4	Résultats quantitatifs . . . . .	82
4.5	Traces . . . . .	83
4.6	Résultats subjectifs . . . . .	84
5.1	Exemple d'interaction non surjective . . . . .	92
5.2	Interaction TAVZ . . . . .	94
5.3	Interaction TAV+ . . . . .	95
5.4	Interaction Arcball . . . . .	96
5.5	Interaction Arcball+ . . . . .	97
5.6	La tâche . . . . .	101
5.7	Hierarchie des techniques de rotations . . . . .	102
5.8	Résultats quantitatifs . . . . .	104
5.9	MDS . . . . .	105
5.10	Évaluation subjective . . . . .	107
6.1	Difficulté des zones . . . . .	115
6.2	Zones à peindre . . . . .	115
6.3	Carte UV du modèle de l'étude expérimentale . . . . .	116
6.4	Implémentation du <i>projection painting</i> . . . . .	117
6.5	Tâche . . . . .	119
6.6	Résultats des mesures de temps . . . . .	122
6.7	Résultats des mesures de nombres de traits . . . . .	124
6.8	Résultats des mesures de vitesse de remplissage . . . . .	125
6.9	Évaluation subjective . . . . .	126
A.1	Capacité projetée . . . . .	139
A.2	Auto-capacité et capacité mutuelle . . . . .	139
A.3	Contacts fantômes . . . . .	140
A.4	Capacité de surface . . . . .	141
A.5	Écran résistif analogique . . . . .	142
A.6	Écran résistif numérique . . . . .	142
A.7	Écran à ondes acoustiques de surface . . . . .	143
A.8	Ondes acoustiques de profondeur . . . . .	144
A.9	Écrans infrarouges . . . . .	145
A.10	Variante infrarouges . . . . .	145
A.11	Écrans à caméras . . . . .	146
A.12	Contacts fantômes sur les écrans à caméras . . . . .	147
A.13	Réflexion interne totale . . . . .	147
A.14	Planar Scatter Detection . . . . .	148
A.15	Force Sensing . . . . .	149
A.16	Sensel Morph . . . . .	149
A.17	Résonance électro-magnétique . . . . .	150
A.18	Stylets capacitifs . . . . .	151
A.19	Stylet optique . . . . .	153
A.20	Intupaint . . . . .	153
A.21	I/O brush . . . . .	154
A.22	Stylet magnétique . . . . .	154

# Liste des tableaux

1.1	Avantages et inconvénients des interactions directe et indirecte au stylet .	15
3.1	Dispositifs d'entrée à étudier pour la peinture numérique 3D . . . . .	62
3.2	Synthèse comparative des technologies de stylet numérique . . . . .	63
3.3	Synthèse comparative des technologies tactiles . . . . .	64
4.1	Effet de l'aspect direct de l'interaction . . . . .	85
4.2	Effet du gain . . . . .	86
5.1	Analyse de critères . . . . .	100
5.2	Résultats . . . . .	104
6.1	Dispositifs d'entrée à étudier pour la peinture numérique 3D . . . . .	111
A.1	Synthèse comparative des technologies tactiles et de stylets numériques . .	156





# Introduction

## Motivations

La création numérique 3D exerce actuellement un attrait important auprès du grand public. Elle est omniprésente dans notre environnement : que ce soit via le cinéma, les jeux vidéos ou encore la publicité, nous avons de nombreuses occasions de voir des images de synthèse et des effets spéciaux de scènes numériques 3D.

La relation du grand public aux images de synthèse 3D est restée passive pendant longtemps : l'utilisateur lambda était spectateur ; les moyens logiciels et matériels de création de contenu 3D étaient réservés aux graphistes professionnels. De nos jours, les avancées technologiques permettent à tout un chacun de disposer chez soi de matériels de qualité suffisante pour l'infographie 3D à des tarifs raisonnables. La disponibilité de ces moyens auprès du grand public a fait de l'infographie 3D un nouveau média d'expression artistique. On peut en effet observer le développement de communautés numériques d'artistes amateurs et professionnels dont l'objectif est la diffusion d'oeuvres et de connaissances sur ce média [1–4].

Cependant, la diffusion de l'infographie 3D comme media artistique auprès du grand public reste relativement limitée. En effet, même si les difficultés d'accès à la création numérique 3D ne sont plus matérielles ou financières, elles perdurent au niveau de l'interaction homme-machine : les logiciels d'infographie 3D offrent en général une interaction complexe. Ils nécessitent de longues périodes d'apprentissage qui sont justifiées dans le cadre professionnel ou pour des amateurs passionnés. Mais cet apprentissage n'est pas adapté à un usage récréatif où l'utilisateur cherche à se divertir et n'est pas prêt à fournir des efforts importants pour cela.

Un effort d'ouverture à un public plus large a cependant lieu depuis quelques années. Les grands éditeurs de logiciels professionnels ont publié des versions simplifiées et abordables à destination des graphistes amateurs. On trouve actuellement sur le marché des logiciels de différents niveaux de complexité. Certains logiciels sont faciles d'accès mais

les possibilités de création sont limitées [5]. D'autres logiciels sont plus complets mais nécessitent encore un apprentissage important [6–8].

Parmi les différentes activités du domaine de la création numérique 3D (modélisation des objets, coloration, mise en lumière, animation, etc.), la *peinture numérique* présente un intérêt particulier. En effet cette activité, qui consiste à définir l'apparence d'un objet une fois sa géométrie 3D définie, devient pour certaines personnes un hobby au même titre que la peinture traditionnelle pour d'autres. Nous pensons qu'elle peut être une *porte d'entrée* accessible et ludique à la création numérique 3D : si l'utilisateur novice est satisfait de son expérience de peinture de modèles, il est possible qu'il soit davantage enclin à faire les efforts nécessaires pour apprendre des interactions éventuellement plus complexes dans le but d'effectuer d'autres tâches de création numérique 3D.

Ce mémoire s'intéresse donc aux approches visant à rendre l'interaction de peinture numérique sur modèles 3D accessible à des utilisateurs novices dans le cadre d'un usage récréatif.

## Contributions

Les difficultés amenées par l'interaction de peinture numérique 3D apparaissent à différents niveaux : interaction avec le pinceau numérique, navigation dans la scène, et combinaison de la peinture et de la navigation.

Ce mémoire propose donc dans un premier temps un état de l'art des différentes techniques de peinture 3D et de navigation 3D ainsi que de leurs possibilités de combinaison. Nous passons également en revue les différents types de dispositifs existant pour la peinture 3D, dans un objectif d'usage récréatif pour utilisateurs novices.

Par la suite, nous proposons une étude de l'influence de l'aspect direct de l'interaction sur la peinture numérique 3D afin de déterminer la famille de dispositifs d'entrée la plus adaptée à cette tâche pour notre contexte.

Nous décrivons ensuite de nouvelles techniques de navigation 3D adaptées à notre tâche, et nous en étudions l'efficacité de manière qualitative et quantitative.

Enfin, nous proposons une étude des combinaisons d'interactions de navigation et de peinture 3D. Cette étude nous permet de mettre en évidence une interaction complète qui semble bien adapté à des utilisateurs novices.

Première partie

État de l'art



# Introduction

La peinture numérique sur modèle 3D est une interaction complexe qui comprend deux composantes distinctes : d'une part l'application de la couleur sur un objet et, d'autre part, la navigation autour de l'objet. La navigation permet de bien percevoir l'objet et d'atteindre toutes ses faces.

Un utilisateur novice peut donc rencontrer des difficultés à deux niveaux : (1) au niveau des interactions nécessaires à chaque composante de la tâche : peinture numérique et navigation ; (2) au niveau de l'utilisation conjointe de ces interactions. Pour que l'interaction de peinture numérique sur modèle 3D soit accessible à un usage novice, il faut que les interactions de peinture numérique, de navigation, et la manière de les combiner soient également adaptées à cet usage.

Nous allons dans un premier temps considérer l'interaction de peinture numérique. Nous nous intéresserons d'abord à l'interaction de peinture numérique en général, puis nous étudierons les spécificités de la peinture sur modèles 3D. Nous exposerons les difficultés posées par ce type d'interaction, puis nous effectuerons un état de l'art des différentes techniques existantes.

Dans un second temps, nous considérerons les différents types de navigation possibles dans le cadre d'une tâche de peinture 3D, en mettant en avant leurs forces et leurs faiblesses par rapport à cette tâche pour les utilisateurs novices.

Enfin, nous étudierons les différentes approches proposées pour combiner la peinture et la navigation, tant au niveau du choix des interactions que du choix des dispositifs d'entrée.



# Chapitre 1

## Peinture numérique

### Sommaire

---

<b>Introduction</b> . . . . .	<b>8</b>
<b>1.1 Origines</b> . . . . .	<b>10</b>
<b>1.2 Interaction</b> . . . . .	<b>11</b>
1.2.1 Le stylet comme dispositif d'entrée . . . . .	11
1.2.1.1 Avantages du stylet . . . . .	11
1.2.1.2 Différences de manipulation entre pinceau et stylo . . . . .	12
1.2.2 Aspect direct de l'interaction . . . . .	13
1.2.3 Coordination oeil-main . . . . .	15
1.2.4 Évaluation de l'interaction de peinture numérique . . . . .	16
<b>1.3 Peinture 3D</b> . . . . .	<b>18</b>
1.3.1 La peinture 3D dans son contexte . . . . .	19
1.3.1.1 Structure standard d'une scène 3D . . . . .	19
1.3.1.2 Place de la peinture numérique dans le workflow 3D . . . . .	19
1.3.1.3 Paramétrage des matériaux . . . . .	20
1.3.1.4 Les textures comme paramètres . . . . .	21
1.3.1.5 Peinture de texture . . . . .	23
Peinture sur canevas plan . . . . .	24
Peinture par projection sur modèle 3D . . . . .	25
1.3.2 Contraintes d'implémentation . . . . .	25
1.3.2.1 Paramétrage planaire . . . . .	26
1.3.2.2 Découpage . . . . .	26
1.3.2.3 Dépliage interactif . . . . .	27
1.3.2.4 Alternatives . . . . .	27
1.3.3 Peinture par projection . . . . .	27
1.3.4 Autres techniques de peinture . . . . .	28
<b>Bilan</b> . . . . .	<b>31</b>

---

## Introduction

La peinture numérique peut être définie différemment selon que l'on se place d'un point de vue artistique ou technique.

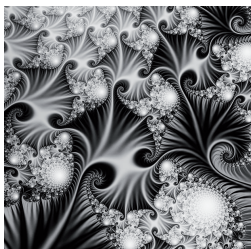
Du point de vue artistique, la peinture numérique est un sous-domaine de l'*art numérique*. L'art numérique est généralement défini comme une pratique artistique qui emploie des technologies numériques pour la majeure partie du processus créatif [9, 10]. Cette définition s'applique à différents types de médias comme l'audio et le visuel.

L'art numérique visuel peut prendre différentes formes selon le type de dispositif d'affichage utilisé (écran, réalité augmentée, environnements immersifs, etc.). La création d'images 2D affichables sur un écran ou imprimables est une des formes de l'art numérique visuel. Le site communautaire artistique DeviantArt propose une classification des sous-domaines de cette forme d'art [11], dont fait partie la peinture numérique :

**Imagerie assistée par ordinateur** Il s'agit de l'ensemble des méthodes de génération d'images basées sur des algorithmes, paramétrables ou non. On place par exemple dans cette catégorie la représentation visuelle de fractales (figure 1.1a).

**Photomanipulation** Cela consiste à transformer et mélanger une ou plusieurs images à l'aide d'outils numérique d'édition d'image (figure 1.1b).

a. *Imagerie assistée par ordinateur*



b. *Photomanipulation*



c. *Dessin vectoriel*



d. *Pixel Art*



e. *Peinture numérique*



f. *Art 3D*



FIGURE 1.1 – Exemples d'oeuvres numériques.



**Dessin vectoriel** Cette activité correspond à la création d'images numériques à partir de formes géométriques élémentaires et de courbes mathématiques. Ces images ont la particularité de ne pas être mémorisées sous forme de tableau de pixel, mais stockées sous forme uniquement mathématique (figure 1.1c). Les logos sont souvent des images vectorielles.

**Pixel Art** Cette technique consiste à créer une image numérique pixel à pixel. Elle a notamment été intensivement utilisée pour la création d'avatars et de décors dans les jeux vidéo des années 1990 (figure 1.1d).

**Peinture numérique** Cela consiste à créer une image numérique à la manière d'une peinture traditionnelle, à l'aide de "coups de pinceau" (*brush stroke*) (figure 1.1e). Les coups de pinceau sont généralement contrôlés par la main de l'utilisateur.

**Art 3D** Il s'agit de la création d'images numériques 2D à partir de rendus de scène 3D (figure 1.1f). Les scènes 3D elles mêmes sont également considérées comme des oeuvres artistiques à part entière.

**Animation** Il s'agit de la manipulation, mise en séquence et mise en mouvement d'images ou de modèles 3D à l'aide d'outils numériques, dans le but de réaliser des films d'animation, des effets spéciaux, etc.

Si l'on se place d'un point de vue technique, A.R. Smith définit la peinture numérique comme un processus de traitement de pixels tel que la fonction de traitement est dirigée par les mouvements de la main de l'artiste via un dispositif d'entrée [12]. Les fonctions de traitement utilisées en peinture numérique (appelées couramment *brosses*) peuvent aussi bien imiter le comportement d'outils traditionnels comme les plumes, les pinceaux, le fusain, etc. (figure 1.2a) que proposer de nouveaux comportements difficilement reproductibles en art traditionnel (figure 1.2b).



FIGURE 1.2 – Exemples de brosses numériques [13]

La peinture numérique a d'abord été développée comme une alternative à la peinture traditionnelle. Après une rapide présentation de l'apparition de cette forme d'art, nous nous intéressons à l'*interaction* de peinture numérique : bien que d'un point de vue extérieur la peinture numérique ressemble à la peinture traditionnelle, nous verrons que ces interactions présentent des différences majeures. Nous nous intéresserons ensuite au cas particulier de la peinture numérique sur modèle 3D, afin de mettre en avant ses problématiques spécifiques.

## 1.1 Origines

Les premiers logiciels de peinture numérique d'images 2D ont été mis au point en 1970. Le premier challenge était de parvenir à la génération d'une image en temps interactif avec les (faibles) moyens numériques disponibles. Cette année-là deux chercheurs, J. Miller et W.J. Kubitz, ont proposé indépendamment des logiciels de peinture dont la gamme de couleur était codée sur 3 bits (8 couleurs distinctes) [12, 14].

En 1973, D. Shoup a développé le premier système complet de peinture numérique sur 8 bits avant de le publier en 1979 sous le nom de *SuperPaint* [15]. Ce système comprend une partie logicielle mais également une partie matérielle adaptée : il s'agit du premier *frame buffer*. Dans ce système, le CPU calcule l'image à afficher et la stocke dans une mémoire vidéo. Cette mémoire est ensuite lue par le dispositif d'affichage tandis que le CPU calcule la prochaine image. Ce fonctionnement est l'ancêtre de celui des cartes graphiques actuelles. *SuperPaint* comprend un stylet comme dispositif d'entrée. Selon A.R. Smith, ce système est à l'origine des programmes de peinture numériques actuels [12]. Il est relativement complet, permettant l'usage simultané de 256 couleurs distinctes sélectionnables parmi 16.7 millions. Ce programme comprend la première implémentation du codage des couleurs sous la forme *teinte/saturation/valeur*, implémentation proposée par A.R. Smith afin de rapprocher la gestion des couleurs par l'utilisateur de celle de la peinture traditionnelle [12]. Ce système est également le premier à implémenter l'interaction de tracé au stylet telle que nous la connaissons actuellement : auparavant, les débuts et fin de traits étaient déterminés par un mouvement de contact (down/up) et le tracé par un mouvement de drag. Shoup propose de commencer le trait avec un contact down, d'effectuer le tracé par un drag puis de terminer le tracé par un up. Cette forme d'interaction permet de réduire le bruit au niveau des extrémités du tracé [12].

A.R. Smith a ensuite développé en 1977 *Paint3*, le premier système de peinture sur 24 bits. Ce système permettait d'utiliser 16.7 millions de couleurs simultanément, ce qui a autorisé un rendu correct de l'anti-aliasing. À l'occasion de la création de ce système, Smith propose le concept de quatrième canal de couleur, le *canal alpha*, pour coder la

transparence des couleurs. Cependant, ce concept ne sera pas intégré à *Paint3* et sera publié plus tardivement [12, 16].

T.Porter en 1982 réalise le premier système qui gère la transparence [17–19]. Il est codé sur 32 bits et permet l'usage simultané de 16.7 millions de couleurs sur 256 niveaux de transparence. Ce système a été développé pour le studio Lucasfilm dans le cadre de la réalisation du film *Star Trek II : The Wrath of Khan*. C'est le premier système de peinture numérique utilisé dans l'industrie du cinéma.

En 1991, les frères Knoll développent le premier logiciel commercial de peinture numérique. Ce logiciel est à l'origine codé sur 24 bits et est le premier à fonctionner sur un ordinateur grand public (Macintosh). Il est racheté par Adobe et est commercialisé sous le nom de Photoshop [12]. Ce logiciel est désormais la référence en matière d'édition d'image et de peinture numérique.

## 1.2 Interaction

### 1.2.1 Le stylet comme dispositif d'entrée

#### 1.2.1.1 Avantages du stylet

La peinture numérique est une activité particulière qui nécessite l'emploi d'un dispositif adapté. Il existe aujourd'hui un périphérique spécifiquement développée pour ce type de tâche, connue sous le nom de *stylet*. Ce dispositif est plus adapté que la souris pour la peinture numérique. On apprend en effet dès le plus jeune âge à utiliser les crayons, feutres, et stylos pour la création artistique. Le stylet, en reproduisant ce type d'interaction, permet donc d'envisager un transfert d'expérience plus important de nos connaissances acquises en interaction avec le monde physique que la souris.

Même si la souris est mise de plus en plus tôt dans les mains des jeunes enfants, elle fonctionne de façon relative avec un gain dynamique, ce qui ne semble pas adapté à la reproduction dans l'espace moteur d'une trajectoire définie mentalement par l'utilisateur. L'industriel Wacom a commercialisé des souris fonctionnant en mode absolu avec un gain fixe, ce qui permet de transférer plus facilement la trajectoire dans l'espace moteur. Cependant, le contrôle de la souris via les mouvements du poignet et du bras est moins précis que le contrôle du stylet via les mouvements des doigts [20].

Enfin, le stylet fournit plusieurs types de données simultanément, à savoir la position du curseur, la pression appliquée par l'utilisateur et l'inclinaison du corps du stylet. Ces différentes informations permettent d'obtenir un meilleur contrôle artistique qu'avec une

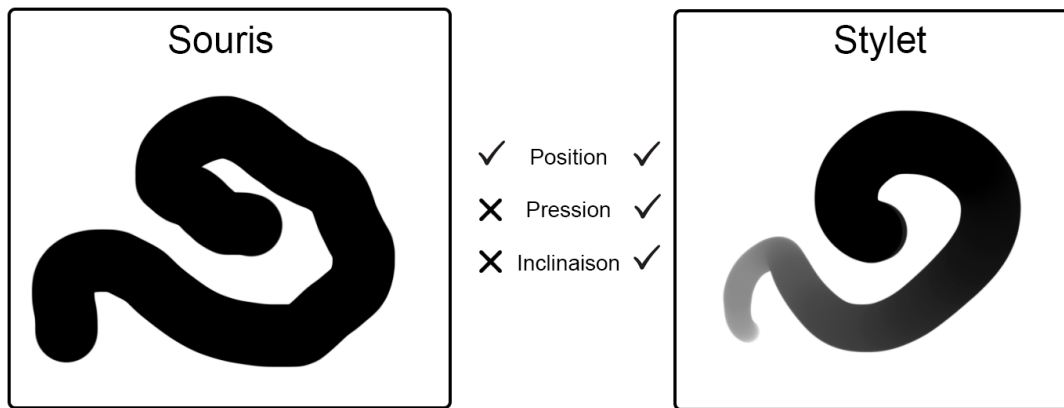


FIGURE 1.3 – Tracé numérique à la souris et au stylet

souris ne donnant que la position du curseur. Comme on peut le voir à la figure 1.3, l'association de la pression du stylet à l'opacité du trait et de l'inclinaison à la forme du pinceau permet davantage d'expressivité qu'un tracé simple réalisé à la souris.

### 1.2.1.2 Différences de manipulation entre pinceau et stylo

La peinture numérique se veut une adaptation de la peinture traditionnelle à l'outil informatique, où le stylet remplace le pinceau et l'écran la toile. Cependant l'interaction de peinture numérique est plus proche du dessin que de la peinture traditionnelle. En effet, le stylet offre une sensation de contact dure qui rappelle la mine de crayon ou de stylo, et non la souplesse des poils de pinceau. De plus, la peinture traditionnelle nécessite un temps de séchage qui impose à l'artiste de garder la main suspendue au dessus de la toile (figure 1.4). À l'inverse, en dessin traditionnel, l'artiste laisse reposer sa main sur la planche à dessin (ou sur un buvard), tout comme en peinture numérique où l'utilisateur pose sa main sur la surface d'acquisition du stylet (figure 1.4).

On peut constater d'autres différences d'interaction, bien que moins systématiques : la souplesse de la toile en peinture traditionnelle apporte un toucher plus dynamique que les supports rigides du dessin traditionnel et de la peinture numérique. Il faut cependant noter que la peinture traditionnelle peut également s'effectuer sur un support rigide comme le bois. On constate également une tendance à utiliser des supports dont l'inclinaison est plutôt verticale pour la peinture traditionnelle (chevalet) et plutôt horizontale pour le dessin traditionnel (table à dessin) et la peinture numérique. Enfin, la manière de tenir un stylo ou un pinceau et d'y positionner les doigts est généralement différente (figure 1.4). En cela, la tenue d'un stylet est de part sa forme plus proche de la tenue d'un stylo que de celle d'un pinceau.

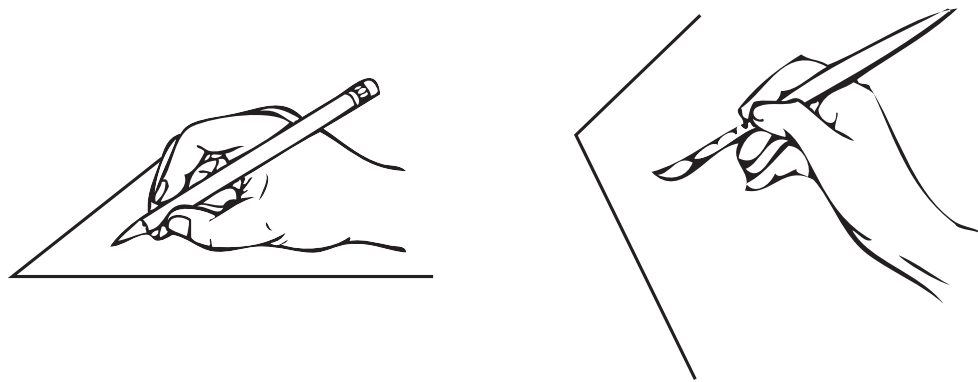


FIGURE 1.4 – Interaction au crayon et interaction au pinceau

Tout en gardant à l'esprit que la forme d'interaction impliquée par ce dispositif est plus proche du dessin traditionnel que de la peinture traditionnelle, nous allons maintenant nous concentrer sur l'interaction à l'aide du stylet. On parlera donc d'*interaction de peinture numérique* pour désigner l'interaction de peinture numérique au stylet.

### 1.2.2 Aspect direct de l'interaction

Le stylet peut être utilisé selon deux types d'interaction différents, à savoir *direct* ou *indirect*.

En interaction directe, la surface d'acquisition et la surface d'affichage sont confondues. L'encre virtuelle apparaît alors à la position de la pointe du stylet, à la manière d'un stylo sur une feuille de papier (figure 1.5a). Dans la suite de ce document, nous utiliserons le terme *écran interactif* pour faire référence à ce type de système.

En interaction indirecte, les surfaces d'acquisition et d'affichage sont dissociées. Le stylet est généralement utilisé à l'horizontale sur la surface d'acquisition tandis que le curseur est affiché sur un écran placé verticalement (figure 1.5b). Ce type de périphérique s'appelle *tablette graphique*.

Ces deux formes d'interaction présentent chacune des avantages et inconvénients qui sont résumés dans le tableau 1.1. L'interaction au stylet direct semble davantage similaire à l'interaction physique sur papier que l'interaction au stylet indirect. Cette plus grande similitude permet d'anticiper que les compétences acquises par l'utilisateur dans le monde physique sont plus facilement transférables au stylet direct qu'au stylet indirect. En effet, Bérard et Rochet-Capellan ont montré qu'une interaction impliquant des gestes similaires aux gestes physique, l'interaction au touché direct, permet un meilleur transfert de compétence comparé au touché indirect et à la souris [21].

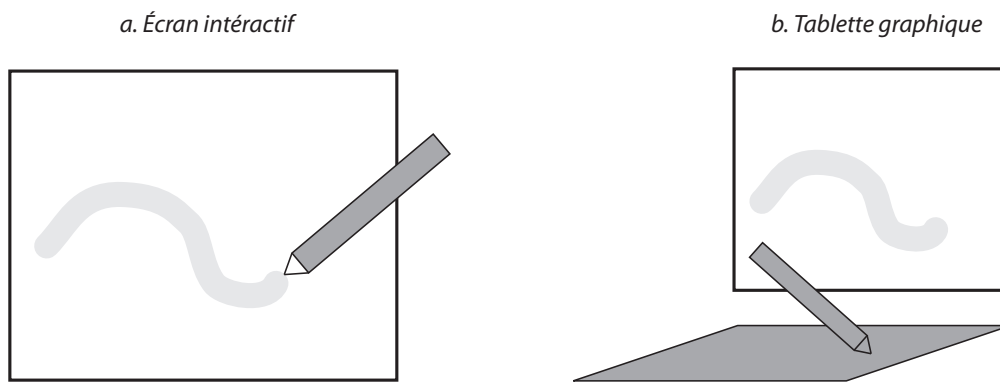


FIGURE 1.5 – Interaction directe et indirecte au stylet

Cependant, le stylet direct souffre de trois problèmes. Tout d'abord, la présence de parallaxe due à l'épaisseur de l'écran : le curseur est décalé et n'apparaît pas directement sous la pointe du stylet. Ensuite, le bras et la main de l'utilisateur occultent une large partie de l'écran. Troisièmement, les espaces moteur et visuel sont superposés, ce qui implique qu'ils ne peuvent pas être réglés indépendamment selon l'orientation la plus confortable pour l'utilisateur. En effet, l'espace moteur devrait être approximativement horizontal afin que la main de l'utilisateur puisse reposer sur sa surface, alors que l'espace visuel devrait être plutôt vertical dans le but de réduire les tensions cervicales. Enfin, Deber et al. montrent que la latence est davantage perceptible en interaction directe qu'en interaction indirecte [22].

À l'inverse, le stylet indirect permet de disposer les espaces moteur et visuel dans leurs orientations idéales. L'aspect indirect supprime également les problèmes de parallaxe et d'occultation et permet l'emploi d'un *gain* entre le contrôle et l'affichage (*control/display gain*) afin de réduire les efforts physiques nécessaires au déplacement du stylet. Cependant, cela conduit à une transformation complexe entre les espaces moteur et visuel. L'assimilation de cette transformation peut être difficile pour les utilisateurs et gêner leur capacité à effectuer les tâches de peinture. En effet, des commentaires d'utilisateurs novices et experts indiquent que les stylets indirects sont plus difficiles à appréhender que les stylets directs [23, 24], et que cette difficulté conduit à une moins bonne qualité du dessin. L'industriel Wacom, acteur majeur du marché des stylets directs et indirects, propose des exercices d'entraînement pour stylet indirect afin d'améliorer la coordination oeil-main et réduire ces problèmes [25].

Au delà des différences apparentes résumées dans le tableau 1.1, des études quantitatives objectives sont nécessaires à une meilleure compréhension des effets de l'aspect direct sur l'interaction, et particulièrement dans le cas d'utilisateurs novices qui ont moins de pratique et d'entraînement que les professionnels. L'effet de l'aspect direct a été étudié

	<b>Direct</b>	<b>Indirect</b>
<b>Avantages</b>	Similarité avec l'interaction physique	Plus confortable Pas de parallaxe Pas d'occultation Contrôle du gain Latence moins perceptible
<b>Inconvénients</b>	Parallaxe Occultation Moins confortable Latence plus perceptible	Transformation entre espaces moteur et visuel

TABLE 1.1 – Avantages et inconvénients des interactions directe et indirecte au stylet

du point de vue de la tâche de pointage, que ce soit à la souris ou au stylet [26–29]. Mais la tâche de peinture présente des requis plus importants que la tâche de pointage, comme nous l'expliquons ci-dessous.

Lorsqu'un utilisateur pointe une cible à l'écran, la trajectoire du stylet importe peu et peut être ajustée jusqu'à ce que la pointe du stylet atteigne la cible. Ce n'est pas le cas dans une tâche de dessin car chaque ajustement sera visible sur le trait final. Par conséquent, le dessin nécessite que l'utilisateur déplace le curseur *dans la bonne direction* à tout instant. Ceci implique que l'utilisateur doit être conscient en permanence (1) de la position du curseur et (2) de la direction du mouvement du curseur.

Dans le cas de l'interaction directe, (1) est fourni par la position de la pointe du stylet tandis que (2) correspond au mouvement prévu par l'utilisateur (car les espaces moteur et visuel sont dans le même espace de référence). Dans le cas de l'interaction indirecte, (1) est fourni par l'affichage du curseur pendant le survol de la tablette, mais (2) n'est pas disponible car les espaces moteur et visuel sont distincts : (2) peut seulement être approximativement déduit des positions précédentes du curseur.

Afin de mieux cerner l'influence de l'association et dissociation des espaces moteur et visuel, il est important de considérer les études portant sur la coordination oeil-main dans le domaine des neurosciences.

### 1.2.3 Coordination oeil-main

Selon Vercher et al., la proprioception du bras a un rôle important dans la calibration de la relation temporelle entre la main et le mouvement des yeux [30]. Les auteurs supposent que la dissociation des espaces moteurs et visuels peut amener à une mauvaise synchronisation des yeux et de la main à cause du changement d'espace de référence, ce qui est le cas en interaction indirecte.

Ketcham et al. observent une corrélation entre les mouvements des yeux et de la main [31] : ils étudient l'évolution de ces derniers lors de gestes complexes et montrent que les yeux restent fixés sur certains emplacements de la trajectoire de la main, qui correspondent aux maxima de courbure. Les yeux précèdent la main tout au long du mouvement, et les changements d'emplacement ont lieu par saccades lorsque la main atteint une vitesse minimale.

Plus récemment, Gowen et al. ont étudié les mouvement des yeux pendant une tâche de décalquage (c'est-à-dire la reproduction directe d'un dessin par dessus son modèle) et une tâche de dessin (c'est à dire sans modèle) [32]. Ils montrent que pendant un décalquage, les yeux restent fixés légèrement en avant de la pointe du stylo et avancent par petites saccades. Ils intitulent ce comportement "*close pursuit*" car la pointe du stylo suit de près la trajectoire des yeux. Le dessin implique un comportement différent, nommé "*target locking*" : les fixations des yeux ont lieu loin en avant de la pointe du stylo et restent stables jusqu'à ce que la pointe atteigne le voisinage de la fixation. Il y a moins de fixations dans ce comportement mais elles sont plus longues, et les changements d'emplacement ont lieu via de grandes saccades oculaires. Le comportement de target locking est particulièrement intéressant dans le cadre de cette thèse car il correspond à la tâche de peinture numérique (on a vu précédemment que les interactions de peinture numérique et dessin traditionnel sont similaires).

Tchalenko étudie les mouvements des yeux pendant une tâche de dessin élémentaire et minimale [33]. Il montre que le comportement de target locking est provoqué lorsque les participants doivent dessiner une ligne droite depuis une origine affichée vers une cible non-affichée. L'emplacement de la cible est deviné par le participant à l'aide d'autres éléments présents sur la feuille, comme par exemple l'intersection des prolongations de deux segments. Dans cette étude, Tchalenko ne trouve pas de différence significative entre les comportements de novices et de graphistes professionnels. De plus, il découvre que la stratégie des mouvements de yeux est similaire pour le dessin de lignes droites et courbes. Ce résultat est confirmé par Reina et al. qui montrent que le mouvement des yeux pendant le dessin d'une ligne courbe est une itération de target locking aux emplacements de maximum de courbure [34]. Ceci permet d'espérer que des études concernant le dessin de lignes droites puissent se généraliser aux lignes courbes, bien que les résultats doivent être confirmés par des expérimentations indépendantes.

#### 1.2.4 Évaluation de l'interaction de peinture numérique

L'évaluation de l'interaction de dessin ou de peinture est difficile car cela fait appel à des critères subjectifs. Plusieurs travaux proposent des évaluations qualitatives de techniques



de peintures à l'aide de questionnaires [35, 36]. Cependant ces évaluations ne permettent pas de quantifier objectivement les améliorations apportées.

Badam et al proposent de noter la “qualité perçue” d'un dessin [37] : pour cela, ils présentent plusieurs dessins simultanément aux participants, en demandant de sélectionner celui de meilleure qualité. Le score de qualité correspond alors au ratio du nombre de sélections sur le nombre total d'affichages. Ce type d'évaluation nécessite cependant un grand nombre de participants pour obtenir des résultats fiables : Badam et al. ont eu besoin de 277 participants pour évaluer 84 dessins.

Dixon et al utilisent des algorithmes de reconnaissance de visage pour évaluer la qualité de reproduction de portraits [38]. Cependant, cette forme d'évaluation reste spécifique au dessin de visages.

Une autre possibilité pour quantifier la qualité d'une interaction de dessin est de quantifier la différence entre les tracés obtenus et les tracés prévus par l'expérimentateur. Cela ramène le problème à une analyse de trajectoire.

Accot et al proposent une extension de la loi de Fitts pour des tâches impliquant des trajectoires de curseur à l'intérieur de “tunnels” [39]. Cependant, cette modélisation permet essentiellement de prévoir le temps de réalisation du tracé mais ne donne pas d'indication forte sur la qualité du tracé.

D'autres travaux proposent des métriques permettant de mesurer la ressemblance entre deux courbes. Certaines utilisent les distances comme celles de Fréchet [40] ou de Hausdorff [41], ou sont personnalisées en fonction de la tâche [42, 43]. Ces mesures prennent en compte la position et l'orientation de la trajectoire mais ne permettent pas d'isoler ces deux composantes afin de les analyser indépendamment. Ceci peut poser problème si l'on cherche à évaluer l'effet de l'interaction indirecte sur le dessin. En effet, une mesure exclusivement basée sur les orientations permettra de mettre en évidence les erreurs d'inférences sur la direction du tracé, et une métrique exclusivement basée sur les positions permettra de mettre en évidence les erreurs de décalage de la trace. En interaction directe, le phénomène de parallaxe à lui seul peut provoquer un tel décalage. Il est donc important d'utiliser des métriques permettant de dissocier l'effet de parallaxe des erreurs de direction.

Mackenzie et al introduisent six mesures de qualité de trajectoire [44]. Quatre sont uniquement basées sur la position et deux sur l'orientation, ce qui permet une analyse indépendante. Les deux métriques basées sur l'orientation permettent de déterminer le nombre de “*mouvement direction change*” et “*orthogonal direction change*” comme définis à la figure 1.6.

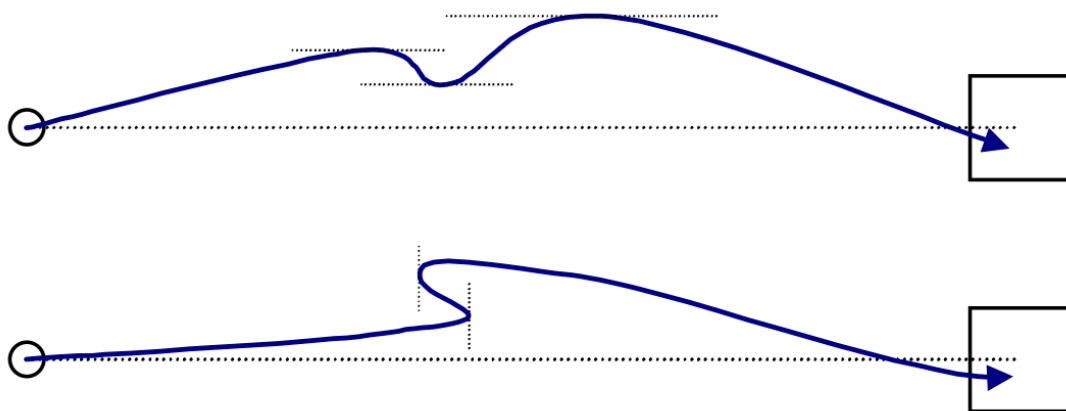


FIGURE 1.6 – Mesures de MacKenzie [44] : en haut, les *Mouvement Direction Changes* et en bas, les *Orthogonal Direction Changes*

Keefe et al. proposent également une mesure basée sur l'orientation qui permet de calculer la différence de direction entre deux courbes [45]. Cette mesure consiste à calculer l'angle moyen entre les tangentes des deux courbes aux points d'échantillonnage correspondants.

À notre connaissance, aucune de ces métriques n'a encore été employée pour l'évaluation d'une interaction de dessin en 2D. Nous allons définir dans le chapitre 4 des métriques spécifiques pour évaluer la qualité de tracés en terme de direction, indépendamment du problème de parallaxe, basées sur les métriques de MacKenzie [44] et de Keefe [45], et adaptées à notre tâche. Ces métriques représenteront donc une nouvelle forme de mesure quantitative de la qualité du tracé pour évaluer les interactions de dessin.

### 1.3 Peinture 3D

La peinture numérique sur modèle 3D nécessite une transformation entre l'espace bidimensionnel du dispositif d'entrée et l'espace tridimensionnel dans lequel l'utilisateur peint. En cela, elle augmente le degré de difficulté de la tâche comparée à la peinture numérique d'image 2D. Cette difficulté est impliquée par différents facteurs que nous allons présenter ici. Pour les comprendre, nous allons dans un premier temps nous intéresser au contexte d'utilisation de la peinture numérique sur modèle 3D, ce qui nous donnera un aperçu de ses différentes problématiques. Après avoir détaillé les travaux de recherche concernant une contrainte technique majeure, le *dépliage UV*, nous présenterons l'interaction de peinture numérique sur modèle 3D la plus répandue, à savoir la *peinture par projection*. Nous détaillerons ensuite les autres techniques existantes avant de conclure ce chapitre par une synthèse.

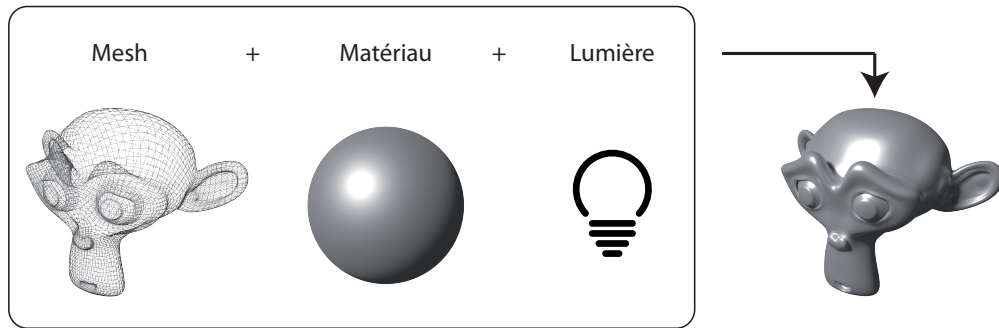


FIGURE 1.7 – Structure standard d'un objet 3D

Pour plus de simplicité, nous désignerons dans la suite de ce document la peinture numérique d'image 2D par *peinture numérique* et la peinture numérique sur modèle 3D par *peinture 3D*.

### 1.3.1 La peinture 3D dans son contexte

#### 1.3.1.1 Structure standard d'une scène 3D

Les scènes numériques 3D sont dans leur grande majorité décomposées en deux types d'information : d'une part la géométrie sous-jacente, et d'autre part l'apparence qui lui est appliquée. La forme courante d'une scène 3D est présentée à la figure 1.7. Comme on peut le voir, elle est constituée avant tout d'un *mesh* qui définit la forme de l'objet en 3D. Pour cela, le mesh est composé de sommets et d'arêtes. L'apparence est alors principalement déterminées par deux types d'éléments : un ou plusieurs *matériaux* qui définissent le comportement de la lumière relativement à l'objet ; des sources lumineuses qui permettent de visualiser l'objet et son relief en générant des effets d'ombre et de reflets sur celui-ci.

#### 1.3.1.2 Place de la peinture numérique dans le workflow 3D

La création de contenu numérique 3D peut s'effectuer suivant divers workflows, cependant ils ont en commun les étapes suivantes :

**Modelage :** C'est la création du mesh de l'objet. Il existe plusieurs techniques pour modeler un mesh. Le *polygonal modeling* consiste à éditer directement les positions des sommets, arêtes et faces du mesh [46]. Le *Surface modeling* utilise des points de contrôle externes au mesh pour définir la forme de celui-ci, à la manière des

points de contrôles des courbes de Bézier. Le *Meta modeling* permet de croquer rapidement des volumes à l'aide de formes primitives (cubes, boules, etc.) qui seront ensuite affinés avec les techniques précédentes. Il est également possible de sculpter un mesh de manière similaire à la sculpture traditionnelle [47].

**Éclairage :** C'est ce qui permet de "voir" la scène 3D. Pour cela, l'utilisateur place différentes sources de lumières dans la scène, de manière à la mettre en valeur et à créer une ambiance visuelle appropriée [48]. Les sources lumineuses apportent des effets d'ombres et de reflets qui fournissent des indices forts pour la perception 3D.

**Matériaux :** Ils permettent de définir les qualités artistiques et physiques de l'objet en contrôlant la manière dont la lumière interagit avec ce dernier [46]. C'est à cette étape que l'on définit les paramètres de couleur, diffusion, réflexion, transparence, réfraction, etc. qui vont donner à l'objet son apparence spécifique (métal, verre, plastique, etc.)

**Animation :** C'est l'étape qui donne vie à la scène 3D. Elle permet de faire bouger les objets et de les modifier au cours du temps [49]. Cette étape peut aussi être présente dans le cas de la production d'illustrations fixes. En effet, elle permet l'utilisation de simulations qui peuvent accélérer le processus de modelage. Par exemple, le modelage d'un drapé est difficile. À la place, on peut modeler un simple rectangle auquel on associe une simulation de tissus. Une fois l'animation de la simulation calculée, il suffit de sélectionner un instant précis et de récupérer la forme du mesh calculée.

**Rendu :** C'est la génération d'une image ou d'une vidéo en 2D à partir de la scène 3D. Cette étape nécessite des calculs très lourds. Un bon rendu est un compromis entre une bonne qualité d'image et un temps de calcul acceptable [50].

**Composition :** Cette étape permet de combiner les différentes images et vidéos créées lors du rendu pour en faire une production finale. La composition permet également d'améliorer et de modifier séparément chaque élément de la production [51].

Comme on peut le voir en examinant ces différentes étapes, le contrôle des qualités artistiques et physiques d'un objet a lieu au niveau des matériaux. On va donc s'intéresser au paramétrage de ces derniers, car c'est là qu'intervient le processus de peinture numérique.

### 1.3.1.3 Paramétrage des matériaux

Un matériau est une fonction définissant le comportement de la lumière sur l'objet auquel il est appliqué. Généralement, dans les logiciels de modélisation l'utilisateur n'a pas

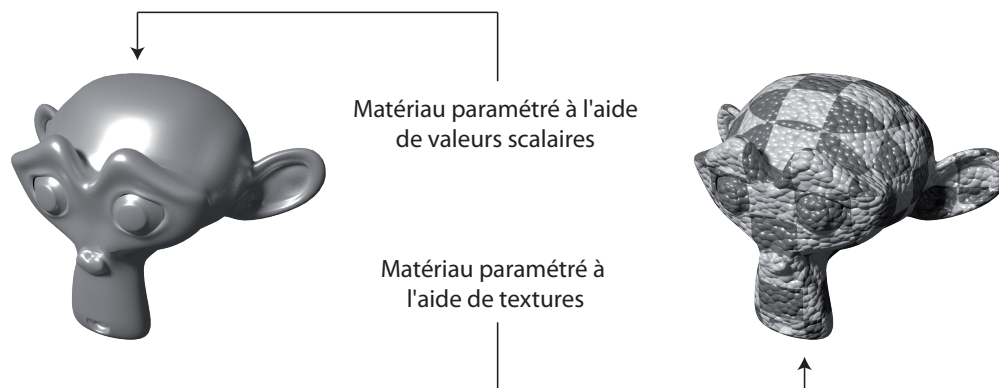


FIGURE 1.8 – Paramétrage d'un matériau à l'aide de textures

directement accès à la programmation de matériaux, car cela nécessite des connaissances avancées en mathématique et en optique. En revanche, l'utilisateur dispose de matériaux de base qu'il peut combiner et paramétrer de manière poussée pour obtenir les qualités artistiques souhaitées. Certains matériaux de base permettent par exemple de contrôler la couleur diffuse, la spécularité, la réflexion, l'émission de lumière, et l'orientation des normales du mesh.

Un matériau peut être paramétré à l'aide de valeurs *scalaires*, c'est à dire de valeurs appliquées uniformément au mesh de l'objet. Par exemple, un objet rouge avec des reflets bleus peut être paramétré à l'aide de la valeur scalaire "rouge" pour la couleur diffuse et de la valeur scalaire "bleue" pour la spécularité.

On peut également paramétrer un matériau à l'aide de valeurs *multidimensionnelles* (vecteurs ou matrices). On parle alors de *textures*. Ceci permet d'appliquer des valeurs distinctes aux différentes parties du mesh. Par exemple, à la figure 1.8, le même matériau est représenté paramétré à l'aide de valeurs scalaires et à l'aide de textures. À gauche, la couleur et les reflets sont uniformes, tandis qu'à droite, une texture de damier est appliquée à la couleur diffuse et une texture de bruit aléatoire est appliquée aux normales. Cette dernière texture simule une impression de relief. L'utilisation de textures permet donc d'apporter du détail au matériau, sans avoir à modifier la résolution du mesh sous-jacent [52, 53]

#### 1.3.1.4 Les textures comme paramètres

Heckbert définit une texture comme une image multidimensionnelle associée à un espace multidimensionnel [53]. Cette association, communément appelée *mapping*, est une fonction qui indique pour chaque point de l'espace ses coordonnées correspondantes dans la

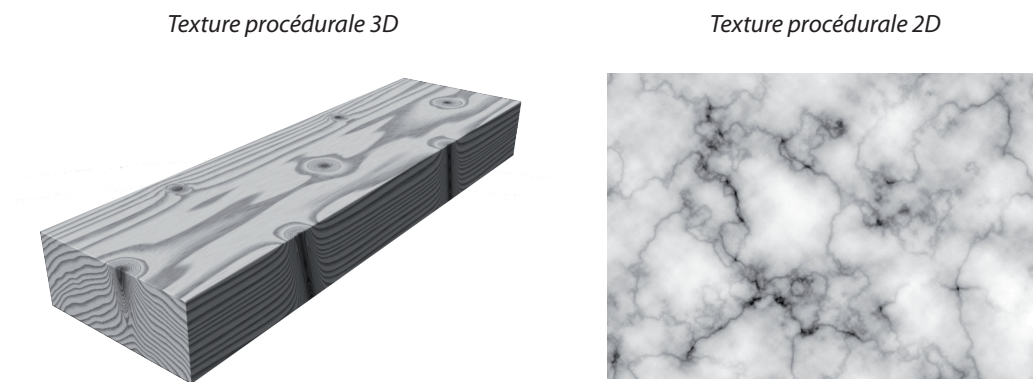


FIGURE 1.9 – Exemples de textures procédurales

texture [54]. Les standards industriels comprennent deux types de textures, à savoir les textures 2D et 3D.

Une texture 3D reste continue quelle que soit la forme du modèle auquel elle est appliquée. Comme on peut le voir à la figure 1.9 la texture "bois" (qui paramètre la couleur du modèle) s'adapte parfaitement à la forme de la planche, donnant l'impression que celle-ci a été réellement découpée dans ce matériau. Pour arriver à un tel résultat, une texture 3D peut être générée de façon procédurale : elle correspond alors à l'évaluation d'une fonction mathématique dans l'espace tridimensionnel, en chaque point de la surface du modèle [55]. On peut également obtenir une texture 3D à partir du monde réel, sous la forme d'une image tridimensionnelle "photographiée" à l'aide d'un dispositif de type scanner. Les textures 3D permettent d'obtenir des résultats d'excellente qualité, cependant elles n'offrent pas la liberté d'expression des textures 2D de part leur nature procédurale ou fixe : il est difficile d'éditer manuellement une texture 3D pour l'adapter au rendu souhaité.

Une texture 2D est une image associée à un modèle 3D de manière à ce que chaque face du mesh soit en correspondance avec une portion de la texture. Le mapping, appelé dans ce cas *dépliage*, consiste à faire correspondre un couple de coordonnées 2D de la texture à chaque sommet du mesh. Ce processus peut être effectué manuellement ou automatiquement.

Il existe de nombreuses techniques de dépliage automatique basées sur des algorithmes de projection [56]. Ces techniques permettent de déplier des formes primitives telles que les cubes, les cônes, les cylindres, etc. Cependant, leur utilisation dans le cadre de formes plus complexes conduit à des problèmes de distorsion. À la figure 1.10, une texture damier est appliquée aux deux modèles. On peut voir l'absence de déformation sur le cube, alors que le damier est complètement déformé sur la figure du singe.

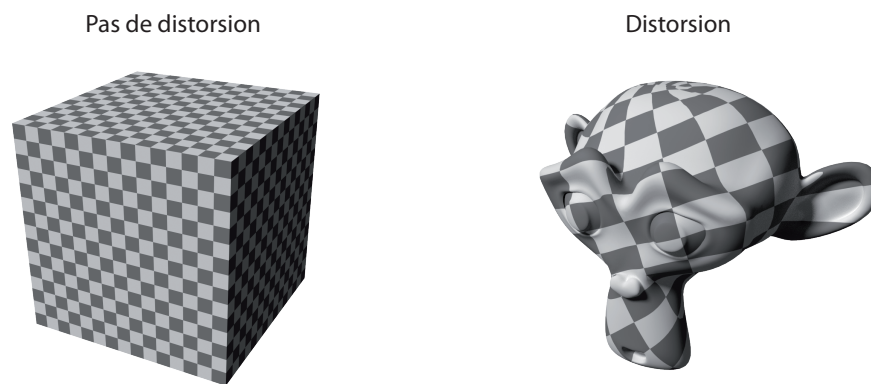


FIGURE 1.10 – Déformation de la texture lors d'un dépliage automatique

Le dépliage manuel permet dans une certaine mesure de remédier au problème de distorsion car il est plus flexible et offre un meilleur contrôle artistique. On parle souvent de *dépliage UV*, car il s'agit de faire correspondre explicitement les coordonnées 3D ( $x,y,z$ ) des sommets du mesh avec leur coordonnées 2D ( $u,v$ ) dans la texture. Pour cela, l'utilisateur marque certaines arêtes du mesh comme étant des *coutures*. Le mesh est alors découpé le long de celles-ci puis aplati. La correspondance obtenue est appelée *carte UV*. L'utilisateur peut ensuite la modifier afin de minimiser les déformations de la texture (figure 1.11).

### 1.3.1.5 Peinture de texture

Une fois le modèle déplié, l'utilisateur a la possibilité d'éditer la texture. Il est alors possible de la générer de façon procédurale, comme pour les textures 3D. Ce type de texture est en général combiné avec un dépliage automatique. Mais une autre possibilité,

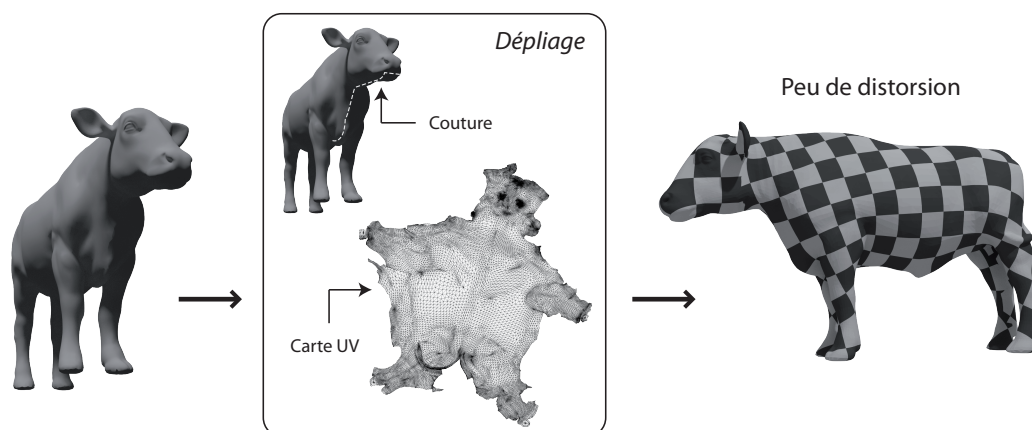


FIGURE 1.11 – Dépliage manuel

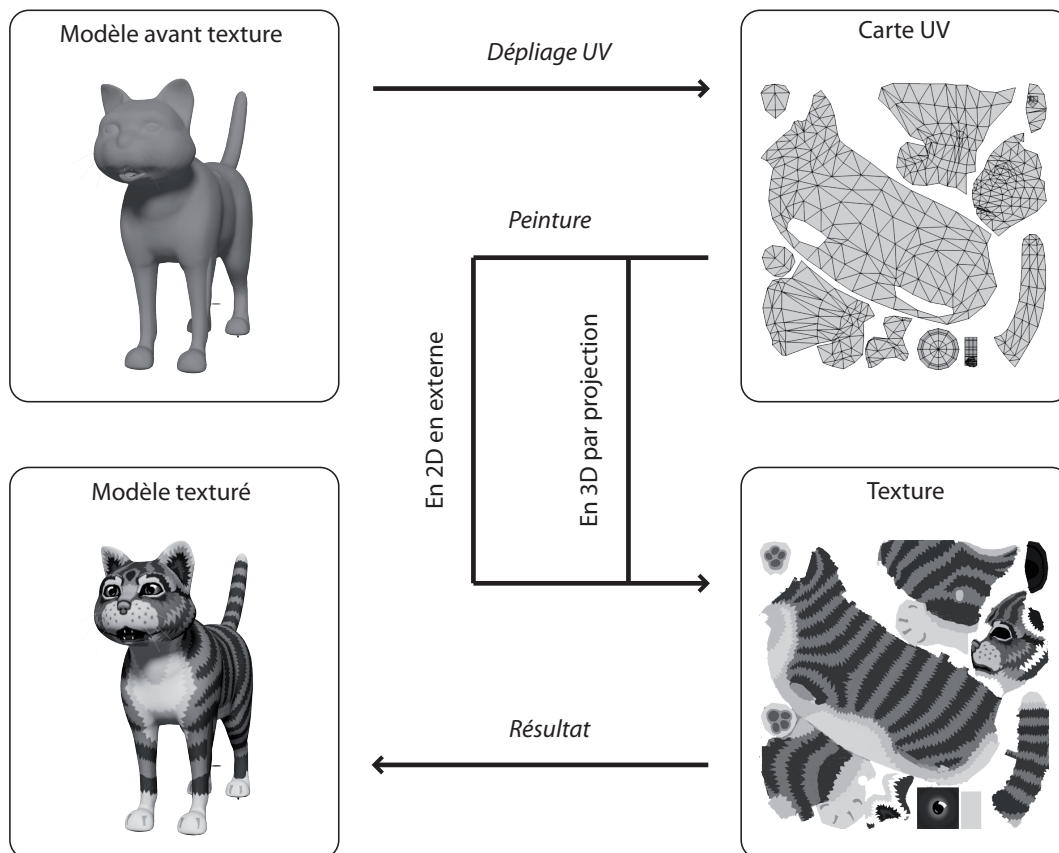


FIGURE 1.12 – Processus de peinture 3D

permettant une plus grande expressivité de l'artiste, est de peindre manuellement la texture. C'est également indispensable lorsqu'il y a besoin de détails précis et spécifiques. La texture nécessite alors un dépliage manuel (figure 1.12), afin qu'il soit "compréhensible", et que les coutures gênent le moins possible.

La peinture d'une texture 2D peut être effectuée suivant deux moyens distincts, à savoir la peinture en 2D à l'aide d'un canevas plan ou la peinture en 3D par projection sur le modèle (figure 1.12).

**Peinture sur canevas plan** L'utilisateur peut éditer la texture directement en 2D à l'aide d'un logiciel de traitement d'image. Cette technique est assez répandue car elle permet d'exploiter la puissance des fonctionnalités d'un tel logiciel, et par là d'obtenir des textures de grande qualité. Cependant, cette méthode implique une lourde charge cognitive pour l'utilisateur car il doit faire lui-même la projection de tête pour se repérer dans la carte UV. Cela nécessite un entraînement et une habitude difficilement accessibles aux graphistes non-professionnels.



**Peinture par projection sur modèle 3D** Cette méthode consiste à peindre directement dans la scène 3D. Dans ce cas c'est le logiciel qui est chargé de faire la projection dans la texture. Cette technique est moins lourde cognitivement, cependant elle implique une interaction de navigation 3D pendant la tâche de peinture. Or, nous verrons dans le chapitre 2 que la navigation 3D est une interaction complexe en général, qui peut distraire l'utilisateur de sa tâche principale.

Concernant les utilisateurs novices, la peinture par projection présente l'avantage d'éviter la phase difficile d'apprentissage de projection mentale imposé par la peinture en 2D. Cependant, il faut que l'interaction de navigation 3D associée soit également accessible aux utilisateurs novices, et suffisamment efficace pour permettre l'accès à toutes les zones d'une topologie complexe.

Quelle que soit la technique de peinture utilisée, il est auparavant nécessaire d'effectuer le dépliage de la texture. Cette étape, même si elle peut être effectuée partiellement à l'aide d'algorithmes de dépliage automatiques, nécessite tout de même une phase d'ajustement manuel afin que le rendu final corresponde aux attentes de l'utilisateur en terme de distorsion et de résolution. Cette étape représente une contrainte technique d'implémentation qui, de part sa difficulté, a une forte influence sur l'accessibilité de la peinture 3D à un public novice.

Nous allons donc nous intéresser dans la suite aux avancées scientifiques concernant d'une part le dépliage UV et d'autre part la peinture par projection.

### 1.3.2 Contraintes d'implémentation

Comme on l'a vu précédemment les textures sont utilisées pour paramétrer les matériaux. À l'origine elles servaient essentiellement à apporter du détail concernant la couleur des objets [53, 56, 57]. Leur utilisation a ensuite été étendue à d'autres notions telles que la réflexion [52], la spécularité [58], la transparence [59] ou encore la modification des normales pour simuler des détails de relief (*bump mapping*) [60].

Cependant les textures 2D posent un problème au niveau du dépliage UV. En effet, il n'existe aucune correspondance entre les coordonnées 2D d'une texture et celles d'une sphère qui soit continue et telle que chaque point soit assigné à un unique couple de coordonnées. Plus précisément, le théorème de Borsuk-Ulam affirme que les points antipodaux sont associés aux mêmes coordonnées planaires [61]. Ce problème se retrouve dans de nombreuses autres surfaces closes. Il en résulte que de manière générale, il n'existe pas de correspondance naturelle entre une texture 2D et une surface 3D, et l'image subit nécessairement un phénomène de distorsion [53]. Ce phénomène est très

bien illustré par le problème du planisphère : quelle que soit la projection utilisée, aucune carte du monde ne représente fidèlement les proportions réelles des différents continents.

On a vu cependant que cette distorsion peut être réduite avec l'ajout de coutures, mais celles-ci introduisent des discontinuités dans la fonction de dépliage. L'objectif d'un dépliage UV est donc d'obtenir un équilibre entre la minimisation de la distorsion et la minimisation des discontinuités, en combinant les techniques de paramétrage planaire et de découpage.

### 1.3.2.1 Paramétrage planaire

De nombreuses études ont été menées sur le paramétrage planaire d'objets 3D. Certaines ne préservent aucune caractéristique du mesh de départ ce qui amène à de fortes distorsions de la texture [52, 57, 62, 63]. D'autres tentent de minimiser les distorsions en conservant les angles [64–73], les distances [74, 75], ou les aires [69, 70, 76]. Floater et al. proposent une étude des différentes techniques de paramétrage planaire existantes [77, 78].

### 1.3.2.2 Découpage

En général les techniques de paramétrage planaire s'appliquent aux objets dont la topologie est équivalente à celle d'un disque [79]. Pour ramener un objet quelconque à ce cas, celui-ci est généralement découpé le long de coutures. Le résultat peut être composé d'un ou plusieurs graphes planaires. On appelle "atlas" la collection des graphes associés à un objet 3D. Il existe deux approches différentes du découpage. Le *découpage de coutures* (*seam cutting*) consiste à générer un unique graphe planaire à l'aide de multiples coutures. Ces techniques permettent de minimiser les discontinuités mais fournissent des graphes complexes sujets aux distorsions [80–88]. À l'inverse, la *segmentation de mesh* (*mesh segmentation*) consiste à minimiser les distorsions en découpant le mesh en plusieurs graphes, quitte à introduire des discontinuités [89, 90]. Certaines techniques de segmentation tendent à produire de nombreux graphes de petite taille [65, 74, 91] et utilisent une approche itérative pour la segmentation [92, 93] éventuellement basée sur les caractéristiques topologiques du mesh [94].

Il est possible de contraindre le paramétrage pour diminuer la distorsion au niveau des discontinuités introduites par les coutures [95, 96]. Le paramétrage contraint permet également de fournir des graphes plus adaptés à l'utilisateur (c'est-à-dire plus faciles à éditer manuellement) notamment pour la peinture à partir de photographies [97, 98] ou pour les textures multi-objet [99].

### 1.3.2.3 Dépliage interactif

Classiquement, le dépliage UV est indépendant de la texture à appliquer à l'objet. Ceci peut potentiellement poser un problème de résolution au niveau de la texture. En effet, certaines parties du mesh peuvent nécessiter plus de détails que d'autres. Il est alors intéressant d'effectuer un dépliage accordant une plus grande surface de texture aux zones nécessitant des détails [80, 100–104]. Certains travaux permettent d'éditer de manière transparente la carte UV au cours de la tâche de peinture : celle-ci s'adapte alors dynamiquement au niveau de détail souhaité [35, 105, 106]. Ces techniques sont particulièrement utiles dans le cas d'une utilisation par un utilisateur novice. En effet, même si elles fournissent un contrôle artistique moins élevé qu'un dépliage manuel, elles permettent à l'utilisateur de s'affranchir de la tâche fastidieuse et complexe de dépliage, pour des résultats corrects dans un cadre non professionnel.

### 1.3.2.4 Alternatives

Devant la difficulté amenée par le dépliage UV, des techniques alternatives sans textures ont été proposées afin de stocker des informations concernant le mesh. Ainsi, Hanrahan et al. proposent de stocker la couleur directement au niveau des sommets du mesh [107]. Cependant cette technique ne permet pas de stocker un niveau de détail supérieur à la résolution du mesh. D'autres techniques proposent de stocker l'information sous forme de "coups de pinceau" au niveau de la surface du mesh [108, 109] ou encore d'utiliser des octrees comme structure de donnée de stockage [110, 111]. Enfin, une autre approche consiste à stocker les informations de géométrie et d'apparence sous forme de nuages de points (*surface pixels*) [112–115]. Ces techniques utilisent un paradigme totalement différent des notions de mesh et de texture.

### 1.3.3 Peinture par projection

Hanrahan et al. sont les premiers à proposer une technique wysiwyg (What You See Is What You Get) où l'utilisateur peint directement dans l'espace 3D [107] : la couleur (ou tout autre paramètre de matériau) est appliquée au niveau de la projection du curseur sur le modèle selon la vue courante (figure 1.13). Cette technique est connue sous le nom de *projection painting*. Hanrahan propose plusieurs types de projection qu'il référence sous le nom de *brushes* :

**Parameter-space brush :** Cette brosse peint directement dans la texture (figure 1.13a). La position du curseur sur l'écran est projetée sur la surface du mesh, et

l’empreinte de la brosse, sans déformation, est alors appliquée à la coordonnée correspondante dans la texture 2D.

**Screen-space brush :** Chaque pixel de l’empreinte de la brosse sur l’écran est projeté indépendamment sur la surface, et donc sur la texture. Cette brosse donne l’impression de peindre à l’aide d’un spray. En effet, des espaces apparaissent entre les projections des pixels au niveau du mesh (figure 1.13b).

**Tangent-space brush :** La position du curseur sur l’écran est projetée sur la surface du mesh. L’empreinte de la brosse est alors projetée dans la texture aux coordonnées de point du mesh, suivant la normale du mesh à cet endroit (figure 1.13c).

La *Parameter-space brush* a deux inconvénients majeurs. Si la carte UV comprend des distorsions trop importantes, l’empreinte de la brosse sera déformée sur le mesh et ne correspondra pas à l’intention de l’utilisateur. De plus, si la carte UV contient des coupures, des discontinuités vont apparaître sur le mesh au niveau de celles-ci. Johnson et al. proposent de modifier automatiquement la taille de l’empreinte pour compenser les distorsions de la carte UV [116]. Cette modification est transparente pour l’utilisateur : ce dernier à l’impression de peindre avec une brosse de taille constante (figure 1.14).

Les travaux de modélisation de brosse en 3D peuvent être également appliqués à la peinture 3D [115, 117–121]. La modélisation de la brosse permet d’obtenir une empreinte de taille et de forme dynamiques au rendu plus réaliste.

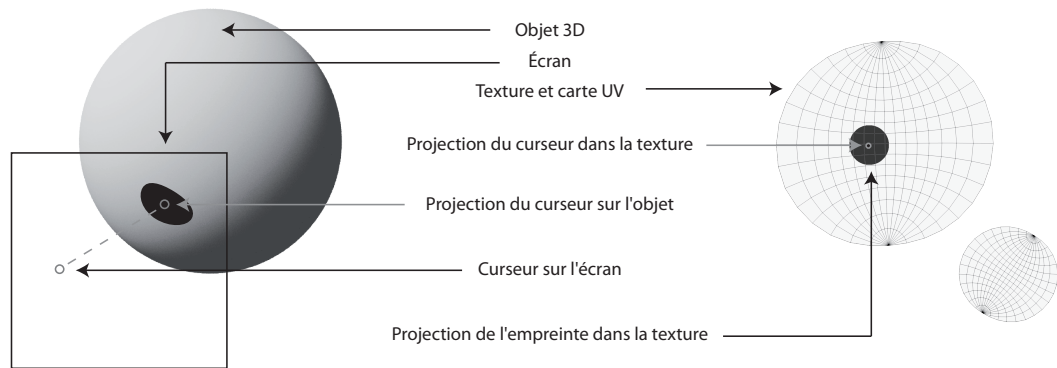
Les calculs de projection sont assez lourds et ne peuvent pas être appliqués en temps réel [107]. Hanrahan propose d’appliquer les modifications aux textures uniquement pendant les périodes de navigation 3D. En effet, tant que le point de vue ne change pas la modification de texture peut être simulée par un affichage 2D des strokes au-dessus de la vue 3D [107]. Sharfig et al. ont proposé d’améliorer les performances de la peinture par projection en déportant les calculs au niveau de la carte graphique [122].

Daily et al. ont étudié l’efficacité de la peinture par projection auprès d’un public de graphistes professionnels [123]. Ils ont montré que cette technique permet de simplifier le workflow du graphiste en supprimant les aller-retour entre travail 2D et 3D. La peinture par projection permet d’avoir un workflow 3D complètement intégré (figure 1.15). Daily et al. ont montré que ce workflow est plus efficace que l’original [123].

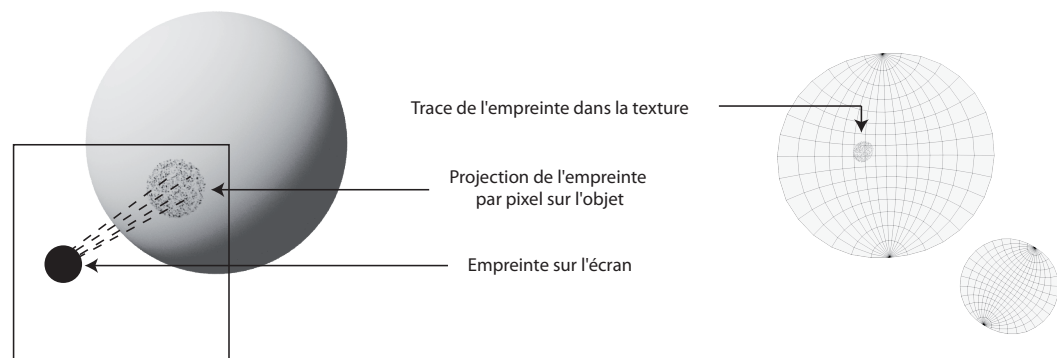
### 1.3.4 Autres techniques de peinture

La peinture par projection n’est pas la seule interaction de peinture 3D. Plusieurs travaux ont étudié la peinture à l’aide d’un dispositif haptique à six degrés de liberté [116, 124,

*a. Parameter-space brush*



*b. Screen-space brush*



*c. Tangent-space brush*

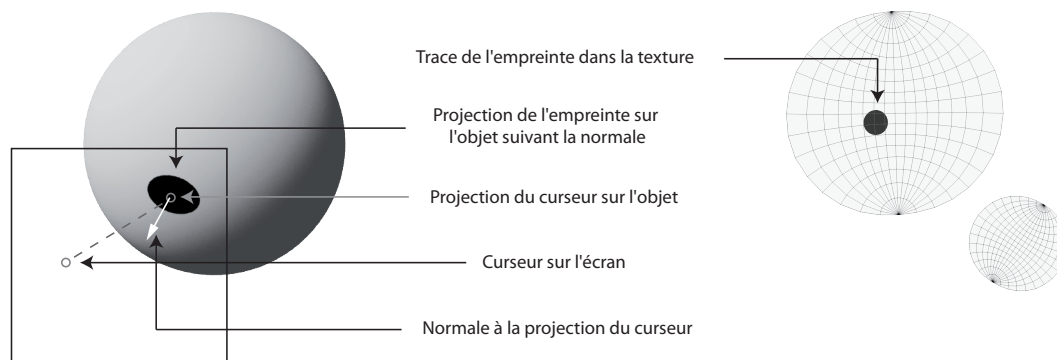


FIGURE 1.13 – Projection painting [107] :

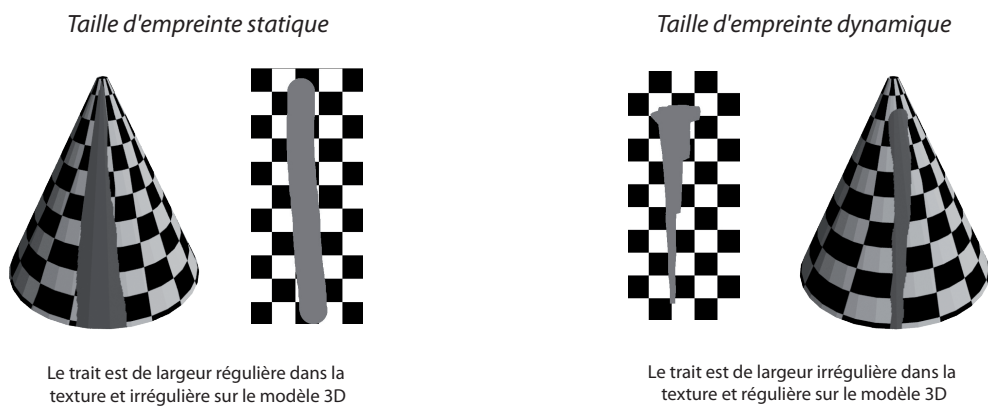


FIGURE 1.14 – Empreinte dynamique pour compenser la distorsion [116]

125]. Dans ces travaux, l'utilisateur positionne et oriente une brosse en 3D. Le contact avec le mesh est signalé par retour de force.

Agrawala et al. ont proposé une méthode de peinture 3D tangible [126]. Un objet réel est tout d'abord modélisé à l'aide d'un scanner 3D, puis l'utilisateur *peint* sur l'objet réel avec un localisateur (capteur de position magnétique). Les mouvements du localisateur sont interprétés afin de reproduire les coups de pinceau sur l'objet virtuel. Le retour visuel est fourni à l'utilisateur en réalité augmentée : la peinture virtuelle est immédiatement projetée sur l'objet physique.

D'autres travaux s'intéressent à la génération des traits (strokes). Kalnins et al. proposent d'associer les strokes au mesh sous forme d'annotations [127], de même que Schmid et al. [36]. À l'inverse Keefe et al. proposent de créer des strokes dans l'espace indépendamment de tout support (c'est à dire sans mesh) [128]. Cette technique a lieu en immersion totale : l'espace 3D est projeté sur les murs de la pièce où se trouve l'utilisateur. Les strokes sont générées à partir des mouvements d'un pinceau équipé d'un

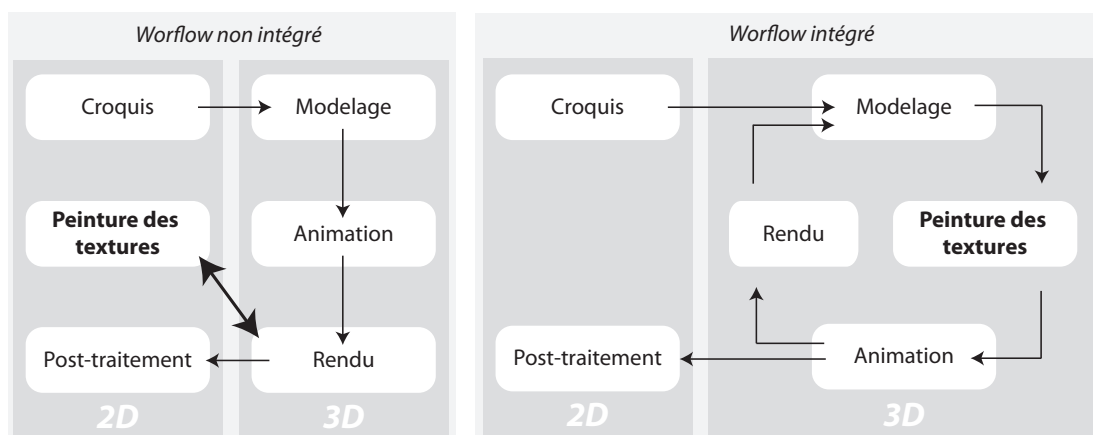


FIGURE 1.15 – Simplification du workflow grâce à la peinture par projection [123]

localisateur dans l'espace. L'utilisateur peint donc dans le vide, ce qui est assez éloigné des interactions traditionnelles de peinture et de dessin où un contact entre l'outil et le support est nécessaire. Une technique similaire a été employée par Clay et al. pour un spectacle artistique [129] : des danseurs génèrent des strokes en 3D grâce aux mouvements de leurs mains dans l'espace, qui sont projetées sur scène en temps réel. Keefe et al. proposent également une technique d'illustration 3D par génération de strokes pour une interface haptique [45].

Cependant, toutes ces techniques ont en commun de nécessiter une mise en place et des périphériques difficilement accessibles au grand public, que ce soit matériellement ou financièrement (dispositifs haptiques, pièce de projection 3D, localisateurs précis, etc.).

## Bilan

Dans ce chapitre, nous avons vu que la peinture 3D pose différents types de problèmes dans le cadre d'usage novice.

Tout d'abord, les deux types de dispositifs d'entrée disponibles pour l'interaction au stylet, *écrans interactifs* et *tablettes graphiques*, présentent chacun des avantages et inconvénients qui sont identifiés d'un point de vue général, mais dont l'impact sur la qualité de la peinture n'a pas été évalué formellement. Afin de palier ce manque, nous proposons au chapitre 4 une étude expérimentale de l'effet de l'aspect direct de l'interaction de peinture numérique pour des utilisateurs novices.

Un autre problème est celui posé par le dépliage UV, préalable complexe à la peinture de textures. Cependant, nous avons vu que certains travaux permettent d'automatiser le dépliage et de le rendre transparent pour l'utilisateur [35]. Nous retenons cette approche qui semble particulièrement bien adaptée à un usage novice.

Du point de vue de l'utilisateur, l'interaction de peinture 3D peut être effectuée en 2D sur une texture dépliée de l'objet, ce qui implique une conversion mentale ardue pour un utilisateur novice, ou en 3D par projection, qui est plus directe et proche de la peinture traditionnel sur objets physiques. La peinture par projection semble donc être une meilleure solution dans un cadre récréatif. De plus, elle nécessite uniquement un matériel facilement accessible au grand public (tablette graphique ou écran interactif).

Cependant, la peinture 3D par projection nécessite la combinaison de techniques de navigation 3D afin de pouvoir changer de point de vue sur l'objet à peindre. La navigation doit donc être également accessible à des utilisateurs novices. Or la navigation 3D est une

tâche interactive difficile. Nous proposons donc dans le chapitre suivant un état de l'art des interactions de navigation, et nous proposons notre contribution dans le chapitre 5.



## Chapitre 2

# Navigation 3D

### Sommaire

---

<b>Introduction</b> . . . . .	<b>33</b>
<b>2.1 Navigation libre</b> . . . . .	<b>35</b>
2.1.1 La Métaphore Rotate/Pan/Zoom . . . . .	36
2.1.1.1 Contrôle Pan/Zoom . . . . .	36
2.1.1.2 Contrôle Rotate . . . . .	38
2.1.1.3 Contrôle Rotate/Pan/Zoom . . . . .	40
2.1.2 Les métaphores Through-the-lens . . . . .	42
2.1.3 Intégralité du contrôle . . . . .	43
<b>2.2 Navigation assistée</b> . . . . .	<b>45</b>
2.2.1 Points d'intérêts discrets . . . . .	45
2.2.2 Orbiting . . . . .	47
<b>Bilan</b> . . . . .	<b>49</b>

---

### Introduction

La peinture numérique sur objet 3D nécessite de pouvoir naviguer autour de l'objet afin d'accéder à toutes les faces de celui-ci et d'appréhender correctement son apparence dans l'espace. La qualité de l'interaction de peinture numérique, notamment en terme de confort et de précision, dépend donc en partie de la qualité de l'interaction de navigation.

Afin de mieux cerner le type d'interaction de navigation nécessaire à la tâche de peinture, on peut s'appuyer sur la taxonomie de navigation 3D proposée par Tan et al.[130]. Il distingue en effet trois grandes familles de navigation : l'*exploration*, qui consiste à acquérir des informations concernant un environnement 3D inconnu, la *recherche*, qui consiste à déterminer la localisation d'un élément dans un environnement et à naviguer

jusqu'à ce dernier, et l'*observation*, qui consiste à établir un point de vue particulier sur un objet.

L'*observation* correspond donc particulièrement bien au besoin de changement de point de vue et de navigation autour de l'objet nécessaire à la peinture sur modèle 3D. L'*exploration* paraît plus appropriée à la peinture d'environnements larges dans leur ensemble comme les décors naturels. La *recherche* est plutôt utile pour une étape préliminaire de sélection d'un objet à peindre dans un environnement plus large. La tâche de peinture sur modèle 3D nécessite donc principalement une interaction de type *observation*, c'est pourquoi nous allons limiter notre état de l'art à cette famille d'interactions.

Christie et al. distinguent deux grandes familles de contrôle interactif de la caméra, à savoir d'une part la navigation qui permet une totale liberté en terme de position et orientation de la camera que nous appellerons *navigation libre*, et d'autre part la navigation qui ne nécessite pas ou peu d'action dédiée de la part de l'utilisateur que nous appellerons *navigation assistée*[131]. Nous allons donc dans ce chapitre considérer les différentes techniques de navigation appropriées aux tâches d'observation, du point de vue de cette taxonomie.

Il est important de noter également la dualité qui existe entre les interactions de navigation et de déplacement 3D. La *navigation 3D* consiste à modifier la position et l'orientation de la caméra. À l'inverse, le *déplacement 3D* consiste à modifier la position et l'orientation d'un objet dans la scène en laissant la caméra fixe. Ces deux interactions sont utilisées à des fins distinctes, à savoir la modification du point de vue pour la première et la modification de la scène 3D pour la deuxième. Cependant, lorsque la scène 3D comprend un unique objet, ces deux types d'interaction se confondent et ne sont pas différenciables du point de vue de l'utilisateur : par exemple approcher la caméra de l'objet génère un rendu strictement identique à celui qui résulte d'approcher l'objet de la caméra. Il s'ensuit que, dans le cas d'un objet unique, une technique de déplacement 3D peut être algorithmiquement adaptée en technique de navigation 3D.

L'activité de peinture numérique 3D s'effectuant généralement sur un seul objet à la fois, nous sommes ramenés à un cas où la manipulation et la navigation sont duales. Lorsque l'objet en question fait partie d'une scène 3D complexe, le logiciel de peinture permet en général d'utiliser une *vue locale* qui cache le reste de la scène. On peut donc réaliser la navigation lors d'une tâche de peinture aussi bien à l'aide de techniques de navigation que de technique de déplacement d'objet. Dans cet état de l'art nous prendrons donc en compte indifféremment ces deux types d'interaction.

Enfin, nous orientons nos recherches vers les techniques s'appuyant sur l'interaction tactile multipoint. L'interaction 3D s'est historiquement orientée vers des dispositifs spécifiques offrant de nombreux degrés de liberté [132, 133]. Mais de par leur spécificité, ces dispositifs ne sont pas adaptés à un usage novice récréatif. Les surfaces tactiles, par contre, ont récemment connu un essor important et sont notamment perçues comme offrant des interactions intuitives. De fait, nous verrons que de nombreux travaux récents indiquent que ce type de surface peu contribuer à la démocratisation de l'interaction 3D.

## 2.1 Navigation libre

Conceptuellement, la navigation consiste à changer de point de vue sur la scène 3D en manipulant la caméra. Le contrôle de la caméra peut être décomposé suivant sept degrés de liberté [131, 134] : comme on peut le voir à la figure 2.1, trois concernent la gestion de la position de celle-ci, et trois autres concernent la gestion de son orientation. Le dernier degré de liberté permet de contrôler la dimension du champ de vision par la manipulation de la distance focale. Ce degré de liberté est en général très peu utilisé en cours de navigation ou de peinture 3D. En effet, il est difficile à appréhender, surtout pour les amateurs, et est plutôt perçu comme une “déformation de l'image”. Dans la grande majorité des cas, la valeur par défaut est conservée, ou bien un réglage est fait une fois pour toute en début de tâche. Nous laissons donc de côté la distance focale et nous nous concentrerons sur le positionnement de la caméra.

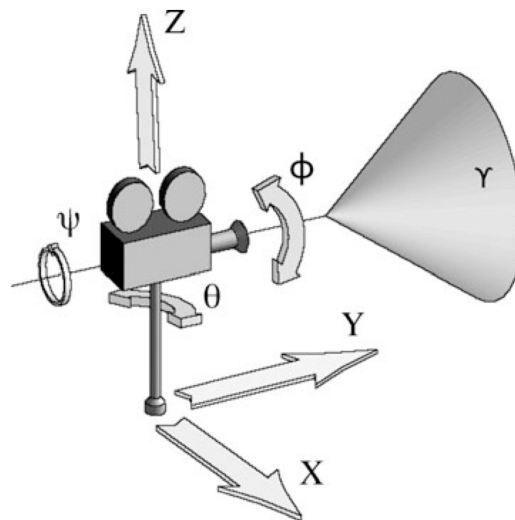


FIGURE 2.1 – Les degrés de liberté du contrôle de la caméra [131] : position  $(X, Y, Z)$ , orientation  $(\theta, \phi, \psi)$ , focale  $(\gamma)$

Nous passons maintenant en revue les différentes métaphores de navigation permettant une totale liberté de position et d'orientation. Nous considérerons dans un premier temps les métaphores de type *Rotate/Pan/Zoom*, puis les métaphores de type *Through-the-lens*.

Enfin, nous considérerons le caractère *intégral* de l'interaction, c'est à dire la possibilité de contrôle simultané de plusieurs degrés de liberté.

### 2.1.1 La Métaphore *Rotate/Pan/Zoom*

Cette métaphore constitue le standard implémenté dans de nombreux logiciels de création de contenu 3D (Maya, 3DsMax, Blender, etc.). Elle est conçue à la base pour l'interaction à la souris, mais elle peut être utilisée avec d'autres dispositifs d'entrée possédant au moins deux degrés de liberté, comme l'interaction tactile ou le stylet par exemple. Cette métaphore est constituée de trois interactions élémentaires :

**Rotate** Cette interaction consiste à faire tourner la caméra autour d'un point de pivot. Celui-ci appartient généralement au modèle observé.

**Pan** Cette interaction consiste à déplacer la caméra dans le plan de l'écran, sans changer son orientation.

**Zoom** Cette interaction consiste à déplacer la caméra le long de la *ligne de vue* (c'est-à-dire l'axe perpendiculaire à l'écran).

On peut faire à ce stade deux remarques : (1) l'interaction *Rotate* est l'interaction de navigation duale de l'interaction de manipulation consistant à modifier l'orientation d'un objet ; (2) le terme *Zoom* est utilisé ici par abus de langage : il ne s'agit pas d'une interaction modifiant la distance focale de la caméra, mais bien d'une interaction modifiant la position de celle-ci.

Ainsi, on peut considérer que les interactions *Pan* et *Zoom* permettent de gérer la position de la caméra, et l'interaction *Rotate* permet d'en gérer l'orientation. Nous allons donc dans la suite nous intéresser indépendamment aux techniques de *Pan/Zoom* et aux techniques de *Rotate*, puis nous considérerons les techniques implémentant ensemble les trois interactions élémentaires *Rotate/Pan/Zoom*.

#### 2.1.1.1 Contrôle *Pan/Zoom*

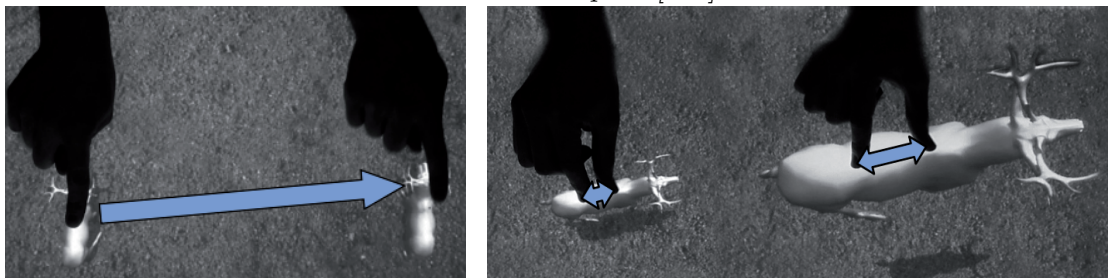
Le contrôle de la position de la caméra s'effectue à l'aide d'interactions de type translation, qui peuvent être commandées grâce à différents dispositifs d'entrée.

Concernant l'interaction à la souris, on peut différencier deux approches principales dans les logiciels de création 3D. La première consiste à associer les mouvements de la souris au *Pan* et les mouvements de la roulette au *Zoom*. La seconde consiste à associer les mouvements de la souris au *Pan* et au *Zoom* de manière modale. La différenciation des modes peut se faire à l'aide de raccourcis clavier ou en pressant un bouton de la souris. Comme la souris possède deux degrés de liberté, son utilisation pour le *Zoom* qui n'en nécessite qu'un seul suppose une agrégation ou l'utilisation d'un seul des deux degrés de liberté. Dans le cas de l'agrégation, la valeur du zoom est généralement proportionnelle à la distance parcourue sur l'écran. Dans le cas de la suppression d'un degré de liberté, la valeur du zoom est proportionnelle à la distance parcourue sur l'écran dans une seule direction : verticale, horizontale ou oblique (plus rarement).

Pierce et al. proposent également ce type de navigation dans le cadre d'environnements immersifs[136] : la distance entre la main et le visage de l'utilisateur contrôle le *Zoom* tandis que les autres mouvements de la main contrôlent le *Pan*.

En interaction tactile multipoints, la combinaison *Pan/Zoom* représente une forme de consensus pour le déplacement de caméra. En effet, de nombreuses études mettent en place les contrôles de position de manière similaire : le *Pan* est souvent contrôlé à l'aide

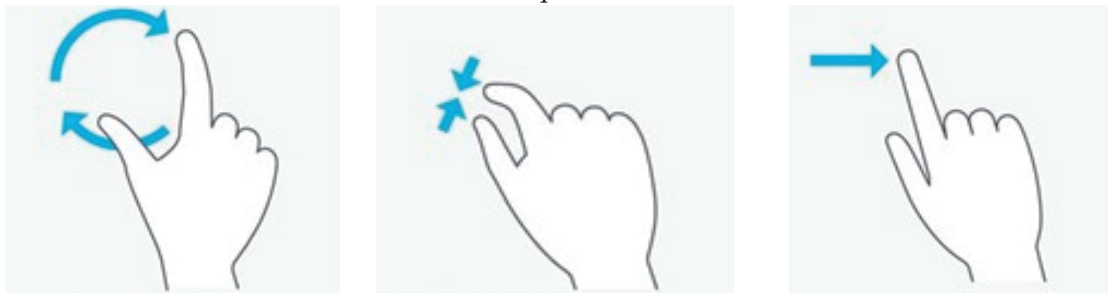
*a. Translations 3D en interaction tactile multipoint[135]*



Translation dans le plan de l'écran

Translation selon l'axe perpendiculaire à l'écran

*b. Gestes RST en interaction tactile multipoint*



Rotation (Twist)

Scale (Pinch)

Translation (Drag)

FIGURE 2.2 – Translations 3D et gestes RST en tactile multipoint : les gestes sont similaires pour les translations 3D et l'interaction RST

d'un geste de *glisser* (drag) à un ou plusieurs doigts sur la surface d'acquisition[135, 137–141, 141] (figure 2.2a gauche). Le *Zoom* est quant à lui souvent contrôlé à l'aide d'un geste de *pincer* (pinch)[135, 137, 140](figure 2.2a droite).

On peut noter que cette combinaison de gestes correspond à l'interaction RST (*Rotate, Scale, Transform*)[142] largement répandue sur tablettes et Smartphones. On s'appuie ici sur un double dualité : la réduction de taille d'un objet dans une scène 2D peut être perçue comme un éloignement de la caméra si l'objet est maintenant considéré dans une scène 3D (première dualité). Par ailleurs, le déplacement de l'objet par rapport à la caméra peut être perçu comme le déplacement de la caméra par rapport à l'objet (dualité évoquée plus haut). On s'appuie donc sur une interaction largement répandue dans les scènes 2D pour assurer le caractère intuitif d'une interaction similaire, mais conçue pour contrôler pan et zoom en 3D.

### 2.1.1.2 Contrôle Rotate

Les interactions de contrôle de la rotation ont été largement étudiées au cours des dernières décennies. Contrairement aux translations, les rotations 3D semblent plus difficiles à appréhender.

Chen et al. étudient différentes techniques de rotations utilisant une souris (et donc uniquement deux degrés de liberté) comme périphérique d'entrée[143]. Ils sont les premiers à proposer l'émulation d'un trackball comme alternative aux barres de défilement pour effectuer des rotations 3D. Ils proposent deux types de trackballs différents : pour le premier, appelé *Continuous XY+Z*, un mouvement de la souris sur l'objet provoque une rotation de celui-ci selon l'axe perpendiculaire au mouvement. Un mouvement de rotation à l'extérieur de l'objet provoque une rotation correspondante selon l'axe perpendiculaire à l'écran. Le second trackball, appelé *Virtual Sphere*, simule l'interaction qui serait obtenue à l'aide d'un trackball physique : l'objet manipulé est à l'intérieur d'une sphère invisible, et les mouvements de la souris agissent comme un contact physique sur celle-ci en la faisant tourner sur elle-même. Comme pour le *Continuous XY+Z*, les mouvements à l'extérieur de la sphère provoquent une rotation selon l'axe perpendiculaire à l'écran. Cependant, la *Virtual Sphere* permet un contrôle intégré des trois axes de rotations lorsque la souris est sur la sphère, contrairement au *Continuous XY+Z* qui ne permet qu'un contrôle intégré des axes dans le plan de l'écran.

Chen et al. montrent que les trackballs *Virtual Sphere* et *Continuous XY+Z* sont plus rapides que les barres de défilement pour des rotations complexes, sans perte mesurable de précision.

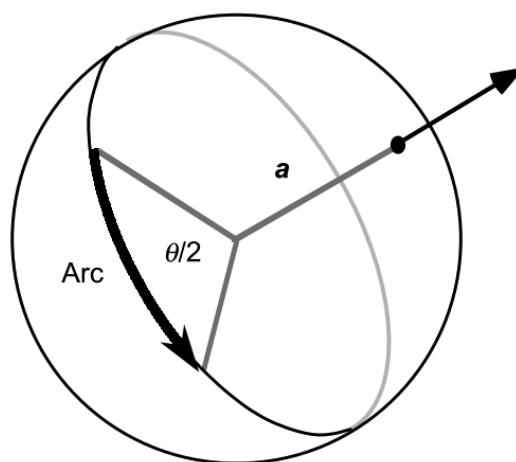


FIGURE 2.3 – ArcBall[144] : Lorsque l'utilisateur déplace la souris selon l'arc indiqué, une rotation selon l'axe  $a$  est effectuée, dont l'amplitude  $\theta$  correspond au double de l'angle défini par l'arc

Shoemake propose une autre technique de trackball appelée *Arcball*[144]. Cette technique simule également la présence d'une sphère transparente englobant l'objet, tout comme le *Virtual Sphere*. Comme on peut le voir à la figure 5.4, lorsque la souris trace un arc sur la sphère, une rotation selon l'axe perpendiculaire au plan défini par cet arc est effectuée. Cependant, afin de permettre à l'utilisateur d'effectuer une rotation très proche de 360 degrés en un seul geste, l'amplitude de cette rotation correspond au double de l'angle défini par l'arc tracé. Shoemake présente cette technique comme plus facile d'utilisation que la *Virtual Sphere* et le *Continuous XY+Z* car elle évite l'effet d'*hystérésis*. Lorsque cet effet est présent, comme avec les techniques de Chen et al., un mouvement de boucle à l'aide de la souris ne ramène pas l'objet dans son orientation initiale, ce qui peut être contre-intuitif. Ainsi, selon Shoemake, l'*Arcball* utilise une meilleure correspondance entre facteurs humains et fondamentaux mathématiques.

Cependant, Hinckley et al. comparent l'efficacité de la *Virtual Sphere* et de l'*Arcball*, et ne trouvent pas de différence significative de performances entre ces techniques[145].

Plus récemment, Bade et al. ont introduit des principes ergonomiques pour comparer les techniques de rotation, en supplément des mesures traditionnelles de performances comme la précision ou la vitesse[146]. Les auteurs affirment que ces principes sont cruciaux pour obtenir des techniques de rotations prévisibles et faciles à utiliser :

1. B1 : Des actions similaires doivent provoquer des réactions similaires
2. B2 : La direction d'une rotation doit correspondre à la direction du mouvement du périphérique d'entrée
3. B3 : Les rotations 3D doivent être transitives.

4. B4 : Le gain contrôle/affichage doit être personnalisable.

En analysant les techniques existantes à l'aide de ces principes. Bade et al. montrent que la technique *Continuous XY+Z* (renommée dans l'article *Two Axis Valuator*) en vérifie la plupart, contrairement à la Virtual Sphere et l'Arcball. Ils présentent donc cette technique comme étant plus prévisible et facile à utiliser que les autres, et confirment ce résultat à l'aide d'une étude expérimentale.

Depuis, le *Two Axis Valuator* est présenté comme la technique de trackball la plus facile à utiliser. Cependant, cette technique a été développée pour un dispositif d'entrée à deux degrés de liberté, la souris, afin de manipuler les 3 degrés de liberté de la rotation de manière modale selon la zone où se trouve le curseur. Elle se décompose de la manière suivante : (1) la rotation pure selon l'axe perpendiculaire à l'écran (Z) est un mode déclenché lorsque le curseur sort du trackball, et (2) la rotation selon les axes dans le plan de l'écran (X : horizontal et Y : vertical) est un mode déclenché lorsque le curseur est à l'intérieur de la projection de la sphère.

Une autre solution, proposée par Zhao et al. est d'utiliser le troisième degré de liberté de la souris (la molette) pour dissocier clairement les rotations selon X/Y de la rotation selon l'axe Z, et dissocier la position du curseur du mode de rotation[147]. La molette est utilisée uniquement pour la rotation selon Z, tandis que les rotations X et Y sont contrôlées par la position du curseur. Avec cette approche cependant, il y a une divergence entre l'axe physique de rotation de la molette, qui est aligné avec l'axe X de l'écran, et l'axe logique de rotation, aligné avec Z.

Comme on peut le voir avec ces différentes études, le contrôle de l'orientation de la caméra à l'aide d'un dispositif d'entrée à deux degrés de liberté est un problème complexe. Il est difficile d'assurer un caractère intuitif aux interactions proposées, de fait ces interactions peuvent être difficile à maîtriser par des utilisateurs novices.

### 2.1.1.3 Contrôle Rotate/Pan/Zoom

On peut obtenir le contrôle de l'ensemble des six degrés de liberté à l'aide de dispositifs d'entrée à deux degrés de liberté en combinant les interactions de contrôle d'orientation et de position décrites précédemment. Cependant avec l'accès généralisé aux écrans tactiles multipoints, des techniques d'interaction récentes tirent parti de l'utilisation simultanée de plusieurs points de contact pour augmenter le nombre de degrés de liberté en entrée.

Par exemple, Hancock et al. proposent *Sticky Tools*, une technique d'interaction permettant de manipuler les six degrés de liberté à l'aide de un à trois points de contacts[135].



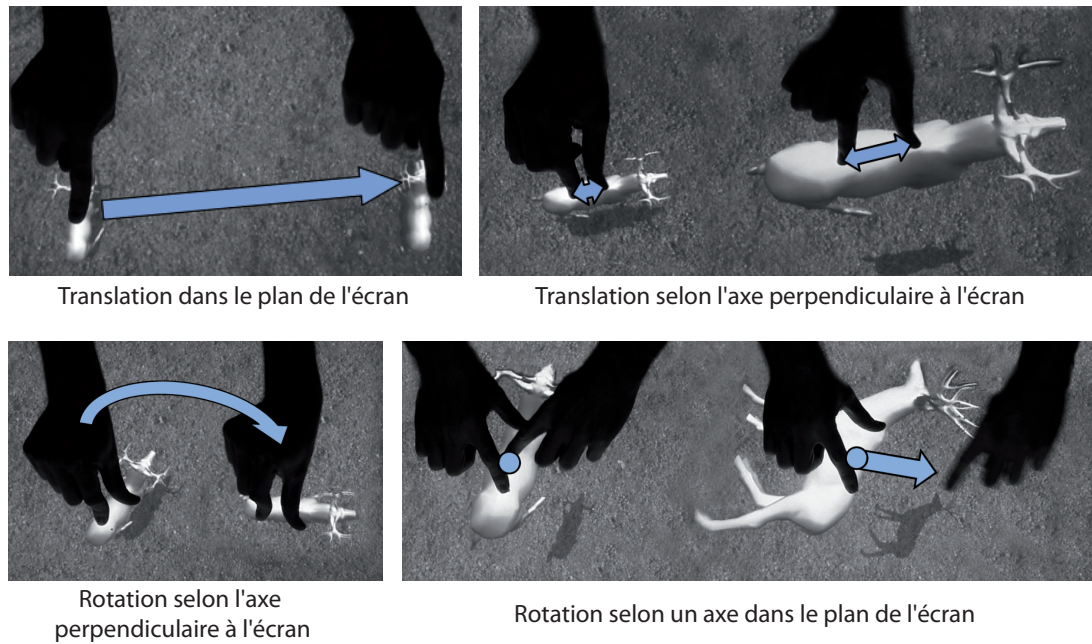
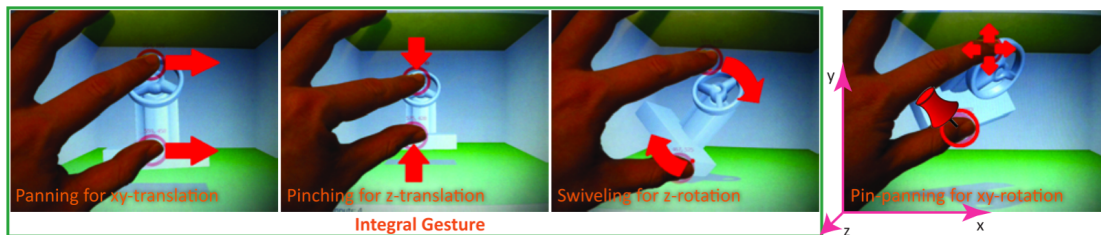


FIGURE 2.4 – Sticky Tools[135]

Comme on peut le voir à la figure 2.4, la gestuelle est similaire au RST décrit en section 2.1.1.1 : le *Pan*, selon les axes X et Y parallèles à l'écran, est contrôlé par le *glissement* d'un unique point de contact ; le *Zoom*, selon l'axe Z perpendiculaire à l'écran, est géré à l'aide d'un geste de *pincer* ; la rotation, selon Z également, correspond à un *twist*. La rotation, selon un axe composé de X et/ou Y, est effectuée à l'aide de trois points de contacts : les deux contacts immobiles définissent l'axe de rotation tandis que le contact mobile définit l'amplitude de la rotation. Cependant, il est difficile d'utiliser une seule main pour gérer deux points de contacts fixe et un point de contact mobile simultanément. Cette interaction nécessite donc généralement l'usage des deux mains, ce qui peut présenter un inconvénient si elle est combinée à une interaction de peinture numérique : elle imposerait en effet d'alterner en main dominante le tracé au stylet et la navigation au touché.

Plus récemment, Liu et al. ont proposé une technique de manipulation à 6 degrés de liberté qui combine les gestes RST présents dans *Sticky Tools* avec le *Two Axis Valuator* pour les rotations dans le plan de l'écran[140]. Comme on peut le voir à la figure 2.5, le mode *Two Axis Valuator* est activé en gardant un point de contact fixe, puis est commandé par le *glissement* d'un autre point de contact. Liu et al. ont montré que cette technique a des performances comparables à *Sticky Tools*. Elle est utilisable à l'aide de seulement deux doigts de la même main, ce qui la rend applicable à une utilisation dans le cadre de la peinture numérique. Cependant, cette technique ne permet pas de contrôler simultanément les 3 axes de rotation : elle nécessite un changement de mode, tout comme le *Two Axis Valuator + Z* proposé par Zhao et al[147].

FIGURE 2.5 – RST et *Two Axis Valuator*[140]

Enfin, Scheurich et al. ont proposé une adaptation du *Two Axis Valuator + Z* pour l'interaction tactile multipoints[141] : deux doigts collés qui glissent simultanément contrôlent un *Two Axis Valuator* ; la rotation autour de l'axe Z correspond à un mouvement de rotation de deux points de contacts dans le plan de l'écran ; les translations dans le plan de l'écran sont effectuées par le *glissement* d'un seul point de contact. Cette technique ne permet pas à l'utilisateur de gérer directement la translation selon l'axe perpendiculaire à l'écran : il s'agit en effet d'une technique d'alignement d'objets et par conséquent la profondeur est calculée automatiquement en fonction de l'environnement. Cette technique de manipulation ne peut donc pas être transposée en l'état en technique de navigation 3D.

### 2.1.2 Les métaphores *Through-the-lens*

Gleicher et Witkin proposent une métaphore de navigation appelée *Through-the-lens*, dans laquelle l'utilisateur contrôle la caméra indirectement, en manipulant et contraignant dans l'image 2D des éléments visibles de la scène 3D[148].

En s'inspirant de cette métaphore, Reisman et al. proposent une technique de manipulation à 6 degrés de liberté en interaction tactile multipoint, intitulée *Screenspace*[137]. Son principe est de modifier la position et l'orientation de l'objet de manière à ce que les mêmes points de celui-ci restent sous les doigts de l'utilisateur au cours de l'interaction (figure 2.6a). Reisman et al. montrent que *Screenspace* est plus efficace que *Sticky Tools*, mais que cette technique n'est pas toujours intuitive et conduit parfois à des comportements imprévisibles. Par exemple, à la figure 2.6b, l'utilisateur tente de faire tourner l'objet vers la droite : au début, le mouvement est celui souhaité, mais il se transforme au bout d'un certain temps en rotation vers le haut. Cela rend cette technique peu appropriée pour des utilisateurs novices. De plus, son principe de fonctionnement implique que certaines orientations de l'objet ne sont pas atteignables en un seul geste simple, en particulier celles nécessitant un point de contrôle invisible en début d'interaction parce que caché derrière l'objet.

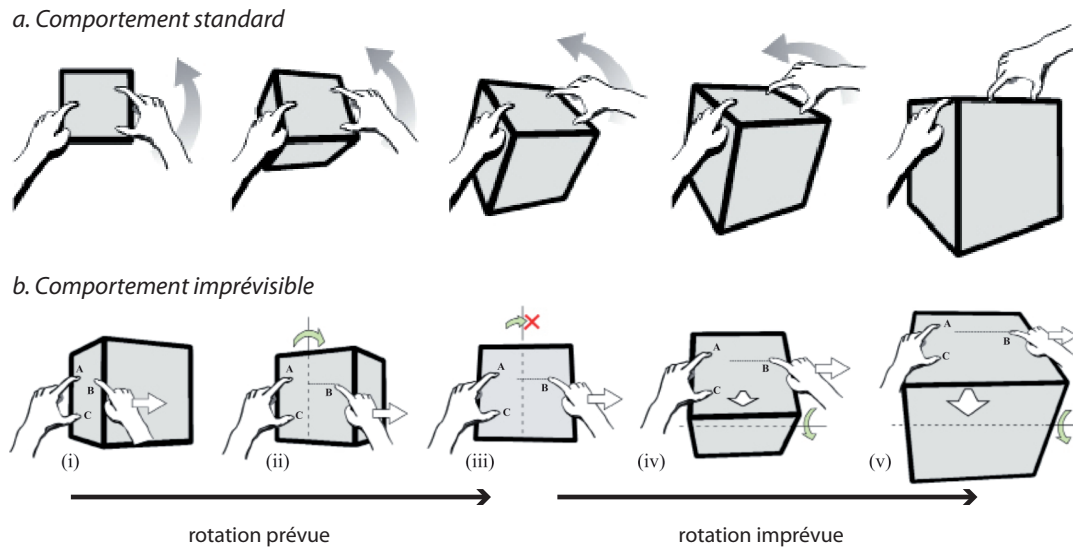


FIGURE 2.6 – Screenspace[137]

Plus récemment, Martinet et al. proposent une technique de manipulation à 6 degrés de liberté intitulée *DS3*[138]. *DS3* emprunte des éléments à la fois à la métaphore *Rotate/Pan/Zoom* et à la métaphore *Through-the-lens*. Celle-ci est en effet basée sur *Screenspace* pour le contrôle des rotations, et sur *Z-technique* pour le contrôle des translations[149]. Cette dernière consiste à effectuer le *Pan* à l'aide d'un geste de *glisser* d'un doigt en interaction directe, et le *Zoom* à l'aide d'un geste de *glisser* d'un autre doigt en interaction indirecte. Martinet et al. montrent que *DS3* est globalement plus efficace que *Screenspace* et *Sticky Tools*. Cependant, l'utilisation de *Screenspace* pour les rotations implique toujours un comportement parfois imprévisible et des orientations inaccessibles, ce qui la rend peu appropriée aux utilisateurs novices.

### 2.1.3 Intégralité du contrôle

Selon la théorie de Garner et al. sur la structure perceptuelle de l'information visuelle, un objet multidimensionnel peut être caractérisé par ses attributs selon deux catégories : une structure *intégrale* (la perception des attributs est combinée) ou *séparable* (les différents attributs sont distinguables)[150]. Par exemple, la couleur et la forme d'un objet sont des attributs séparables alors que la luminosité et la saturation d'une couleur sont des attributs intégraux.

Jacob et al. étendent cette notion de structure perceptuelle à l'interaction[151] : chaque dispositif d'entrée possède sa propre structure perceptuelle, de même que chaque tâche pour laquelle il est utilisé. Ils supposent que la correspondance entre les structures de la

tâche et du dispositif d'entrée est primordiale pour autoriser une interaction efficace, et confirment cette hypothèse par une étude expérimentale.

Par la suite, le *contrôle intégral*, c'est-à-dire le contrôle simultané de plusieurs degrés de liberté, a été longtemps considéré comme une manière d'améliorer l'efficacité des techniques d'interaction[143, 145, 152]. Cependant la théorie de Jacob et al. ne prend pas en compte l'aptitude des utilisateurs à manipuler simultanément plusieurs degrés de liberté : les capacités cognitives des utilisateurs en termes de contrôle moteur ont des limites. On peut les observer en demandant aux utilisateurs de contrôler trop de degrés de liberté à la fois, ou d'utiliser une technique d'interaction qui présente une correspondance inappropriée entre les entrées et les sorties. Wang et al. mettent notamment en évidence que les utilisateurs ont des difficultés à manipuler simultanément la position et l'orientation d'un objet[153].

Hinckley et al. comparent les performances de la Virtual Sphere et de l'Arcball avec celles d'un dispositif magnétique permettant le contrôle intégral des 3 axes de rotation[145]. Ils montrent que les utilisateurs sont plus efficaces avec le contrôle intégral, sans perte de précision.

Plus tard, Masliah et Milgram ont étudié la répartition du contrôle sur une tâche de docking à 6 degrés de liberté, en utilisant deux types de périphériques d'entrée, l'un isotonique et l'autre isométrique[154]. Ils proposent une métrique, la *m-metric*, afin de mesurer l'intégration du contrôle. Cette métrique a révélé que dans les deux cas les 6 degrés de liberté sont trop nombreux pour les participants, car ces derniers séparent le contrôle en deux groupes : les 3 translations et les 3 rotations. Cependant, les utilisateurs sont capables de contrôler les 3 rotations de manière intégrale. Comme l'étude est faite à l'aide de périphériques d'entrée ayant une correspondance directe entre la rotation appliquée au dispositif et la rotation résultante de l'objet virtuel (les directions de rotations sont identiques), ce résultat n'est pas directement applicable aux surfaces tactiles multipoints où cette correspondance est impossible.

Plus récemment, Martinet et al. ont étudié l'aspect intégral du contrôle dans le cadre de grands écrans tactiles multipoints[138]. Ils confirment l'étude de Masliah et Milgram en montrant que l'intégration des six degrés de liberté est moins efficace que la séparation des rotations et des translations.

En résumé, le contrôle intégral des trois axes de rotation est prouvé comme plus efficace qu'un contrôle séparé, mais uniquement dans le cas d'une correspondance directe entre entrées et sorties. Comme la correspondance directe est impossible sur les surfaces tactiles planes, la capacité des utilisateurs à effectuer efficacement un contrôle intégral des rotations reste à montrer.

## 2.2 Navigation assistée

Une autre approche de la navigation 3D dans le cadre d'une tâche d'observation consiste à amener automatiquement la caméra dans une position et une orientation adéquate pour mettre en valeur un *point d'intérêt* défini explicitement par l'utilisateur[155] ou implicitement par le système en fonction des actions de l'utilisateur[156].

La définition d'un point d'intérêt peut être

*discrète* : c'est à dire qu'un unique point est défini à la fois par le système ou l'utilisateur. Avec la technique *UniCam* par exemple, l'utilisateur définit des positions "clés" de la caméra, et le système se charge de créer les mouvements interpolés entre ces positions[157].

*continue* : lorsque le système redéfinit en permanence le point d'intérêt à atteindre. Cela correspond principalement aux techniques d'*orbiting*, qui contraignent la caméra à une distance constante de l'objet qu'elle observe[158].

### 2.2.1 Points d'intérêts discrets

Mackinley et al. proposent une technique de navigation automatique où lorsque l'utilisateur sélectionne un point d'intérêt, le système déplace la caméra vers la nouvelle position et orientation définie à partir du point d'intérêt (voir figure 2.7)[155]. Il s'agit d'une interpolation logarithmique, qui permet de couvrir rapidement de grandes distances puis de ralentir à proximité du point de vue souhaité.

Plus récemment, Zeleznik et al. proposent une technique qui tient compte de la géométrie de l'objet[157]. Après la sélection d'un point d'intérêt par l'utilisateur, le système génère un chemin de transition dans lequel la caméra reste à égale distance de l'objet, et pointe vers celui-ci de manière à être légèrement inclinée par rapport à la normale de la face visée

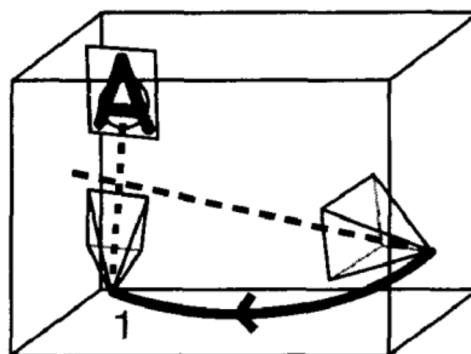


FIGURE 2.7 – Point d'intérêt[155] : Lorsque l'utilisateur sélectionne le A, la caméra se déplace et s'oriente suivant une interpolation logarithmique

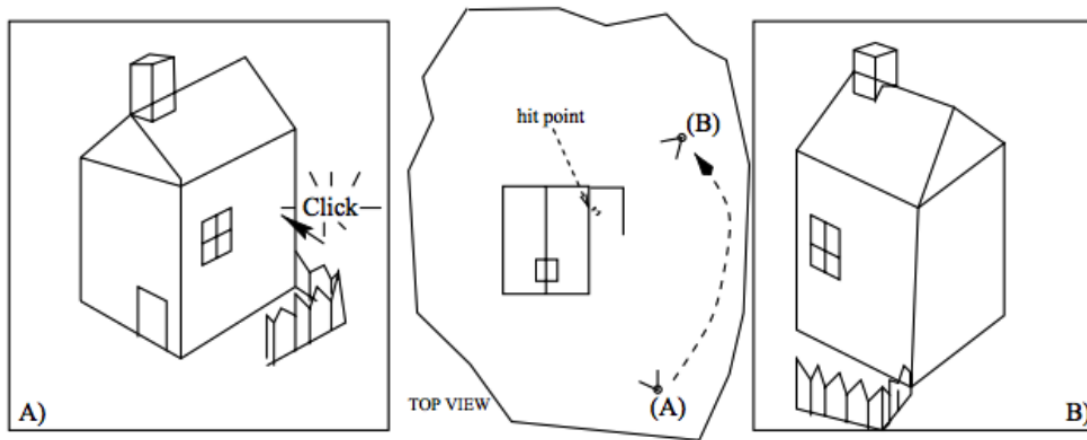


FIGURE 2.8 – Point d'intérêt[157] : Après la sélection du point d'intérêt en A), la camera s'anime selon le tracé présenté ci dessus pour atteindre les positions et orientations en B)

(figure 2.8). Tan et al. proposent une autre approche de cette méthode, où l'animation du déplacement est contrôlée par l'utilisateur[130].

Decle et al. ont développé la technique *Scruticam*[159], où lorsque l'utilisateur clique sur l'objet, un *glisser* vers le centre de l'écran amène la caméra face à la zone cliquée (figure 2.9). Si l'utilisateur déplace le curseur dans une autre direction, la caméra est translattée de manière à ce que le point sélectionné reste sous le curseur.

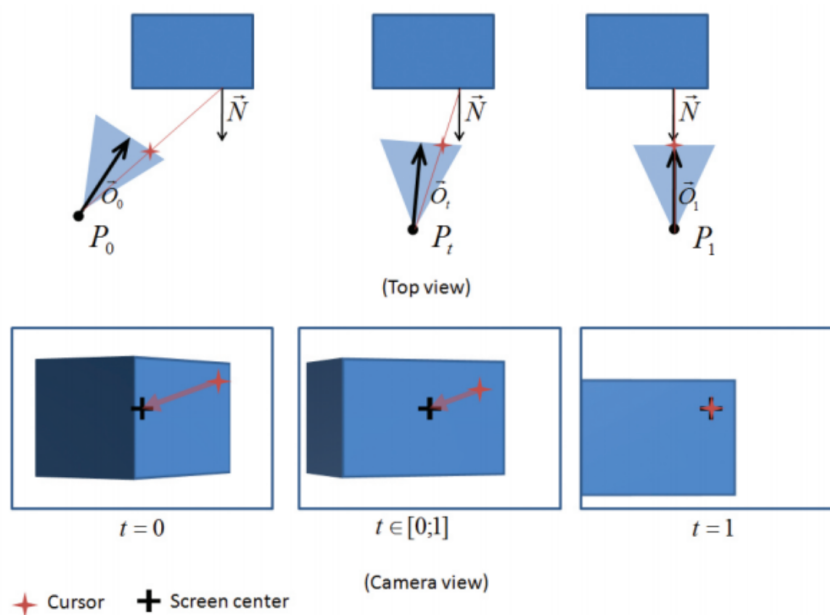


FIGURE 2.9 – Scruticam[159]

### 2.2.2 Orbiting

Dans les techniques que l'on vient de citer, les points d'intérêts sont définis de manière discrète. À l'inverse, Khan et al. proposent une interaction de navigation 3D basée sur la définition continue par le système de points d'intérêts successifs[158]. Cette interaction, intitulée *HoverCam*, est conçue pour un dispositif d'entrée à deux degrés de libertés, comme la souris. Elle consiste à déplacer la caméra selon une trajectoire similaire à celle d'un satellite en orbite autour de l'objet et soumis à un champ de gravité (figure 2.10). La direction générale du déplacement est contrôlée par la souris via une métaphore de type *grab* : lorsque l'utilisateur effectue un geste de *glisser* dans une direction, l'objet donne l'impression de se déplacer dans la même direction, ce qui implique que la caméra suit le mouvement inverse. Cette navigation est donc semi-automatisée dans la mesure où elle permet de contrôler les 6 degrés de liberté de la caméra à l'aide des deux degrés de liberté de la souris.

La technique *Hovercam* a cependant certaines limitations relevées par ses auteurs : elle peut avoir des comportements erratiques sur des objets dont la surface comprend trop de détails à haute fréquence. À l'inverse, son utilisation sur des objets à faible résolution conduit à des problèmes de tremblements dus aux trop grandes différences entre les normales des faces successives. De plus, l'algorithme de déplacement nécessite un calcul coûteux de "point d'intérêt le plus proche" qui peut réduire les performances sur les zones de forte densité du mesh, et par conséquent ralentir les mouvements de la caméra.

Ortega et al. ont développé une autre technique basée sur la définition continue de points d'intérêts, intitulée "Automated Camera Control for Drawing" (ACCD)[160]. Cette technique, adaptation de *HoverCam* à l'interaction directe, est spécifiquement conçue pour la peinture numérique. En effet, la gestion de la caméra est totalement automatisée dans la mesure où les points d'intérêts correspondent aux points de tracé définis par le stylet dans la tâche de peinture (figure 2.11). Cependant, cette technique ne peut être utilisée que pour le tracé de lignes continues, car elle ne permet pas de

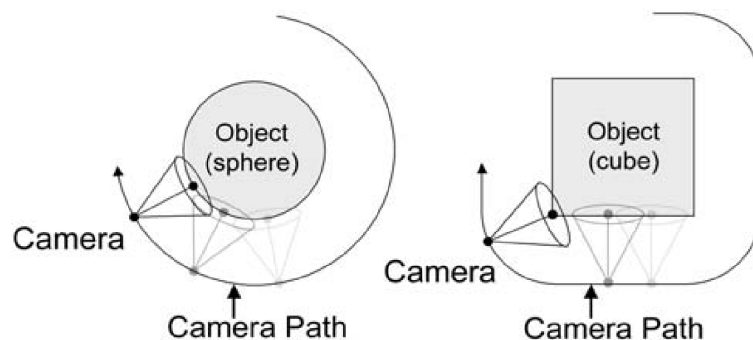


FIGURE 2.10 – *HoverCam*[158]

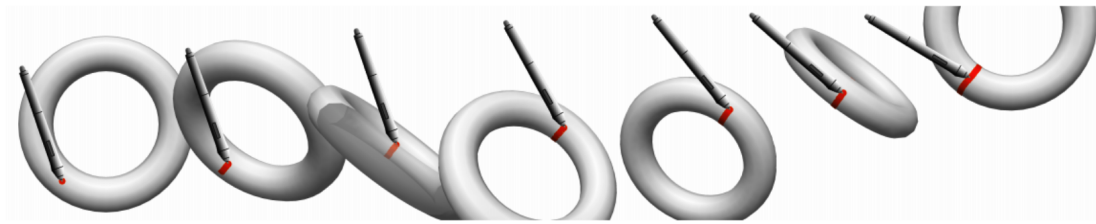


FIGURE 2.11 – Automated Camera Control for Drawing[160]

manipuler la caméra indépendamment de la tâche de peinture. Une technique similaire a été également développée par Xia et al.[161].

Ortega et al. proposent également la technique *Smooth HoverCam (SHOCam)* qui est une amélioration de *HoverCam*[162]. La technique *SHOCam* permet une interaction plus fluide que la technique *HoverCam*, en réduisant (1) la variabilité des mouvements et de l'orientation de la caméra (figure 2.12), et (2) les incohérences de mise en correspondance gestes utilisateurs et comportement de la caméra.

Ces différentes techniques de navigation assistée, qu'elles soient basées sur des points d'intérêts discrets ou continus, permettent de naviguer à l'aide de peu de degrés de liberté : elles ne nécessitent en effet pour le contrôle que deux[130, 155, 157–159, 162], voire zéro[160], degrés de liberté. Il semble que ce contrôle réduit puisse rendre ces techniques plus accessibles à des utilisateurs novices et amateurs que les techniques de navigation libre qui imposent le contrôle des six degrés de liberté, mais il faudrait vérifier expérimentalement cette hypothèse.

a. *HoverCam*b. *SHOCam*

FIGURE 2.12 – Comparaison des trajectoires de la caméra pour les techniques *Hovercam*[158] et *Smooth HoverCam*[162] : Les discontinuités du mouvement de la caméra avec *HoverCam* sont visibles au niveau des angles où la caméra s'arrête. Les discontinuités concernant l'orientation sont visibles via les espacements irréguliers des vecteurs le long de la trajectoire.



Cet avantage doit être mis en balance avec un inconvénient inhérent aux techniques de navigation assistée concernant la peinture 3D : c'est toujours le système qui contrôle le point de vue, or, dans certaines situations, l'utilisateur peut souhaiter définir un point de vue adapté en prévision d'un geste de tracé. Par exemple, les tracés suivant la direction de l'avant-bras sont inconfortables, ce qui conduit les dessinateurs à tourner et déplacer régulièrement leur feuille pour les éviter. Ainsi, dans le cas de la peinture 3D, l'utilisateur peut souhaiter orienter le modèle d'une manière ergonomique pour son geste de tracé, mais qui ne corresponde pas au point de vue proposé par la technique de navigation assistée.

Dans le cadre de la peinture numérique sur modèle 3D, les techniques de navigation assistée ne peuvent donc pas être utilisées de façon exclusive. Il est indispensable de pouvoir les désengager pour revenir à une interaction moins contrainte, permettant à l'artiste de choisir son point de vue. Le problème du choix d'une interaction de navigation libre adaptée à des utilisateurs novices reste présent.

## Bilan

On a vu dans ce chapitre qu'il existe deux types d'approches pour effectuer la navigation 3D pendant la tâche de peinture. La première consiste à fournir à l'utilisateur une totale liberté de mouvement. Si les interactions de translations ne semblent pas poser de problèmes aux utilisateurs, il n'en est pas de même pour le contrôle des rotations, en particulier pour des utilisateurs novices.

La seconde approche consiste à automatiser totalement ou en partie le processus de navigation. Les techniques qui en découlent sont *à priori* plus accessibles auprès d'un public novice. Cependant la limitation du choix du point de vue peut être contraignante pour la peinture 3D. Ces techniques ne peuvent donc pas être exclusives.

Nous avons donc choisi de nous intéresser au problème des rotations dans le cadre de la navigation libre pour utilisateurs novices. Nous proposons au chapitre 5 deux nouvelles techniques de rotations 3D intuitives qui tirent parti des degrés de liberté supplémentaires disponibles en interaction tactile pour permettre une manipulation intégrale des trois degrés de liberté d'orientation.

Dans le cadre de la peinture numérique sur objets 3D, nous allons maintenant aborder l'état de l'art sur la combinaison des interactions de navigation 3D et de peinture.



## Chapitre 3

# Peinture et Navigation

### Sommaire

---

<i>Introduction</i> . . . . .	<b>51</b>
<b>3.1 Interaction bi-manuelle</b> . . . . .	<b>52</b>
3.1.1 Coordination . . . . .	52
3.1.2 Le modèle de chaîne cinématique . . . . .	53
3.1.3 Performances . . . . .	56
<b>3.2 Choix des dispositifs d'entrée</b> . . . . .	<b>58</b>
3.2.1 Quels dispositifs pour un usage récréatif? . . . . .	59
3.2.2 Répartition des dispositifs . . . . .	60
3.2.2.1 Main dominante . . . . .	60
3.2.2.2 Main non dominante . . . . .	61
3.2.2.3 Synthèse . . . . .	62
3.2.3 Choix des technologies . . . . .	62
3.2.3.1 Stylet . . . . .	62
3.2.3.2 Tactile . . . . .	63
<b>3.3 Optimisations</b> . . . . .	<b>64</b>
<i>Bilan</i> . . . . .	<b>67</b>

---

### Introduction

Le dépôt d'encre au stylet et la navigation sont deux tâches bien différentes, mais qui doivent être alternées fréquemment dans la tâche globale de peinture numérique sur modèle 3D. Nous avons vu au chapitre précédent un moyen de faciliter cette alternance : en confiant la tâche de navigation au système par des interaction fondées sur les caméras automatiques. Mais l'utilisateur a parfois besoin de définir explicitement le point de vue

souhaité. Par conséquent, l'interaction globale de peinture sur modèle 3D nécessite deux modes distincts, l'un pour la peinture et l'autre pour la navigation.

Peinture et navigation peuvent être effectuées à l'aide d'un dispositif d'entrée unique en mettant en oeuvre un changement de mode explicite. Le changement de mode peut également être implicite en utilisant deux dispositifs, chacun étant affecté à l'un des deux modes. Il est également possible de concevoir une interaction bi-manuelle, une main étant affectée à la navigation et l'autre à la peinture. Dans ce cas, on peut utiliser un seul dispositif d'entrée s'il est capable de distinguer les deux mains, ou bien on peut utiliser deux dispositifs d'entrée qui peuvent être identiques ou différents.

Dans ce chapitre, nous allons tout d'abord nous intéresser aux principes généraux de l'interaction bi-manuelle, afin de déterminer la meilleure répartition des interactions de peinture et navigation entre les mains dominante et non-dominante. Nous passerons ensuite en revue les différentes études comparatives de performances entre techniques uni-manuelles et techniques bi-manuelles.

Nous considérerons par la suite l'ensemble des dispositifs d'entrée, ainsi que les manières de les combiner, qui peuvent être appropriés à la peinture numérique 3D, aussi bien pour une interaction uni-manuelle que bi-manuelle.

Enfin, nous considérerons les travaux portant sur l'optimisation du flux interactif dans le cas précis de la combinaison peinture et navigation : certaines techniques permettent en effet de réduire les temps de peinture et de navigation dans le cas d'une tâche de peinture sur modèle 3D.

## 3.1 Interaction bi-manuelle

### 3.1.1 Coordination

On trouve dans la littérature sur le contrôle moteur humain plusieurs théories concernant les performances relatives des deux mains [163–165]. Certaines études mettent en évidence la nature coopérative et intégrée des actions effectuées simultanément à l'aide des deux mains : il est en effet complexe d'effectuer deux tâches indépendantes simultanément [166].

La génération de poly-rythmes est un exemple bien connu : la plupart des gens éprouve des difficultés à désynchroniser les deux mains de manière à produire deux séquences rythmiques indépendantes et simultanées qui ne sont pas multiples l'une de l'autre. De manière générale, seuls les musiciens entraînés arrivent à produire ces types de rythmes.

Summers et al. ont montré que les personnes effectuant des rythmiques de la forme 5 : 3 adoptent une organisation moteur intégrée où les mouvements de la main dédiée au rythme lent sont subordonnés aux mouvements de la main dédiée au rythme rapide [167]. Ils ont également montré qu'entraîner les participants à produire les rythmes sur chaque main successivement avant de jouer les rythmes simultanément ne leur permet pas de mieux se synchroniser, dans la mesure où cela ne les incite pas à utiliser une approche intégrée.

Ainsi, les deux mains coopèrent de manière intégrée pour effectuer une tâche commune. Cependant, le rôle joué par chacune des deux mains peut être différent : par exemple, lors de certaines préparations culinaires, l'une des deux mains mélange la préparation à l'aide d'un fouet tandis que l'autre main maintient le récipient en place. Balkrishnan et al. définissent ce type d'interaction comme étant *asymétrique* [168]. À l'inverse, dans une interaction *symétrique*, les deux mains ont un rôle identique. Un exemple typique d'interaction symétrique est le redimensionnement d'un élément à la manière du geste de *pincer* en interaction tactile, mais en utilisant un doigt de chaque main. Cependant, selon Balakrishnan et al., une interaction symétrique peut tout de même amener à des performances asymétriques en termes d'erreur et de parallélisme [168].

Dans le cas de la peinture numérique sur modèle 3D, une interaction sous forme bi-manuelle nécessite que chaque main soit affectée à un rôle distinct, à savoir la navigation ou la peinture. Il s'agit donc d'une interaction asymétrique, c'est pourquoi nous allons dans la suite nous concentrer sur ce type d'interaction.

### 3.1.2 Le modèle de chaîne kinématique

Guiard propose une modélisation de l'interaction bi-manuelle asymétrique sous forme de *chaîne kinématique* [169]. Il définit une *chaîne kinématique* comme un ensemble de liaisons abstraites entre articulations tel que pour chaque liaison l'une des articulations organise son mouvement relativement à l'autre. L'articulation dépendante est appelée "articulation distale" tandis que l'autre articulation est intitulée "articulation proximale". Par exemple, si l'on considère une interaction consistant à attraper un objet, l'avant bras constitue une liaison kinématique dont le poignet est l'articulation distale et le coude l'articulation proximale.

Cependant, comme précisé précédemment, une liaison kinématique est abstraite et ne correspond pas nécessairement à une liaison physique (un os par exemple). Le modèle de Guiard suppose justement que les mains gauche et droite composent une chaîne kinématique *fonctionnelle* dans la mesure où la main dominante (distale) se déplace

relativement à la main non-dominante (proximale). Guiard propose les principes suivants concernant les relations entre main dominante et main non-dominante :

1. Référence spatiale : la main non dominante définit le cadre de référence spatiale dans lequel agit la main dominante.
2. Échelles asymétriques : les deux mains évoluent selon des échelles spatiales et temporelles différentes.
3. Antériorité de la main non dominante : la main non dominante agit avant la main dominante.

L'écriture donne un exemple concret d'activité mettant en évidence ces différents principes : l'utilisateur commence par poser sa main non-dominante sur la feuille (antériorité) de manière à orienter et placer correctement celle-ci (référence spatiale). Ensuite, la main dominante écrit (petits mouvements à haute fréquence) tandis que la main non-dominante ajuste régulièrement la position de la feuille (grands mouvements à basse fréquence). Les deux mains évoluent donc selon des échelles asymétriques.

Plusieurs travaux ont corroboré et approfondi cette modélisation : Hinckley et al. montrent que ces principes s'appliquent dans le cadre d'une tâche de docking bi-manuelle sur objet physique, consistant à encastrier un outil de pointage dans une cible [170]. Ils montrent également dans cette étude que l'on ne peut pas inverser la chaîne cinématique : l'interaction naturelle consiste à tenir la cible dans la main non-dominante et l'outil dans la main dominante. Or lorsque l'inverse est imposé aux participants, ces derniers ont tendance à amener la cible sur l'outil, donc à garder le comportement initial, tout en obtenant des performances significativement moins bonnes en terme de temps d'exécution. Cette étude confirme donc la modélisation de Guiard pour une tâche où les cadres de référence visuels et kinesthésiques sont communs.

Balakrishnan et al. étudient les domaines d'application des principes de Guiard selon les variations des cadres de référence kinesthésiques et visuels [171]. Ils évaluent pour cela l'effet la séparation des espaces kinesthésiques de chaque main et du retour visuel sur le temps d'exécution d'une tâche de connexion de points. Pour effectuer cette tâche, les participants déplacent une palette de couleur en main non dominante et tracent des lignes de couleur imposée entre les cibles avec la main dominante, à l'aide de deux palets sur une surface d'acquisition. La sélection de la couleur est effectuée en plaçant le slot correspondant de la palette sous le curseur de tracé en début de trait suivant le principe de la *Toolglass* développée par Bier et al [172](voir figure 3.1a). Balakrishnan et al. considèrent différentes conditions : concernant le cadre de référence kinesthésique, les deux mains évoluent soit dans un espace commun, soit dans un espace séparé (cf. figure 3.1b), et concernant le cadre de référence visuel, soit le curseur et la palette sont affichés

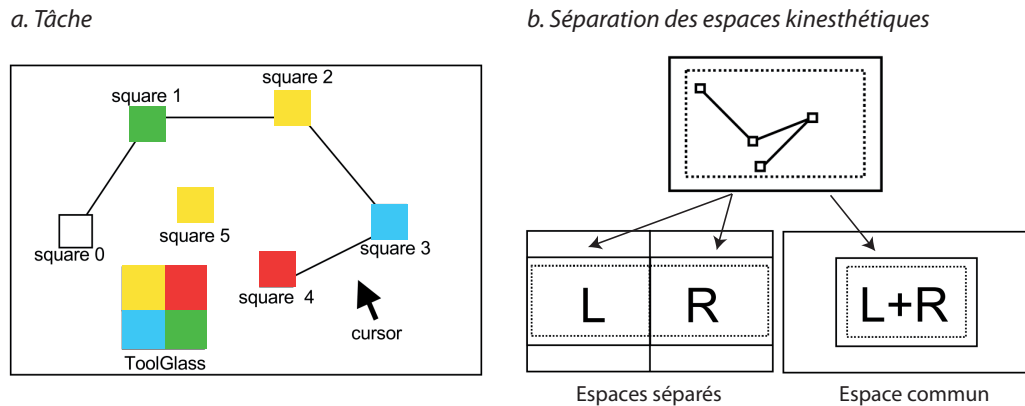


FIGURE 3.1 – Rôles des cadres de référence kinesthésiques [171] : *a.* La couleur à sélectionner à l'aide de la palette est celle de la cible à atteindre. *b.* L'affichage est associé de manière absolue soit à l'ensemble de la surface d'acquisition, soit à chacune des sous parties dédiées aux mains gauches et droites

en permanence, soit la palette est uniquement affichée lorsqu'elle est à proximité du curseur. Selon Balakrishnan et al., les résultats obtenus montrent que la modélisation de Guiard reste valable même si les deux mains n'évoluent pas dans le même espace kinesthésique, du moment qu'un retour visuel approprié est présent. Plus précisément, ils montrent que le retour visuel est suffisant à l'application de la modélisation de Guiard, mais non nécessaire. En effet, en cas d'absence de retour visuel, ils montrent que les principes de Guiard s'appliquent uniquement si les deux mains évoluent dans le même espace kinesthésique. En revanche, toujours en cas d'absence de retour visuel mais avec de repères kinesthésiques différents, les performances sont significativement dégradées.

Ceci est cohérent avec les résultats rapportés par Wickens et al. montrant que la vision domine les autres modalités perceptives, notamment en cas de conflit avec les sensations kinesthésiques [173]. L'expérience de Balakrishnan est également corroborée par les résultats de Hinckley et al. [174]. Ces derniers considèrent une tâche de positionnement de deux objets 3D via deux capteurs de position 3D, un dans chaque main, où ils demandent aux participants de placer une première fois les objets dans l'environnement, puis de retrouver la position de la main dominante sans retour visuel uniquement grâce aux sensations kinesthésiques de celle-ci ou alors de celles fournies par la main non-dominante. Ils montrent que l'usage des deux mains simultanément pour cette tâche fournit suffisamment d'indices kinesthésiques pour former un cadre de référence indépendant du retour visuel. Ils montrent également que l'usage des deux mains améliore l'estimation de la position de celles-ci alors qu'une interaction uni-manuelle ne reçoit des informations kinesthésiques que des sensations internes en termes d'articulations et est donc plus dépendante du retour visuel.

Ces différentes études mettent en évidence la robustesse du modèle de chaîne cinématique. Si l'on considère le cas de la peinture numérique sur modèle 3D, la navigation permet de définir la référence spatiale de la peinture numérique, tandis que la peinture numérique elle-même nécessite une forte précision. Une répartition évidente des rôles en interaction bi-manuelle asymétrique consiste donc à attribuer la peinture à la main dominante et la navigation à la main non-dominante.

### 3.1.3 Performances

La modélisation de Guiard a inspiré le design de nombreuses techniques d'interaction bi-manuelle asymétrique [175–184] ainsi que des comparaisons de performances entre techniques uni-manuelles et bi-manuelles : plusieurs études montrent que l'interaction bi-manuelle est plus efficace que son équivalent uni-manuel lorsqu'une adaptation est possible [42, 185–188]

Par exemple, Buxton et al. étudient une tâche de navigation par scrolling et sélection par pointage de mots dans un document textuel [186]. Ils comparent l'utilisation de techniques uni-manuelle et bi-manuelle pour cette tâche, et montrent que la technique bi-manuelle est significativement plus efficace en terme de temps d'exécution. Cependant, ils notent que globalement les utilisateurs ne parallélisent pas l'interaction. Ils interprètent l'amélioration de performance comme étant due à une meilleure efficacité des mouvements de la main dominante (pointage) et l'absence de temps de changement de mode dans le cas de l'interaction bi-manuelle.

Kabbash et al. comparent différentes version de la technique d'interaction *Toolglass* via une tâche similaire à celle déjà décrite à la figure 3.1a. Dans cette étude, la *Toolglass* est déclinée notamment en une version uni-manuelle sous forme de menu, en une version bi-manuelle à deux curseur (un pour la palette et un pour le tracé), et en une version bi-manuelle où la palette est positionnée par la main non dominante (technique originale) [180]. Il apparaît que la technique bi-manuelle originale est plus performante que les deux autres techniques. Cependant, la technique bi-manuelle à deux curseurs est la moins performante. Selon Kabbash, cela est dû à sa conception qui impose une division de l'attention visuelle du participant entre deux tâches distinctes (le tracé d'une part et la sélection dans la palette d'autre part) et par conséquent ne suit pas le modèle de chaîne cinématique. Kabbash en conclut que l'usage des deux mains pour une effectuer une tâche peut être pire que l'usage d'une seule main lorsque la répartition des rôles ne suit pas le modèle de chaîne cinématique.

Leganchuk et al. étudient les bénéfices *manuels* et *cognitifs* apportés par l'interaction bi-manuelle dans le cadre du modèle de chaîne cinématique [188]. Ils définissent les



bénéfices manuels comme ceux provenant d'une amélioration de l'efficacité du rapport mouvement/temps due au doublement du nombre de degrés de libertés utilisables simultanément. Les bénéfices cognitifs sont définis comme ceux provenant d'une réduction de la charge cognitive lorsque l'interaction uni-manuelle impose une décomposition artificielle à bas niveau d'abstraction de la tâche. Les auteurs étudient ces bénéfices sur une tâche de sélection rectangulaire à l'aide de techniques uni-manuelle et bi-manuelles. Ils montrent que les techniques bi-manuelles améliorent les performances en terme de temps d'exécution, et que le bénéfice augmente avec la difficulté de visualisation de la tâche.

Ces différentes études de performances concernent des interactions en espace bidimensionnel. Cependant, plusieurs systèmes explorent également l'interaction bi-manuelle dans le cadre de l'interaction 3D, que ce soit dans des environnements de bureau classiques [185, 189] ou des environnements immersifs [182].

Certains systèmes sont développés pour répondre à un besoin de *visualisation*. Hinckley et al. ont par exemple développé un système bi-manuel à destination des neurochirurgiens [190]. Ce système permet aux chirurgiens de visualiser une tranche spécifique du cerveau, définie à l'aide d'objets tangibles : une plaque transparente représente le plan de coupe et une boule représente le cerveau. En positionnant la plaque contre la boule, le chirurgien indique à l'ordinateur l'orientation et la position du plan de coupe. Hinckley et al. mettent en évidence à l'aide d'une étude qualitative que les participants préfèrent utiliser cette technique plutôt qu'une technique classique de visualisation de plan de coupe contrôlée à la souris. Cependant, comme cette technique répond à une tâche de visualisation, l'interaction de navigation est déjà répartie sur les deux mains. Cette technique n'est donc pas adaptée à un usage pour lequel la navigation représente seulement un *rôle* de l'interaction bi-manuelle.

D'autres systèmes répondent à des besoins de *manipulation* : Balakrishnan et al. étudient par exemple des tâches de sélection et manipulation de cible en 3D à plusieurs niveaux de difficulté, nécessitant à la fois une interaction de navigation 3D (contrôle caméra) et une interaction de manipulation 3D [185]. Ils comparent une interaction uni-manuelle à la souris permettant d'effectuer la navigation et la manipulation de manière séquentielle, avec une interaction bi-manuelle asymétrique à deux souris où la navigation est déléguée à la main non dominante et la sélection à la main dominante en parallèle. Ils montrent une amélioration de 20% du temps de sélection avec l'interaction bi-manuelle, pour une tâche de sélection facile. Dans le cas de la tâche difficile, l'amélioration est obtenue après entraînement des participants.

D'autres systèmes s'intéressent plus spécifiquement aux tâches de *modélisation* : Sachs et al. ont par exemple développé un système bi-manuel dédié aux applications de conception 3D [191]. Leur système comprend une paire de capteurs d'orientation et de position

respectivement montés sur une palette et un stylet. La palette est utilisée comme plan de référence pour la construction d'objets 3D, et permet également de modifier l'orientation et la position d'objets existants et de naviguer. Le stylet permet de dessiner et éditer les courbes des objets. Il est repéré relativement à la palette. Cependant, ce système n'est pas approprié à une utilisation de longue durée dans la mesure où il nécessite de tenir les dispositifs d'entrée en suspension, ce qui provoque rapidement une fatigue musculaire.

Lopes et al. considèrent également la combinaison du stylet et de l'interaction tactile dans le cadre de la modélisation 3D [192] : ils montrent que l'interaction bi-manuelle avec stylet en main dominante (tracé) et tactile en main non dominante (contrôle) est plus efficace que l'interaction uni-manuelle modale au stylet.

Plus généralement, d'autres systèmes ont été élaborés de manière à permettre l'usage des deux mains pour l'interaction 3D [177, 193–198], cependant aucun d'entre eux ne concerne la peinture numérique sur objet 3D.

Ces différentes études mettent en évidence une tendance à ce que l'interaction bi-manuelle, correctement élaborée, soit plus efficace que son équivalent uni-manuel. Cependant il est impossible de généraliser ces résultats d'une tâche à l'autre, et de tels résultats restent à montrer concernant la peinture numérique sur objet 3D, notamment dans le cadre d'un usage novice.

## 3.2 Choix des dispositifs d'entrée

La peinture numérique et la navigation 3D peuvent être effectuées à l'aide de nombreux dispositifs d'entrée différents (cf. chapitres 1 et 2). Cependant, la combinaison des deux dans le cadre de l'interaction de peinture numérique sur modèle 3D impose des contraintes quant à l'utilisation de ces dispositifs. Les contraintes dépendent en partie de la volonté de rendre cette interaction accessible à des utilisateurs novices, mais aussi de la répartition uni-manuelle ou bi-manuelle des rôles.

Nous allons considérer ces différentes contraintes, afin de déterminer les dispositifs d'entrée appropriés à notre contexte d'usage.

Comme on a pu le voir dans la partie précédente de ce chapitre, peinture et navigation peuvent être réalisées de façon uni-manuelle toute deux en main dominante, ou de manière bi-manuelle avec la peinture en main dominante et la navigation en main non dominante. Nous allons donc considérer les différents dispositifs d'entrée selon leur usage en main dominante ou non dominante.

### 3.2.1 Quels dispositifs pour un usage récréatif?

Les environnements de types immersifs ou semi-immersifs paraissent peu appropriés à une pratique récréative de la peinture numérique 3D dans la mesure où ils nécessitent des équipements onéreux et volumineux difficilement accessible aux particuliers. Nous focalisons notre étude sur les dispositifs d'entrée pouvant s'intégrer dans un environnement de bureau traditionnel :

- la souris,
- le stylet,
- les surfaces d'acquisition tactile multipoint,
- les dispositifs d'entrée à 6 degrés de liberté isométriques.

L'interaction tactile présente l'intérêt particulier d'augmenter le nombre de degrés de liberté en entrée par rapport à la souris ou au stylet, tout en restant accessible à un public novice et amateur : en effet, les surfaces tactiles sont désormais répandues auprès du grand public via les Smartphones et les tablettes. On peut d'ailleurs observer l'apparition d'applications d'édition 3D dédiées au grand public sur ces dispositifs [5, 199, 200].

Concernant les dispositifs d'entrée à 6 degrés de liberté, on peut considérer l'utilisation de dispositifs *isotoniques* ou *isométriques* (figure 3.2). Un dispositif est *isotonique* si la force qui lui est appliquée est constante [132]. On utilise un dispositif isotonique en le déplaçant dans l'espace, comme par exemple la souris. Dans le cas de déplacements dans l'espace 3D, Bérard et al. montrent que ce type d'interaction est imprécise et provoque une fatigue du bras [201]. Les dispositifs *isotoniques* ne sont donc pas adaptés à un usage prolongé que nécessite la peinture numérique 3D. De fait, le développement des dispositifs d'entrée à 6 degrés de liberté en industrie se fait principalement sous forme *isométrique*, c'est à dire que le dispositif est fixe et le contrôle est effectué via les variations des forces

a. Dispositif isotonique



b. Dispositif isométrique



FIGURE 3.2 – Dispositifs d'entrée à 6 degrés de liberté *isotonique* et *isométrique*

appliquées à celui-ci, à l'instar des joystick des consoles de jeux [132]. Nous allons donc dans la suite considérer l'usage de ce type de dispositifs, que nous nommerons *joystick 6dof* (6 degrees of freedom) pour plus de simplicité.

### 3.2.2 Répartition des dispositifs

#### 3.2.2.1 Main dominante

L'utilisateur peut être amené à effectuer différentes actions en main dominante selon l'aspect bi-manuel ou uni-manuel de l'interaction de peinture 3D considérée :

- soit à la fois la peinture et la navigation (interaction uni-manuelle)
- soit uniquement la peinture (interaction bi-manuelle)

Dans les deux cas, nous avons justifié le choix d'un stylet comme dispositif d'interaction pour le dépôt de peinture numérique (cf. chapitre 1). Cependant, dans le cas de l'interaction uni-manuelle, cela impose soit d'effectuer aussi la navigation à l'aide du stylet, soit d'utiliser alternativement un second dispositif d'entrée spécifique à la navigation.

Si la navigation est également effectuée au stylet, le changement de mode doit être explicite. La plupart des logiciels professionnels de création 3D proposent ce type d'interaction, avec des changements de mode accessibles via des boutons dans l'interface ou des raccourcis claviers.

Si la navigation est faite à l'aide d'un autre dispositif d'entrée, le changement de mode est implicite. La principale contrainte, dans ce cas, est d'avoir un coût de changement de mode minimal. Par exemple, il ne semble pas souhaitable d'utiliser conjointement une souris pour la navigation et un stylet pour la peinture, tous deux manipulés par la main dominante : le changement de dispositif serait alors trop coûteux. Le même problème apparaît si l'on utilise un joystick 6dof pour la navigation. En revanche, une combinaison intéressante est celle du tactile pour la navigation et du stylet pour la peinture, du moment que les surfaces d'acquisition sont confondues. En effet, dans la vie courante, il est habituel de coincer temporairement le stylo entre les doigts pour utiliser sa main dominante pour autre chose.

Par conséquent le stylet et les surfaces tactiles semblent être des dispositifs appropriés en main dominante pour la peinture numérique 3D, bien que cela nécessite d'être confirmé de manière expérimentale.

### 3.2.2.2 Main non dominante

Si l'interaction de peinture numérique 3D est sous forme bi-manuelle, la navigation est déléguée à la main non dominante (cf. section 3.1.2). Cela laisse donc plus de liberté que l'interaction uni-manuelle quant au choix du dispositif d'entrée.

L'interaction à la souris est conçue pour exploiter la précision de la main dominante, et s'accommode mal de la main non dominante. Ainsi, les premiers succès en conception d'interaction à deux mains mettent en oeuvre un trackball dans la main non dominante [172].

Plusieurs travaux considèrent l'usage d'un joystick 6dof en main non dominante pour la navigation dans le cadre d'interactions 3D [190, 191, 202]. Bien qu'il ne s'agisse pas de techniques de peinture 3D, ces travaux donnent bon espoir qu'un tel dispositif soit approprié à notre contexte. Ce cas mérite donc d'être étudié.

Concernant l'usage du stylet en main non-dominante, Brandl et al. étudient une tâche de dessin 2D en interaction bi-manuelle, où la main dominante dessine tandis que la main non dominante gère les contrôles de l'interface et la navigation [203]. Ils comparent différentes combinaisons de dispositifs d'entrée pour cette tâche, et montrent que la combinaison d'un stylet en main dominante et une interaction tactile pour la main non dominante est plus efficace que les deux alternatives testées : l'interaction tactile pour les deux mains, et l'interaction au stylet pour les deux mains.

Ces résultats sont cohérents avec la caractérisation des propriétés des entrées tactiles et stylet proposée par Hinckley et al. [204]. Cette caractérisation indique que le stylet est bien approprié aux "tracés", tandis que les *tap*, *hold*, et *drag* sont facilement réalisés aussi bien avec les doigts que le stylet. Les gestes multipoints tel que le geste de *pincer* sont quant à eux appropriés à l'interaction tactile. Hinckley recommande l'utilisation du stylet en main dominante pour le dessin et l'interaction tactile pour les commandes.

La combinaison d'une interaction au stylet en main dominante pour le tracé et d'une interaction tactile en main non dominante pour la navigation présente une forte ressemblance avec l'activité de dessin dans le monde physique : la main non dominante positionne la feuille tandis que la main dominante dessine. Hinckley et al. ont étudié le transfert des compétences physique de dessin et d'annotation vers le monde virtuel [205]. Ils proposent un ensemble d'interactions, couplant tactile et stylet, dérivées de l'observation des comportements d'un ensemble de participants durant une session de travaux manuels incluant des tâches d'annotations manuscrites et de tracé.

	Bimanuel		Unimanuel
	<i>Main dominante</i>	<i>Main non dominante</i>	<i>Main dominante</i>
<b>Peinture</b>	Stylet	x	Stylet
<b>Navigation</b>	x	Tactile / 6dof	Stylet / Tactile

TABLE 3.1 – Dispositifs d'entrée à étudier pour la peinture numérique 3D

Cependant, ces études portent uniquement sur des tâches de dessin 2D. L'adéquation de l'interaction tactile pour la navigation en main non dominante reste donc à étudier dans le cadre de la peinture sur modèle 3D par des utilisateurs novices.

### 3.2.2.3 Synthèse

L'ensemble des dispositifs d'entrée qui, au vu de l'état de l'art, paraissent appropriés à notre problématique de peinture numérique 3D en usage récréatif, est synthétisé dans le tableau 3.1. Cependant, l'utilisabilité de ces dispositifs dans ce cadre reste à être confirmée de manière expérimentale.

### 3.2.3 Choix des technologies

Le choix de la technologie adéquate pour chaque type de dispositif d'entrée peut être délicat. En effet, certains dispositifs d'entrée, comme le stylet et les surfaces tactiles, peuvent être implémentés de différentes manières. Cela conduit à des disparités concernant les propriétés de ces dispositifs, qui peuvent influencer sur la qualité de l'interaction.

Nous allons présenter dans cette partie la synthèse d'une revue des différentes technologies tactiles et de stylet disponible en annexe A.

#### 3.2.3.1 Stylet

Les qualités principalement attendues d'un stylet pour la peinture numérique, outre la précision de celui-ci, sont la détection de la pression et de l'inclinaison, qui permettent un contrôle artistique accru (cf. chapitre 1). Si l'on considère le tableau de synthèse 3.2, on peut voir que les technologies les plus intéressantes sont l'*ElectroMagnetic Resonance (EMR)*, la *Frustrated Total Internal Reflection (FTIR)*, et les grilles de capteurs de pression. Ces technologies sont précises et fournissent des informations sur la pression et l'inclinaison du stylet. La technologie *FTIR* permet également de détecter

Technologie	Précision	Pression	Inclinaison	Survol	Qualité optique	Tactile simultané
EMR	++++	✓	✓	✓	+++	x
Cap. passive	+	x	x	x	++	✓
Cap. active	+++	✓	x	✓	++	✓
Cap. Semi-active	++	✓	x	x	++	✓
Résistive	+	x	x	x	+	✓
Caméra	+++	✓	x	✓	++	x
FTIR	+++	✓	✓	x	++	✓
Magnétique	++	x	✓	✓		x
Capteurs de pression	+++	✓	✓	✓		✓
Infrarouge	+	✓	x	x	++	x

TABLE 3.2 – Synthèse comparative des technologies de stylet numérique

simultanément les contacts tactiles sans couche supplémentaire sur l'écran, ce qui évite d'aggraver les problèmes de parallaxe inhérents à l'interaction directe.

Cependant, la technologie *FTIR* ne fournit aucune information concernant le survol de la surface d'acquisition par le stylet. Elle est donc inutilisable en interaction indirecte. Les grilles de capteurs de pressions ne sont quant à elles pas encore commercialisée pour l'interaction directe (cf. A.1.5).

À l'inverse, la technologie *EMR* fournit des informations concernant le survol du stylet, ce qui la rend utilisable aussi bien en direct qu'en indirect. De plus, l'EMR n'altère pas la qualité optique de l'écran en interaction directe, car la surface d'acquisition se situe derrière celui-ci (cf. A.2.1).

L'EMR apparaît donc comme la technologie la plus appropriée pour la peinture numérique.

### 3.2.3.2 Tactile

L'EMR peut être combinée avec une couche de technologie tactile pour détecter les contacts des doigts. Il s'agit usuellement de la capacité passive mutuelle. Comme on peut le voir au tableau 3.3, cette technologie permet un bon rendu optique, est suffisamment précise pour une interaction tactile et permet la détection multipoint. De plus, elle ne provoque pas de faux contacts.

Technologie	Précision	Qualité optique	Parallaxe	Multipoint	Absence de faux contacts
Cap. passive	+	++	++	✓	✓
Cap. active	+++	++	++	✓	✓
Cap. Semi-active	++	++	++	✓	✓
Résistive	+	+	+	x	✓
FTIR	+++	++	++	✓	✓
Capteurs de pression	+++			✓	✓

TABLE 3.3 – Synthèse comparative des technologies tactiles

La combinaison des entrées de type stylet et tactile sur une même surface d'acquisition pose cependant le problème d'activation non désirée du tactile : il faut éviter de générer des événements tactiles lorsque l'utilisateur pose la paume de sa main dominante sur le dispositif pour stabiliser son poignet et contrôler le stylet par des mouvements de doigts. Annett et al. [206] proposent une revue des algorithmes permettant d'ignorer ces événements non désirés.

Ainsi, la combinaison EMR et capacité mutuelle est particulièrement appropriée à la peinture numérique 3D. Les expérimentations réalisées dans cette thèse s'appuient donc sur ces technologies qui sont mises en oeuvre dans l'écran interactif que nous avons utilisé<sup>1</sup>.

### 3.3 Optimisations

Certaines techniques ont pour but non pas de faciliter l'interaction de navigation ou de peinture elle-même, mais d'optimiser le temps qui leur est consacré.

Igarashi et al. ont proposé une interaction appelée "laser paint" [35]. Il s'agit d'une variante de peinture par projection où la peinture traverse le modèle 3D de manière à colorer toutes les surfaces pointées par le curseur (figure 3.3). Ainsi, les faces cachées sont peintes en même temps que les faces visibles. Igarashi a constaté, dans une étude utilisateur où le *laser paint* est proposé comme outil supplémentaire au projection painting, que cette technique est "très utile" et que "tous les participants l'ont utilisée intensivement au cours de l'étude". Cette technique permet en effet de réduire le temps de peinture et de navigation en peignant plusieurs zones à la fois.

1. Wacom 24HD Touch



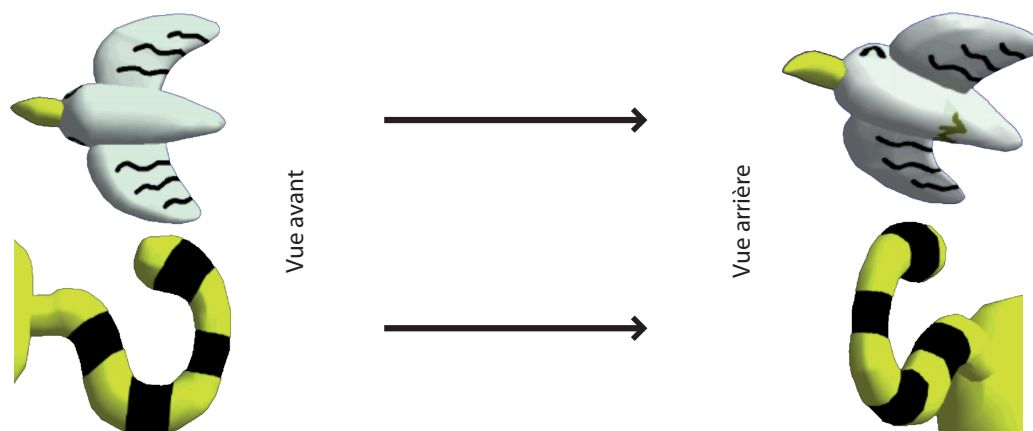


FIGURE 3.3 – Laser Paint [35] : Quand l'utilisateur peint en mode laser (à gauche), les coups de pinceau traversent le modèle et sont visible des deux cotés (à droite).

Fu et al. ont développé le système *LayerPaint* qui permet non seulement de peindre les faces visibles d'un modèle, mais également les faces cachées suivant leur niveau de profondeur [207]. Ce système permet d'atteindre aisément des zones inaccessibles (intérieur d'un modèle), ou des zones pour lesquelles il est difficile de trouver un point de vue adéquat à cause de la topologie environnante. Pour cela, le système partitionne automatiquement le modèle en fonction de l'ordre d'occultation de chaque point de la surface suivant l'angle de vue courant. L'utilisateur peut ensuite choisir le niveau d'occultation sur lequel il veut peindre. Plusieurs types d'interactions sont disponibles dans *LayerPaint* :

**Layer-Aware Painting** : cette technique permet de continuer automatiquement à peindre dans la même zone, même si celle-ci est cachée par une zone plus proche (figure 3.4a).

**Interactive Region Pop-Up** : l'utilisateur peut sélectionner une zone pour afficher la surface cachée par celle-ci, si elle existe (figure 3.4b). Plusieurs sélections successives permettent d'itérer de manière cyclique sur l'ensemble des surfaces cachées au niveau de cette sélection.

**Interactive Paint-to-hide** : L'utilisateur peut redéfinir le partitionnement du modèle à l'aide de l'outil de peinture. Une fois la nouvelle partition validée, les zones peintes sont cachées (figure 3.4c).

**Layer-Aware Object Rotation** : L'utilisateur peut définir un point appartenant à une face cachée comme point de pivot pour les rotations de navigation (figure 3.4d).

La technique *LayerPaint* permet d'optimiser la stratégie de peinture de l'utilisateur : ce dernier passe moins de temps à naviguer pour atteindre les différentes zones du modèle

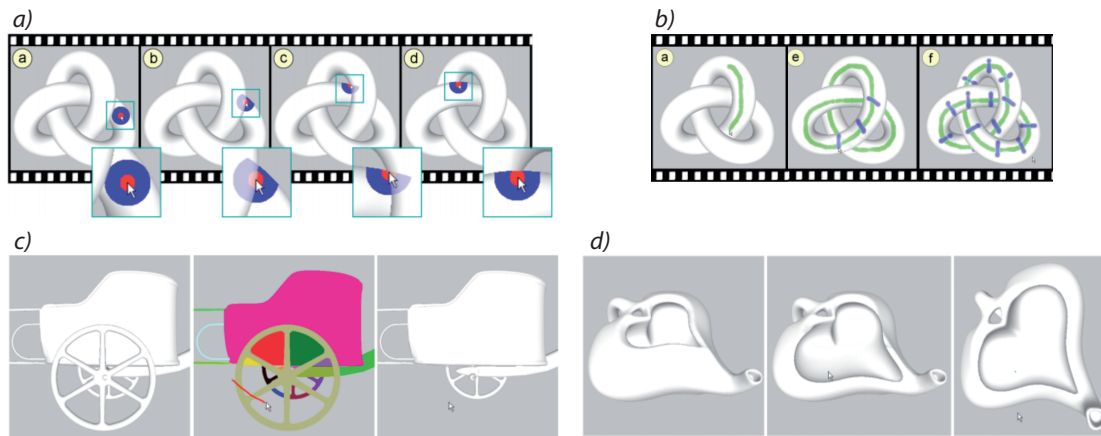


FIGURE 3.4 – Layer Paint [207] : *a)* La zone à peindre reste sur la même surface. *b)* L'utilisateur peut afficher les régions cachées à l'aide d'un click puis peindre celles-ci. *c)* L'utilisateur peut peindre dans la segmentation pour cacher certaines zones. *d)* L'utilisateur peut définir un point de pivot caché.

3D. De plus, dans le cas de modèles complexes avec de nombreuses zones d'occlusion, cette technique apporte un meilleur confort de peinture : elle permet de choisir un point de vue adapté qui n'est pas gêné par les zones plus proches. Cette technique a cependant certaines limitations : elle n'est pas applicable aux modèles qui s'auto-intersectent. De plus, elle ne fonctionne pas sur des modèles bruités ou comprenant des détails à haute fréquence car ils font exploser le nombre de niveaux d'occultation.

Ortega et al. ont mis au point une approche plus simple permettant de travailler efficacement sur des modèles complexes : lorsque l'utilisateur peint une surface donnée, le curseur reste associé à cette surface, même si une autre partie du modèle vient occulter la zone en cours de peinture (figure 3.5). La zone d'occultation devient alors transparente pour que l'utilisateur puisse continuer à peindre la surface sous-jacente.

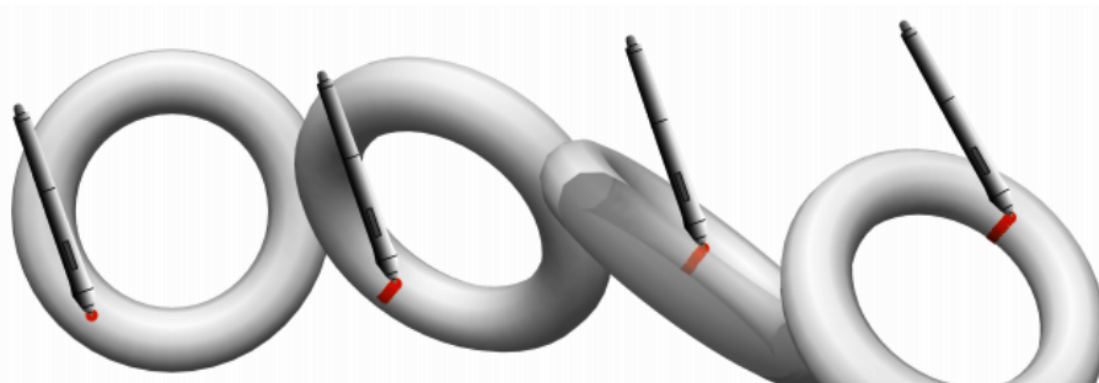


FIGURE 3.5 – ACCD [160] : Le curseur reste sur la surface en cours de peinture. Lorsque cette surface est occultée, la zone d'occultation devient transparente.

## Bilan

Nous avons pu voir dans ce chapitre que l'interaction de peinture sur modèle 3D, si elle est déclinée sous forme bi-manuelle, se classe dans la catégorie des interactions bi-manuelles asymétriques. Le modèle de chaîne cinématique nous indique alors que la répartition la plus appropriée pour cette tâche est de peindre avec la main dominante et de naviguer avec la main non dominante.

De plus, nous avons pu voir que l'interaction bi-manuelle pour la 3D a été essentiellement étudiée selon les axes suivants : pour la navigation uniquement (tâche de visualisation), pour combiner la navigation et la manipulation, et pour combiner la navigation et la modélisation. L'efficacité d'une combinaison bi-manuelle des rôles de navigation et de peinture pour une tâche de peinture numérique sur objet 3D reste donc à étudier, notamment dans le cas d'utilisateurs novices.

Concernant le choix des dispositifs d'entrée, nous avons proposé dans le tableau 3.1 une répartition en fonction de l'usage uni-manuel ou bi-manuel qui nous paraît appropriée à la vue de l'état de l'art. Cependant, cette répartition doit être évaluée de manière expérimentale.

Nous proposons donc au chapitre 6 une évaluation comparative de différentes formes d'interactions bi-manuelles et uni-manuelles pour la peinture 3D, prenant en compte notre proposition de périphériques d'entrée.



# Bilan de l'état de l'Art

Cet état de l'art nous ouvre plusieurs pistes de recherches concernant les trois axes principaux que nous avons pris en compte, à savoir la peinture numérique, la navigation 3D, et la combinaison de la peinture avec la navigation. Pour ces trois axes, nous focalisons notre étude sur un usage approprié à un public novice.

Concernant la peinture numérique, nous avons pu constater que l'interaction indirecte est plus difficile à appréhender que l'interaction directe, mais elle présente des avantages en terme de confort. Cependant, nous ne connaissons pas, à ce jour, d'étude permettant de quantifier les bénéfices et inconvénients des deux approches. Ce sera l'objet de notre première contribution.

Nous avons également pu voir qu'en matière de navigation 3D, les interactions *Rotate/Pan/Zoom* permettent de manipuler librement la caméra virtuelle pour trouver le point de vue le mieux adapté aux contraintes ergonomiques des gestes de dessin. Cependant, même si la partie *Pan/Zoom* de l'interaction est intuitive, la partie *Rotate* reste complexe à appréhender, notamment pour des utilisateurs novices. Nous présenterons une seconde contribution visant à développer des techniques de rotations adaptées à un usage novice.

Enfin, nous nous sommes appuyé sur l'état de l'Art pour considérer différentes manières de combiner les interactions de peinture et de navigation 3D. Mais cette étude a fait apparaître le manque de connaissances empiriques sur la qualité de ces combinaisons. Ce sera l'objet de notre troisième contribution.



Deuxième partie

**Contributions**





# Introduction

Les travaux de recherches que nous avons réalisés dans cette thèse ont pour objectif d'étudier et d'améliorer les interfaces de peinture numérique sur modèle 3D. Suite à la présentation de l'état de l'Art du domaine, nous allons maintenant exposer nos contributions. La structure de cette partie du document fait écho à celle de la partie précédente : nous présentons nos trois contributions principales, dédiées respectivement à (1) la peinture numérique, (2) la navigation 3D, et (3) la combinaison des deux.

Chacun de ces trois sujets a donné lieu à une validation expérimentale fondée sur des expérimentations utilisateurs avec des participants novices. Dans toutes les expérimentations, nous avons utilisé le même écran interactif sensible au touché et au stylet<sup>2</sup> de manière à assurer la cohérence des résultats entre les études.

---

2. Wacom Cintiq 24 HD Touch



## Chapitre 4

# Influence de l'aspect direct de l'interaction sur la peinture numérique

### Sommaire

---

<i>Introduction</i> . . . . .	<b>76</b>
<b>4.1 Évaluation expérimentale</b> . . . . .	<b>77</b>
4.1.1 Matériel . . . . .	77
4.1.2 Tâches . . . . .	77
4.1.3 Conditions . . . . .	79
4.1.4 Protocole . . . . .	79
4.1.5 Mesures . . . . .	80
<b>4.2 Résultats</b> . . . . .	<b>81</b>
4.2.1 Temps de réalisation . . . . .	82
4.2.2 Erreurs d'orientation . . . . .	83
4.2.3 Variabilité d'orientation . . . . .	83
4.2.4 Points d'inflexion . . . . .	84
4.2.5 Évaluation subjective . . . . .	84
<b>4.3 Discussion</b> . . . . .	<b>85</b>
4.3.1 Direct vs. indirect . . . . .	85
4.3.2 Effet du gain dans le cadre de techniques indirectes . . . . .	86
<i>Bilan</i> . . . . .	<b>87</b>

---

## Introduction

La peinture numérique est considérée par les utilisateurs comme plus difficile à appréhender en interaction indirecte qu'en interaction directe (cf. section 1.2.2). Cependant, même si nous pouvons lister des avantages et inconvénients de ces deux configurations en terme d'ergonomie, à notre connaissance il n'existe pas d'étude objective sur l'effet de l'aspect direct de l'interaction sur la peinture numérique. Dans ce chapitre, nous présentons une étude qui vise à comprendre l'effet de chaque configuration sur la peinture numérique, et plus précisément à étudier l'influence de deux composantes fondamentales sur la qualité de la production finale : le gain entre le contrôle et l'affichage et la dissociation oeil-main (cf. section 1.2.3).

La modification du gain présente des avantages. Un gain supérieur à 1 permet par exemple de concevoir des dispositifs moins encombrants car un grand déplacement virtuel se traduit par un petit déplacement dans l'espace du dispositif. De même, on peut envisager une moindre fatigue musculaire de l'utilisateur car les tracés virtuels sont plus amples que les tracés physiques. Cependant, cette modification peut influencer sur la qualité du tracé puisque tout défaut du tracé est amplifié.

La peinture numérique est une interaction différente de l'interaction de pointage, plus commune en IHM, dans la mesure où elle nécessite une connaissance permanente de la direction du mouvement du curseur afin d'éviter les erreurs et les ajustements de trajectoires (cf. section 1.2.2). Or, en interaction indirecte cette information est inférée *approximativement* à partir des positions précédentes du curseur (cf. section 1.2.2). On peut donc supposer que des erreurs dans cette inférence puissent conduire à davantage d'ajustements et donc à une moins bonne qualité de la trajectoire, comparativement à l'interaction de peinture directe.

Par conséquent, nous cherchons à mettre en évidence les erreurs de trajectoires et d'ajustements en peinture indirecte à l'aide d'une étude expérimentale. Celle-ci va se concentrer sur un élément de base de la peinture : le dessin de lignes droites à main levée. Ceci nous permettra de mieux cerner l'effet de l'aspect direct de l'interaction sur les variations d'orientation.

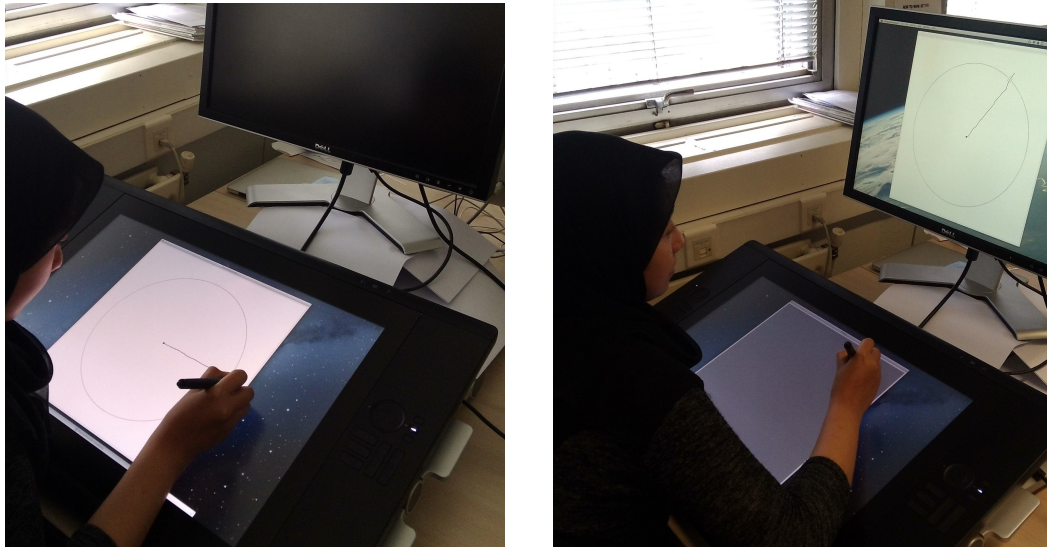


FIGURE 4.1 – Interaction directe vs. interaction indirecte : en interaction directe (à gauche) le participant interagit uniquement avec l'écran interactif. En interaction indirecte (à droite), il dessine sur l'écran interactif tout en regardant l'écran standard

## 4.1 Évaluation expérimentale

### 4.1.1 Matériel

L'expérience a été effectuée sur un ordinateur portable comprenant un processeur Intel Core i7, 2,6Ghz, ainsi qu'une carte graphique NVidia GT 650M. L'expérience repose sur une application programmée en Objective-C à l'aide du framework Cocoa. Nous avons utilisé un écran standard de 24 pouces à sa résolution native de 1920x1200 pixels, ainsi qu'un écran interactif de 24 pouces à la même résolution (Wacom Cintiq 24HD touch, avec la dalle tactile désactivée).

Les participants ont utilisé le stylet fourni avec la Wacom Cintiq (Wacom Grip Pen) sur l'écran interactif dans les conditions directes et indirectes. Les traits générés par les participants étaient affichés sur l'écran interactif dans la condition directe, et sur l'écran standard dans la condition indirecte (figure 4.1). La sensibilité à la pression du stylet était désactivée pendant l'expérience.

### 4.1.2 Tâches

Nous avons opté pour une tâche impliquant un comportement de *target locking*. En effet, nous avons vu précédemment dans l'état de l'art qu'il s'agit du comportement naturel de la main et des yeux dans une tâche de dessin traditionnelle [32]. Selon Tchalenko, le

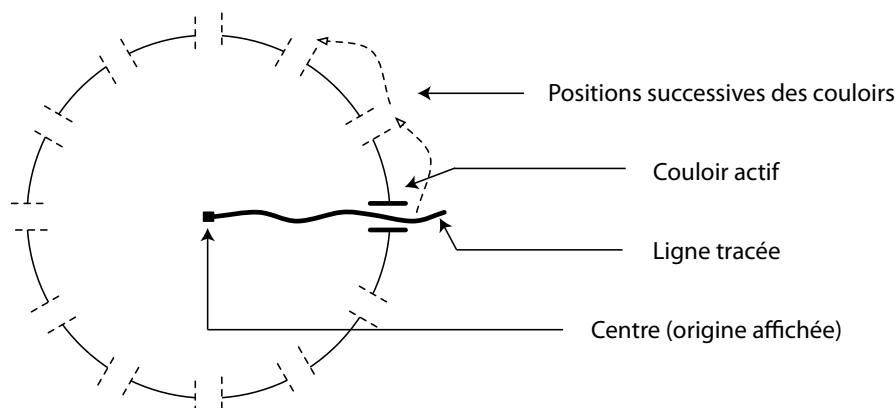


FIGURE 4.2 – La tâche : les participants doivent tracer une ligne droite depuis le centre du cercle vers l’extérieur en traversant le couloir. Le couloir est successivement affiché à douze emplacements différents sur le cercle, dans le sens contraire des aiguilles d’une montre.

*target locking* peut être provoqué en demandant aux participants de dessiner une ligne depuis une origine affichée vers une cible suggérée mais non affichée [33].

Nous avons donc présenté aux participants un cercle de 12 cm de rayon dont le centre était affiché à l’aide d’un petit carré. Celui-ci faisait office d’origine affichée. Les participants ont tracé des rayons depuis ce centre vers l’extérieur du cercle, en passant par le milieu de couloirs situés en périphérie du cercle. Les intersections entre l’axe central des couloirs et le cercle jouaient le rôle de cibles non affichées. Les couloirs étaient successivement présentés à douze emplacements différents, dans le sens contraire des aiguilles d’une montre (Figure 4.2).

Nous avons choisi d’afficher les couloirs parallèlement au rayon demandé afin d’encourager le dessin de lignes droites. En effet, une orientation différente du couloir peut influencer sur la trajectoire effectuée par le participant [208].

Nous avons demandé aux participants d’effectuer la tâche selon deux *styles de dessin* différents :

- En *dessin précis*, les participants avaient pour instruction de tracer les lignes droites soigneusement et précisément, en ignorant toute contrainte de temps.
- En *dessin balistique*, les participants avaient pour instruction de tracer les lignes en optimisant le compromis vitesse/précision : aller le plus vite possible tout en dessinant une droite la plus droite possible. Il s’agit d’un geste typique des tâches de croquis rapides.

### 4.1.3 Conditions

En combinant les variations de gain et d'affichages, nous avons demandé aux participants de réaliser les tâches dans quatre conditions différentes, que nous appellerons également techniques d'interaction :

**Direct** Le cercle est affiché sur l'écran interactif. Les participants interagissent directement sur l'écran (Figure 4.1, à gauche). Le gain est réglé à 1 dans les deux dimensions, et le tracé apparaît directement sous le stylet.

**Indirect** Le cercle est affiché sur l'écran standard. Le stylet est alors utilisé sur l'écran interactif dont l'affichage est réglé de manière à montrer une zone neutre de couleur grise où le curseur ne s'affiche pas. (Figure 4.1, à droite). Le gain est réglé à 1 dans les deux dimensions : le curseur virtuel parcourt sur l'écran standard la même distance physique que le déplacement du stylet sur l'écran interactif.

**Homogène** Cette interaction est similaire à INDIRECT, mais elle utilise un gain égal à 2 dans les deux dimensions : Le curseur parcourt deux fois sur l'écran standard la distance physique parcourue par le stylet.

**Hétérogène** Cette interaction est similaire à INDIRECT, mais elle utilise un gain différent selon l'axe : celui-ci est de valeur 2 sur l'axe horizontal et de valeur 1 sur l'axe vertical.

L'étude du gain HÉTÉROGÈNE est justifiée par la présence encore très importante chez les graphistes de configurations dans lesquelles le ratio de la tablette graphique n'est pas égal à celui de l'écran. Une configuration classique est la combinaison d'une tablette à ratio 4/3 avec un écran 16/9.

### 4.1.4 Protocole

Nous avons recruté 16 participants droitiers, âgés de 41 ans en moyenne ([25-65], std-dev=15.4). Tous ces participants étaient novices en peinture numérique.

L'expérience a eu lieu dans une pièce silencieuse. Les participants étaient assis devant l'écran interactif, incliné à 30° de l'horizontale (Figure 4.1). L'écran standard était placé derrière l'écran interactif, aligné avec celui-ci de manière à faire face au participant. Au début de chaque session d'expérimentation, le chercheur expliquait l'ensemble de la procédure au participant.

Nous avons utilisé un design expérimental de type intra-sujet : chaque participant a dessiné des rayons selon les douze angles différents, pour chacune des quatre techniques

d'interaction (DIRECT, INDIRECT, HOMOGENÈ, HÉTÉROGÈNE). Chaque technique a été testée en dessin précis et dessin balistique, avec 8 répétitions. Nous avons pu enregistrer en tout : 16 (participants) x 4 (techniques) x 2 (styles de dessin) x 12 (angles) x 8 (répétitions) = 12288 trajectoires. Chaque nouvelle combinaison de technique et style de dessin commençait par une phase d'entraînement dans laquelle le participant effectuait 24 essais (2 séries de 12 angles).

L'ordre de présentation des styles de dessin a été équilibré entre 2 groupes de 8 participants. À l'intérieur de chaque groupe, nous avons également équilibré l'ordre de présentation des techniques.

Les participants ont ensuite noté les techniques selon leurs préférences durant un entretien post-expérimental. Les notes sont comprises entre 1 et 4, où 4 est la meilleure note. Les participants ont également pu commenter librement les différentes interactions.

#### 4.1.5 Mesures

Afin d'observer un potentiel effet des conditions sur la performance et la qualité du dessin, nous avons tout d'abord relevé le temps de réalisation des traits (*Completion Time CT*). Il s'agit du temps écoulé entre l'évènement de premier contact et l'évènement de dernier contact du stylet. Nous avons également relevé 3 mesures de *qualité de dessin* qui se concentrent sur l'orientation des traits :

**Orientation Error (OE)** : Il s'agit de l'écart entre *l'orientation globale du trait* et celle de la prescription (ie. le rayon idéal passant à travers le centre du couloir). Cette mesure est une simplification de celle proposée par Keefe et al. [45]. OE est calculé à l'aide des étapes suivantes, qui sont illustrées à la figure 4.3. Tout d'abord, les événements stylet sont ré-échantillonnés de manière à ce que deux évènements successifs soient au moins à une distance de 10 pixels l'un de l'autre. Ceci fournit une succession de petits segments de droite. On utilise alors la valeur médiane de l'orientation de tous les segments en tant qu'orientation globale du trait d'origine. On définit ensuite OE comme la valeur absolue de la différence entre cette orientation globale et celle de la prescription. Cette approche est plus robuste aux valeurs aberrantes que la régression linéaire.

**Orientation Variability (OV)** : Il s'agit de la quantité de variation d'angle pendant le tracé du trait. Cette mesure utilise le même ré-échantillonnage que OE. Pour chaque segment, on calcule l'écart relatif entre son orientation et celle de la prescription. On définit alors OV comme la valeur absolue de la différence entre les quantiles à 10% et 90% de la distribution des écarts (figure 4.3).



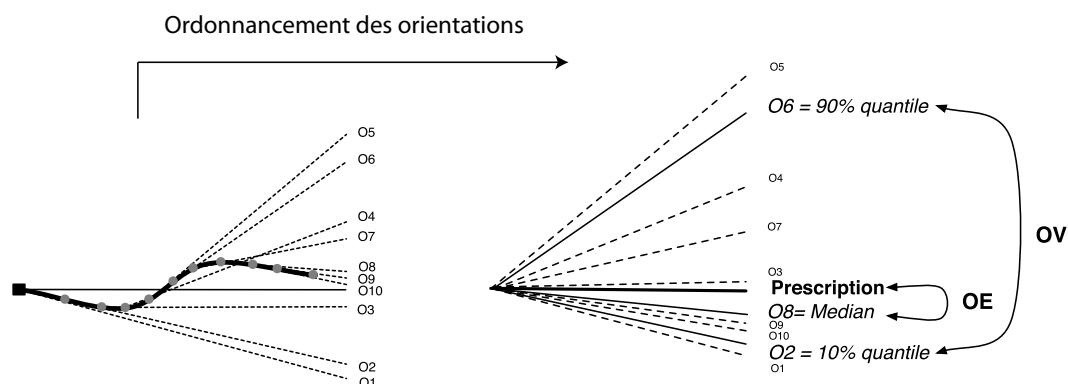


FIGURE 4.3 – Calcul de OE et OV : Après le ré-échantillonnage du trait en petits segments, les orientations de chaque segment sont ordonnées. OE est la valeur absolue de la différence entre l'orientation de la prescription et la médiane des orientations. OV est la valeur absolue de la différence entre les quantiles à 10% et 90%.

**Number of Inflection points (NI)** : Il s'agit du nombre de points d'inflexion du trait tracé, (ie. les points de changement de courbure du tracé)

## 4.2 Résultats

Nous avons étudié dans un premier temps l'effet de *l'angle* de l'emplacement du couloir sur OE, OV, NI et CT. Sur aucune des mesures nous n'avons trouvé d'effet significatif. Ceci peut être expliqué par le fait que nous n'avons pas contraint la posture du participant pendant l'expérience. En effet, chaque participant pouvait ajuster sa position afin d'obtenir le geste le plus confortable dans chaque orientation cible. Les contraintes anatomiques que peuvent imposer certaines positions n'étaient donc pas ou peu présentes.

Par conséquent, pour chaque mesure OE, OV, NI et CT, et pour chaque combinaison *participant/style de dessin/technique*, nous avons agrégé les valeurs des différents angle selon la médiane pour les calculs ultérieurs.

Des test de Shapiro ont révélé que la distribution des médianes selon les participants de OE, OV, NI et CT ne suivaient pas une distribution de loi normale. Nous avons donc considéré l'emploi de tests statistiques non-paramétriques. L'effet des facteurs a été déterminé à l'aide de tests de Friedman, et les comparaisons post-hoc avec des tests de Wilcoxon avec correction de Bonferroni.

Les résultats sont présentés à la figure 5.8. La superposition de tous les traits dessinés par les participants est présentée à la figure 4.5.

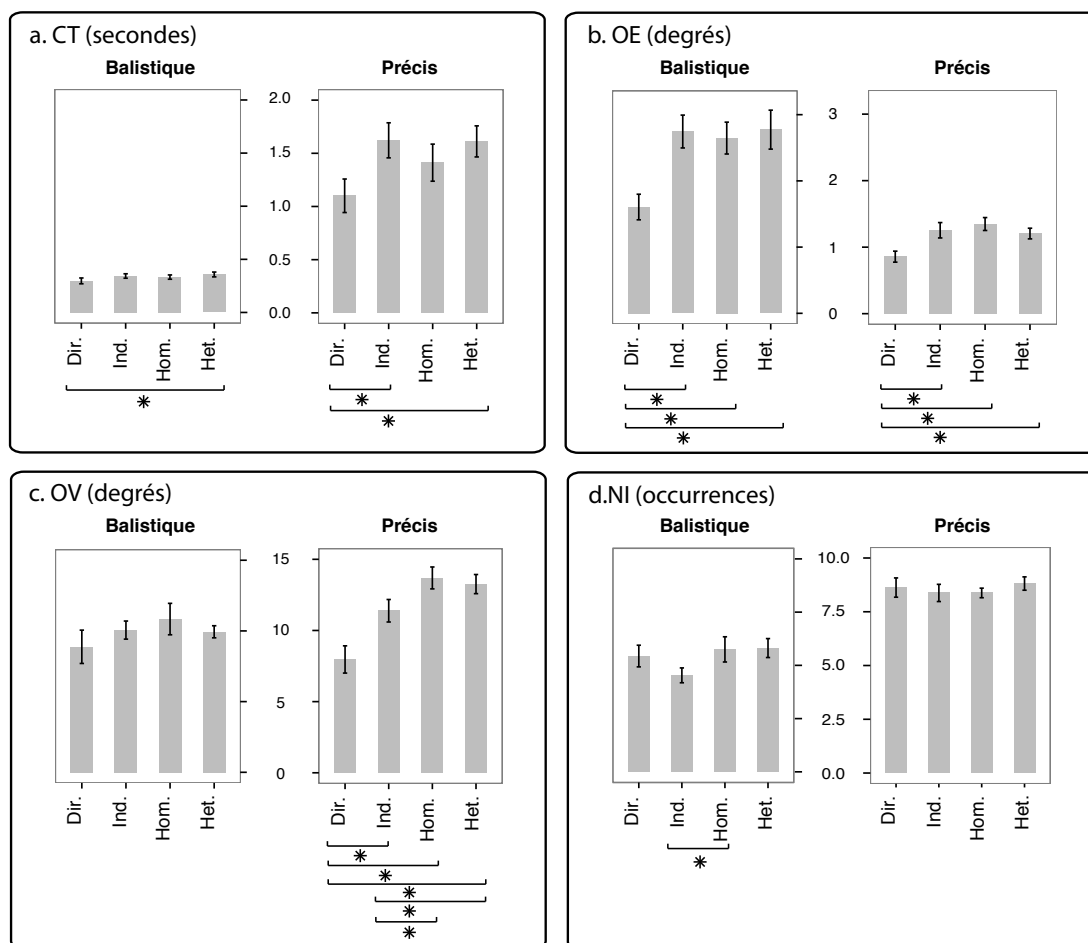


FIGURE 4.4 – Résultats (moyennes et 95% CI) : a) Temps (CT, en secondes), b) Erreur d'orientation (OE, en degrés), c) Variabilité d'orientation (OV, en degrés) et d) Nombre de points d'inflexion (NI, en occurrences) pour les quatre techniques, pour le dessin balistique (à gauche) et le dessin précis (à droite)

#### 4.2.1 Temps de réalisation

Les temps de réalisation (CT) sont représentés à la figure 5.8a. Un test de Friedman a révélé un effet global significatif de la *technique d'interaction* sur CT, à la fois pour les styles de dessin précis et balistique (respectivement  $p=1.42e-03$  et  $p=5.21e-03$ ). Les tests post-hoc ont révélé que (1) pour le dessin balistique CT est significativement plus court avec DIRECT qu'avec HÉTÉROGÈNE ( $p < 0.05$ ) avec une amélioration de 17%, et (2) en dessin précis CT est significativement plus court avec DIRECT que INDIRECT ou HÉTÉROGÈNE ( $p < 0.01$ ) avec des améliorations respectives de 55% et 54%.

Nous n'avons pas trouvé de différence significative de CT entre les techniques d'interaction indirectes, que ce soit pour le dessin balistique ou précis.

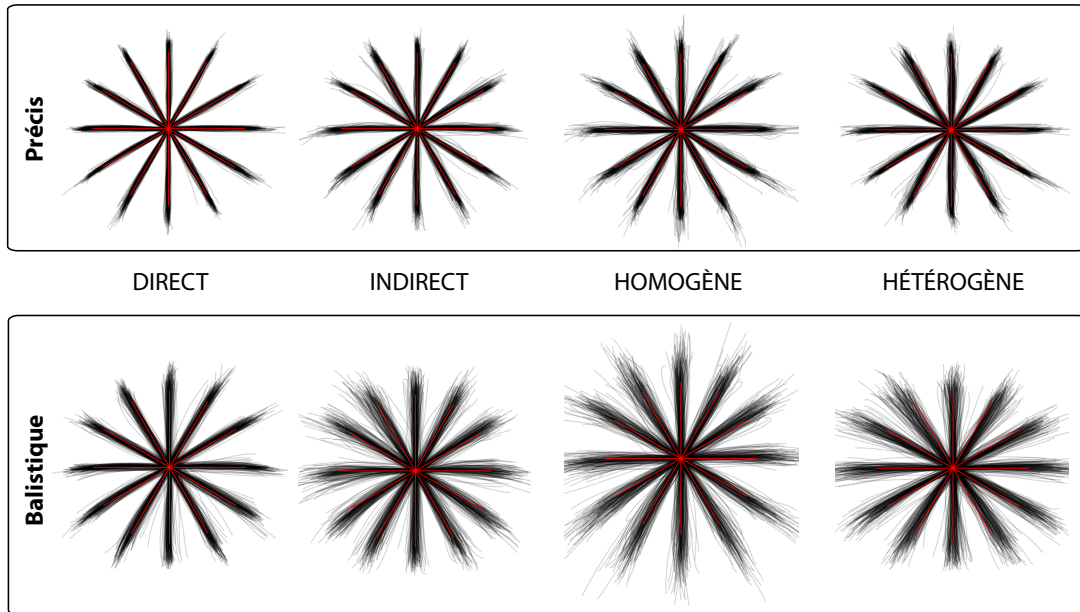


FIGURE 4.5 – Lignes tracées par les participants dans les conditions DIRECT, INDIRECT, HOMOGÈNE et HÉTÉROGÈNE, pour les styles de dessin précis et dessin balistique. On observe de plus grands secteurs d'angles, plus d'irrégularités, et donc une moins bonne qualité de dessin, avec les techniques indirectes qu'avec DIRECT, dans chaque style de dessin.

#### 4.2.2 Erreurs d'orientation

Les erreurs d'orientation (OE) sont présentées à la figure 5.8b. Un test de Friedman a révélé un effet global significatif de la *technique d'interaction* sur OE à la fois pour les styles de dessin précis et balistique (respectivement  $p=2.09e-05$  et  $p=1.56e-05$ ). Les tests post-hoc ont révélé que (1) pour le dessin balistique OE est significativement plus petit avec DIRECT que INDIRECT, HOMOGÈNE ou HÉTÉROGÈNE ( $p < 0.001$ ) avec des améliorations respectives de 43%, 38% and 40% et (2) en dessin précis OE est significativement plus petit avec DIRECT que INDIRECT, HOMOGÈNE and HÉTÉROGÈNE ( $p < 0.001$ ) avec des améliorations respectives de 32%, 37%, and 29%.

Nous n'avons pas trouvé de différence significative de OE entre les techniques d'interaction indirectes, que ce soit pour le dessin balistique ou précis. Ceci peut être observé à la figure 4.5 : les lignes tracées avec les techniques indirectes emplissent des secteurs plus larges que celles dessinées avec DIRECT.

#### 4.2.3 Variabilité d'orientation

Les erreurs de variabilité d'orientation (OV) sont présentées à la figure 5.8c. Un test de Friedman a révélé un effet global significatif de la *technique d'interaction* sur OV

pour le style de dessin précis ( $p=7.22e-08$ ) mais pas pour le style de dessin balistique. Les tests post-hoc ont révélé que pour le dessin précis (1) OV était significativement plus faible avec DIRECT que INDIRECT, HOMOGÈNE ou HÉTÉROGÈNE ( $p < 0.001$ ) avec des améliorations respectives de 30%, 42% et 38% et (2) OV était significativement plus faible avec INDIRECT que HOMOGÈNE ou HÉTÉROGÈNE ( $p < 0.05$ ) avec des améliorations respectives de 17% et 14%.

Nous n'avons pas trouvé de différence significative entre HOMOGÈNE et HÉTÉROGÈNE. Ceci peut être observé sur les irrégularités des traits de la figure 4.5 : il y a plus d'irrégularités sur les lignes dessinées avec les techniques indirectes que celles dessinées avec DIRECT.

#### 4.2.4 Points d'inflexion

Les nombres de points d'inflexion (NI) sont présentés à la figure 5.8d. Un test de Friedman a révélé un effet global significatif de la *technique d'interaction* sur NI en dessin balistique ( $p=2.63e-03$ ) mais pas en dessin précis. Les tests post-hoc ont révélé que pour le dessin balistique NI était significativement plus faible avec INDIRECT que HÉTÉROGÈNE ( $p < 0.01$ ) avec une amélioration de 25%.

#### 4.2.5 Évaluation subjective

Les préférences des participants ont été représentées par un score dans l'ensemble {1, 2, 3, 4}, 4 étant la technique préférée. Les résultats sont présentés à la figure 4.6.

Un test de Friedman (*rank sum test*) a révélé qu'il y a un effet significatif de la technique d'interaction sur les notations des participants ( $p=5.85e-07$ ). Nous avons effectué un test de Wilcoxon (*pairwise rank sum test*) avec correction de Bonferroni. Ce dernier révèle que les participants ont noté l'interaction DIRECT significativement mieux que

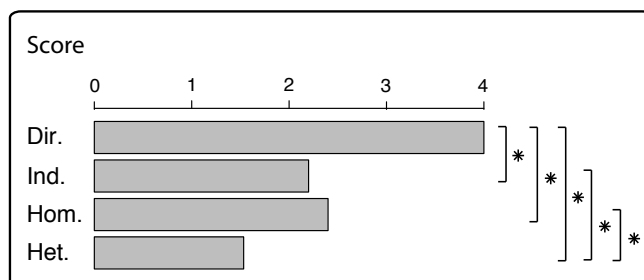


FIGURE 4.6 – Préférences des utilisateurs. Plus le score est élevé, plus les participants ont préféré la technique.

chacune des autres techniques ( $p < 0.001$ ). De plus, nous avons pu noter que DIRECT a été unanimement notée comme meilleure technique par tous les participants. Le test indique également que INDIRECT et HOMOGÈNE ont été notées significativement mieux que HÉTÉROGÈNE ( $p < 0.05$ ).

## 4.3 Discussion

### 4.3.1 Direct vs. indirect

Notre expérience montre que l'interaction directe à l'aide d'écrans interactifs est plus appréciée que l'interaction indirecte des tablettes graphiques concernant le tracé de lignes orientées par des utilisateurs novices. Les participants étaient en général plus rapides avec la technique d'interaction directe, tout en effectuant une orientation générale des traits plus précise, avec moins de variations de l'orientation. D'un point de vue subjectif, les participants ont clairement exprimé leur préférence pour l'interaction directe, chacun d'entre eux lui donnant la meilleure note. Certains participants ont eu l'impression de parcourir des distances plus grandes en INDIRECT qu'en DIRECT.

Afin de mieux cerner l'effet de la dissociation oeil-main en interaction indirecte, nous nous concentrons maintenant sur les résultats des techniques DIRECT et INDIRECT, qui utilisent toutes les deux un gain de 1, et nous ignorons temporairement les résultats de HOMOGÈNE and HÉTÉROGÈNE. Cette analyse est résumée dans la table 4.1.

Dans le cas du dessin balistique, les participants ont effectué 43% d'erreur d'orientation (OE) de moins en DIRECT qu'en INDIRECT, mais nous n'avons pas trouvé de différence significative concernant OV, NI et CT. Notre interprétation est la suivante : le dessin balistique est contraint dans le temps, ce qui explique l'absence de différence significative pour CT. De plus, ce style de dessin laisse moins d'opportunités aux participants pour ajuster leur geste. Peu de corrections implique logiquement peu de NI et peu de OV, et donc le manque de différence significative. Avec peu d'ajustements possibles, la direction générale du trait est proche de la direction initiale prise par la main. Notre

	Balistique	Précis
CT	-	DIRECT < INDIRECT
OE	DIRECT < INDIRECT	DIRECT < INDIRECT
OV	-	DIRECT < INDIRECT
NI	-	-

TABLE 4.1 – Effet de l'aspect direct de l'interaction sur le temps (CT), l'erreur d'orientation (OE), la variabilité d'orientation (OV), et le nombre de points d'inflexion (NI), pour les styles de dessin balistique (à gauche) et précis (à droite)

expérience apporte ainsi des preuves empiriques qu'il est plus difficile de diriger la main dans une orientation donnée quand il y a dissociation oeil-main, c'est-à-dire en utilisant une technique d'interaction indirecte.

Dans le cas du dessin précis, nous avons trouvé une différence significative, bien que faible en moyenne, concernant OE (0.4 degré) entre DIRECT et INDIRECT. Comme ce style de dessin ne contraint pas le temps d'exécution, les participants ont pris tout le temps nécessaire pour tracer une "belle" ligne. Cependant, pour dessiner une ligne avec une erreur d'orientation (OE) similaire en DIRECT et en INDIRECT, les participants ont besoin de 55% de plus de temps en INDIRECT qu'en DIRECT. De plus, même si nous n'avons pas trouvé de différence significative concernant le nombre de points d'inflexion (NI), DIRECT a conduit à 30% de moins de variabilité d'orientation (OV) que INDIRECT. Une interprétation est que la fréquence d'ajustement est une caractéristique propre au participant, tandis que l'intensité de l'ajustement dépend de la technique d'interaction. En résumé, les participants ont pu effectuer une orientation de trait presque aussi bonne en INDIRECT qu'en DIRECT dans le cadre du dessin précis, mais au prix de performances et de qualité significativement plus faibles. Ceci fournit une autre explication de la plus grande difficulté de tracer des lignes orientées en interaction indirecte qu'en interaction directe.

#### 4.3.2 Effet du gain dans le cadre de techniques indirectes

Nous considérons maintenant l'effet du gain en nous concentrant sur les comparaisons entre techniques indirectes uniquement. Certains participants ont exprimé avoir apprécié davantage HOMOGÈNE par rapport à INDIRECT car ils ont trouvé pratique de parcourir moins de distance sur la tablette que sur l'écran. À l'inverse, d'autres participants ont trouvé que HOMOGÈNE était "trop sensible". HÉTÉROGÈNE a été considéré comme "perturbante" par la majorité des participants, même s'ils ne pouvaient pas en expliquer la raison.

	Balistique	Précis
<b>CT</b>	-	-
<b>OE</b>	-	-
<b>OV</b>	-	INDIRECT < HOMOGÈNE INDIRECT < HÉTÉROGÈNE
<b>NI</b>	INDIRECT < HÉTÉROGÈNE	-

TABLE 4.2 – Effet du gain sur le temps (CT), l'erreur d'orientation (OE), la variabilité d'orientation (OV), et le nombre de points d'inflexion (NI), pour les styles de dessin balistique (à gauche) et précis (à droite).

L'analyse des résultats quantitatifs pour les techniques d'interaction indirectes est résumé dans la table 4.2. Le gain a affecté la variabilité des orientations (OV) dans le cas du dessin précis : INDIRECT amène à 17% de variabilité de moins que HOMOGENE et 14% de variabilité de moins que HÉTÉROGENE. De plus, INDIRECT provoque 25% de moins de points d'inflexion (NI) que HÉTÉROGENE en dessin balistique. Étant donné l'importante distorsion de la condition HÉTÉROGENE (2 :1), notre étude révèle que l'effet d'un gain non homogène est relativement limité.

La confrontation des résultats qualitatifs et quantitatifs nous indique que même si certains participants ont apprécié la réduction du mouvement physique offerte par le gain, l'augmentation de la valeur du gain réduit la qualité de la production en augmentant la variabilité de la direction du trait. C'est un résultat attendu dans la mesure où dans l'interaction HOMOGENE, on peut voir le trait sur l'écran comme étant un zoom du trait de l'espace moteur. Ainsi, le gain amplifie la difficulté d'atteindre une bonne direction, ce qui peut expliquer une plus grande variabilité.

Nous n'avons pas inclus de gains inférieurs à 1 dans notre étude. D'autres expériences seront nécessaires pour déterminer si la réduction du gain peut améliorer la qualité de production, tout en restant à un niveau acceptable de performances.

## **Bilan**

Dans ce chapitre nous avons étudié l'influence de l'aspect direct de l'interaction sur l'utilisation de périphériques à stylet par des utilisateurs novices pour le dessin balistique et précis, en terme de temps de réalisation et de qualité de production. Nous avons montré que l'interaction directe améliore les performances de dessin et conduit à des erreurs d'orientation plus faibles et une moindre variabilité comparée à l'interaction indirecte. De plus, nous avons également montré qu'un gain fort entre déplacement physique et virtuel augmente la variabilité de l'orientation des traits.

Nous avons relevé dans l'état de l'Art que les utilisateurs expriment de la difficulté à utiliser un dispositif d'interaction indirect pour réaliser des tâches de tracé. Notre étude fournit des données objectives qui contribuent à expliquer cette difficulté. Si l'interaction indirecte présente une difficulté pour le tracé de simples lignes droites en 2D, nous pouvons anticiper qu'elle risque de constituer une difficulté importante pour une tâche de peinture numérique 3D dans le cadre d'un usage novice. Cet inconvénient est à mettre en opposition avec les avantages de l'interaction indirecte en terme de confort : on peut supposer que cet aspect, s'il est d'une grande importance dans un cadre professionnel

où les utilisateurs peuvent passer 10h par jour dans la même position, est moins crucial dans le cadre d'un usage récréatif.

Notre étude indique donc que l'usage d'écrans interactifs est plus adapté que les tablettes graphiques pour la peinture numérique auprès d'un public novice. Dans le cadre de la peinture numérique sur modèle 3D, il convient d'étudier et proposer une navigation 3D efficace adaptée aux écrans interactifs. Cette étude fait l'objet du chapitre suivant.



# Chapitre 5

## Interactions surjectives pour les rotations 3D

### Sommaire

---

<i>Introduction</i> . . . . .	90
<b>5.1 Le critère de surjection</b> . . . . .	<b>91</b>
5.1.1 Définition . . . . .	91
5.1.2 Surjection et manipulation directe . . . . .	92
<b>5.2 Nouvelles techniques de rotation 3D</b> . . . . .	<b>93</b>
5.2.1 TAV+ . . . . .	93
5.2.1.1 TAVZ : la technique originale . . . . .	93
5.2.1.2 TAV+ : un TAVZ intégral et surjectif . . . . .	94
5.2.2 Arcball+ . . . . .	96
5.2.2.1 Arcball : la technique originale . . . . .	96
5.2.2.2 Arcball+ : un Arcball intégral et surjectif . . . . .	97
<b>5.3 Analyse de critères</b> . . . . .	<b>98</b>
5.3.1 Ergonomie . . . . .	98
5.3.2 Intégralité . . . . .	99
5.3.3 Surjection . . . . .	99
<b>5.4 Évaluation expérimentale</b> . . . . .	<b>99</b>
5.4.1 Matériel . . . . .	100
5.4.2 Tâches . . . . .	100
5.4.3 Conditions . . . . .	101
5.4.4 Protocole . . . . .	102
5.4.5 Mesures . . . . .	103
<b>5.5 Résultats</b> . . . . .	<b>103</b>
5.5.1 Effet de la technique d'interaction . . . . .	105
5.5.2 Effet de la surjection . . . . .	106

---

5.5.3 Évaluation subjective . . . . .	106
<b>5.6 Discussion . . . . .</b>	<b>107</b>
<b><i>Bilan</i> . . . . .</b>	<b>108</b>

---

## Introduction

Les interactions permettant d'effectuer des rotations 3D sont complexes et difficiles à manipuler par des utilisateurs novices (cf. section 2.1.1.2). Elles constituent donc un frein à la navigation dans le cadre d'une tâche de peinture 3D.

Nous avons trouvé dans l'état de l'Art différents critères qui permettent d'analyser à priori les qualités d'une technique d'interaction. Nous nous appuyons en particulier sur le travail de Bade et al. qui proposent quatre critères d'ergonomie que nous avons présentés en section 2.1.1.2. D'après les auteurs, ces critères sont primordiaux pour que l'interaction de rotation 3D soit intuitive et facile à utiliser [146].

Nous appuyons également notre analyse sur le critère d'intégralité de l'interaction proposé par Jacob et al. [151] (cf. section 2.1.3). Les auteurs indiquent que la structure du dispositif d'interaction doit correspondre à la structure de la tâche pour que l'interaction soit ergonomique. Or Masliah et al. montrent que la structure de la tâche de rotation 3D est intégrale suivant les trois degrés de liberté [154]. Il est donc souhaitable que la structure de l'interaction permette de manipuler les trois degrés de liberté de manière intégrale.

Dans ce chapitre, nous introduisons un nouveau critère que nous appelons *surjection*. Nous définissons une interaction de rotation 3D comme *surjective* si elle offre la possibilité d'atteindre n'importe quel état final de l'objet manipulé depuis n'importe quel état courant de celui-ci au cours de l'interaction. Nous expliquons en quoi la surjection est un critère clef pour permettre des transitions fluides entre les différentes phases de l'interaction.

Ensuite, nous présentons deux nouvelles techniques d'interaction pour les rotations 3D qui sont à la fois *surjectives* et *intégrales*. Nous analysons que ces deux techniques respectent un plus grand nombre de critères ergonomiques que celles de l'état de l'art.

Enfin, nous effectuons une étude expérimentale afin de comparer les performances de ces deux nouvelles techniques avec celles des techniques de l'état de l'art. Nous montrons que les nouvelles techniques sont plus performantes, et sont préférées par les participants.

## 5.1 Le critère de surjection

### 5.1.1 Définition

Nous avons vu, en section 2.1.1.2, deux manières de séparer le contrôle des 3 degrés de liberté des rotations 3D. Certains techniques d'interaction utilisent différents dispositifs d'entrée, par exemple en dédiant la molette de la souris pour le contrôle de la rotation selon l'axe perpendiculaire à l'écran [147]. D'autres techniques d'interaction utilisent une répartition spatiale des modes pour définir le degré de liberté qui est contrôlé, c'est le cas par exemple avec l'utilisation de la position du contact initial dans la technique Arcball [144]. Ceci a pour conséquence de rendre certaines orientations de l'objet manipulé hors d'atteinte à l'aide d'un unique *geste* sur le dispositif d'entrée.

Dans le cas de l'Arcball, l'*axe de contrôle* est défini comme l'axe passant par le centre de l'Arcball et le point de contact initial (voir figure 5.4). Une fois que la souris ou le doigt contrôle l'Arcball (et donc que l'axe de contrôle est défini), il est impossible d'effectuer une rotation autour de l'axe de contrôle. Lorsqu'un utilisateur souhaite appliquer des rotations simples à l'objet, il fait généralement en sorte de bien choisir le point de contact initial afin que son orientation cible soit accessible. Cependant, pour des rotations plus complexe, le choix d'un point de contact initial approprié peut être difficile, voire impossible. En effet, si l'on considère le cas où l'utilisateur veut observer le coté caché d'un objet, il ne lui est pas possible de connaître l'orientation de ce coté, et donc de choisir le point de contact initial. Par conséquent, l'orientation souhaitée est inconnue tant que l'interaction n'a pas commencé et que la face souhaitée n'est pas visible. L'utilisateur ne peut pas voir quelle rotation dans le plan de l'écran est nécessaire pour atteindre cette orientation (figure 5.1). Il en résulte que l'interaction à l'aide de l'Arcball est fréquemment décomposée en une séquence de gestes atomiques qui sont interrompus lorsque l'utilisateur doit définir un nouvel axe de contrôle, ce qui impose au doigt ou à la souris d'interrompre puis de reprendre l'interaction.

Nous généralisons ce problème, identifié ici sur la technique d'interaction Arcball, en définissant le critère de *surjectivité* d'une interaction : une technique *surjective* permet d'atteindre tout état final de l'objet contrôlé depuis tout état courant de celui-ci, durant une seule interaction. Comme expliqué ci-dessus, la technique Arcball *n'est pas* surjective.

Ce critère se nomme *surjectivité* car il s'inspire du concept mathématique correspondant : une surjection est une fonction  $f$  définie sur un ensemble  $A$  et prenant ses valeurs dans un ensemble  $B$  de manière à ce que pour tout  $b \in B$  il existe un  $a \in A$  pour lequel  $f(a) = b$ . Dans le cas d'une technique d'interaction,  $B$  correspond à l'ensemble

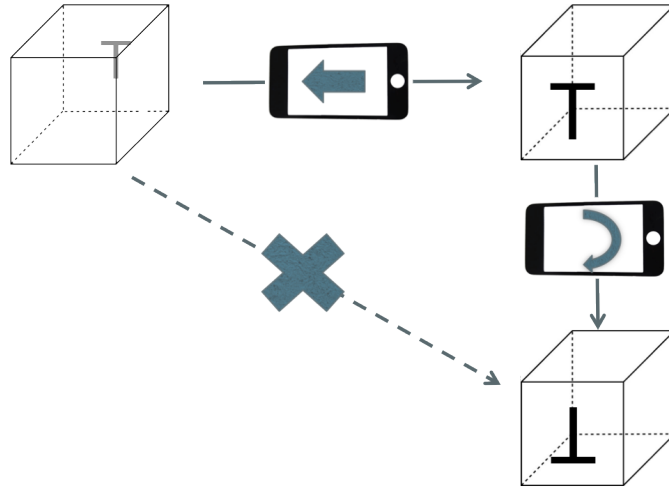


FIGURE 5.1 – Exemple d’interaction non surjective : si l’utilisateur souhaite afficher le T à l’envers, il doit d’abord tourner l’objet de manière à faire apparaître le T, puis retourner celui-ci.

de tous les *états* possibles que l’objet contrôlé peut prendre. L’ensemble  $A$  correspond à l’ensemble des séquences d’évènements possibles d’un *geste unique* de l’interaction. Ce type de séquence contient généralement uniquement des évènements de type *motion*, et aucun évènement de type *mouse-button-up*, *mouse-button-down*, *finger-touch*, ou *finger-untouch*. La fonction  $f_{b_c}$  correspond à la *fonction de transfert* de la technique d’interaction, qui définit l’état final  $b_f$  de l’objet contrôlé en fonction de l’état courant  $b_c$  et des entrées  $a$  de l’utilisateur.

Une technique d’interaction satisfait le critère de surjection si pour tout état courant de l’objet contrôlé  $b_c$  et pour tout état final  $b_f$ , il existe une séquence d’évènements de type *motion*  $a$  qui amène l’objet de  $b_c$  à  $b_f$ . Cela se traduit sous forme mathématique par l’expression :

$$\forall b_c, \forall b_f, \exists a : f_{b_c}(a) = b_f \quad (5.1)$$

### 5.1.2 Surjection et manipulation directe

Une technique d’interaction surjective permet à l’utilisateur d’ajuster son geste de manière *continue* lorsqu’il cherche à atteindre un objectif. Par conséquent, la surjection peut être vue comme un critère clef pour réaliser le troisième principe du paradigme de “Manipulation directe”. Ce principe déclare que l’interaction doit être “rapide, incrémentale, et réversible” [209]. Lorsque l’on compare les interactions *non* surjectives aux interactions surjectives, on constate que les premières sont moins réversibles dans la mesure où une erreur initiale ne peut pas être corrigée par des ajustements ultérieurs. Elles sont

également moins incrémentales au niveau gestuel, puisque certaines rotations nécessitent de combiner plusieurs gestes successifs : on ne peut pas atteindre l'orientation cible grâce à des ajustements additionnels du geste de départ. Enfin, les interactions non surjectives sont moins rapides dans la mesure où stopper un geste pour en commencer un autre est généralement plus coûteux qu'ajuster continûment un unique geste.

Comme les principes de manipulation directe restent un ensemble de recommandations fondamentales pour la création d'interactions facile à apprendre et efficaces, nous attachons une importance particulière à concevoir des techniques d'interactions pour rotations 3D qui soient surjectives.

## 5.2 Nouvelles techniques de rotation 3D

Nous avons développé deux nouvelles techniques d'interaction pour les rotations 3D dans le contexte de l'interaction tactile multipoint à une main, de manière à pouvoir les combiner avec une interaction de peinture numérique au stylet. Nous avons étendu les techniques *Two Axis Valuator + Z* (TAVZ) de Scheurich et al. [141] et *Arcball* de Shoemake [144] présentées en section 2.1.1.2, qui sont les techniques de rotations 3D les plus répandues.

Les techniques étendues s'intitulent respectivement *TAV+* et *Arcball +*. Elle nécessitent chacune l'usage de deux doigts d'une seule main, permettent un contrôle intégral des trois degrés de liberté et sont surjectives.

### 5.2.1 TAV+

#### 5.2.1.1 TAVZ : la technique originale

L'interaction TAVZ de Sheurich utilise deux modes de rotations qui sont déclenchés par le nombre de doigts en contact avec le dispositif tactile multipoint [141].

Lorsqu'un seul doigt est en contact avec le dispositif, celui-ci contrôle la rotation autour d'un axe défini dans le plan de l'écran et perpendiculaire au mouvement (voir Figure 5.2a). Pour une meilleure compréhension, nous considérons ici le cas des mouvements élémentaires rectilignes : en effet un mouvement curviligne est une suite d'événements reliés entre eux de manière rectiligne. Le comportement global de l'interaction consiste donc en l'application successive des différentes rotations pour chaque segment élémentaire. L'amplitude de la rotation est proportionnelle à la distance parcourue sur l'écran durant le mouvement. Dans la version originale de l'interaction TAVZ, cette amplitude est égale

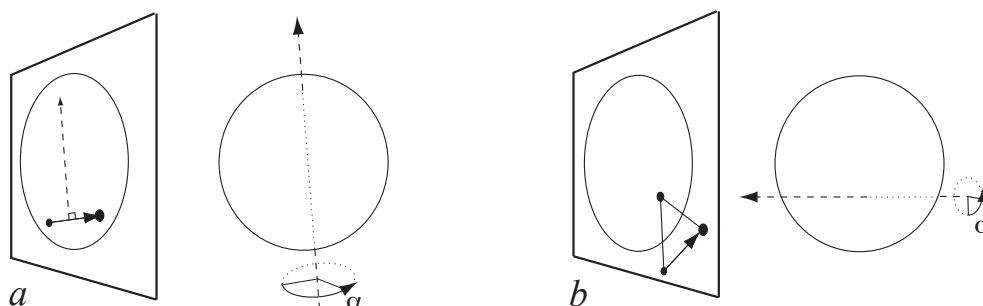


FIGURE 5.2 – Interaction TAVZ. *a* : avec 1 doigt, l'axe de rotation est défini dans le plan de l'écran, et perpendiculaire au mouvement élémentaire du doigt. *b* : avec 2 doigts, l'axe de rotation est perpendiculaire à l'écran, et passe par le point d'ancrage défini par le premier doigt.

à 180 degrés lorsque le doigt parcourt entièrement l'écran de gauche à droite. Cependant, ce gain est paramétrable.

Le contact de deux doigts avec le dispositif permet de contrôler la rotation selon l'axe perpendiculaire à l'écran. Le premier doigt définit le *point de pivot* et reste fixe, tandis que le second doigt indique l'amplitude de la rotation (voir figure 5.2b). Le gain de cette amplitude est égal à 1, mais il est également paramétrable.

Les deux modes introduisent une divergence au niveau de la gestion du *point de pivot* de la rotation : le mode à un doigt applique la rotation à un point de pivot arbitraire défini préalablement par l'utilisateur, tandis que le mode à deux doigts utilise un point de pivot temporaire défini pendant l'interaction. Lorsque les deux points de pivot ne sont pas confondus, le mode à deux doigt provoque une combinaison de rotations et de translations de l'objet. Dans certaines situations, cela déplace l'objet en dehors de l'écran, et il est alors difficile de pouvoir ramener celui-ci. Autoriser la définition du point de pivot ajoute de la complexité à l'interaction, et le bénéfice n'est pas clair dans le cas d'une tâche d'observation simple pour des utilisateurs novices.

### 5.2.1.2 TAV+ : un TAVZ intégral et surjectif

Nous avons conçu une extension de la technique TAVZ qui satisfait les contraintes suivantes :

- l'interaction doit appliquer la rotation à un point de pivot fixe, quel que soit l'axe de rotation
- l'interaction doit être surjective
- l'interaction doit permettre le contrôle intégral des 3 degrés de liberté.

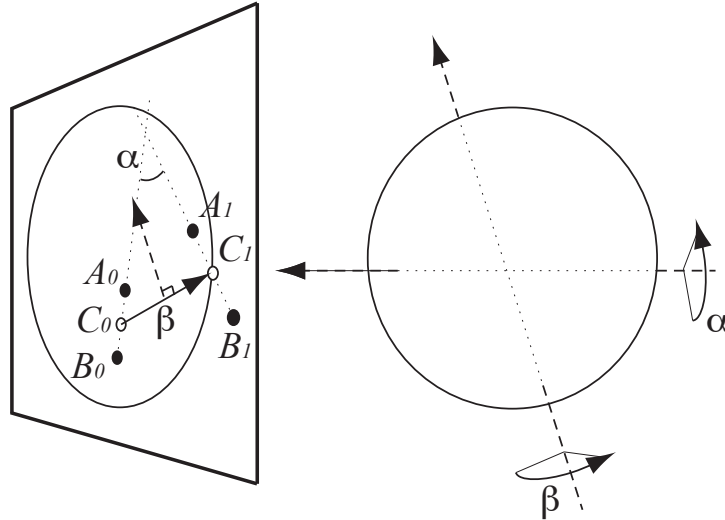


FIGURE 5.3 – Interaction TAV+ : Les rotations en X et en Y sont contrôlées par la position du milieu des deux points de contact ( $C_i$ ). La rotation en Z est contrôlée via la rotation du vecteur défini par les points de contact.

Nous avons appelé cette interaction TAV+. Elle prend en entrée deux points de contacts, et nécessite par conséquent l'usage permanent et simultané de deux doigts. De même que pour l'interaction TAVZ, les rotations selon les axes dans le plan de l'écran sont contrôlées par la position d'un seul point de contrôle : il s'agit du barycentre des deux points de contact. La rotation selon l'axe perpendiculaire à l'écran est calculée à partir de la rotation du vecteur défini par les deux points de contact.

Ceci est illustré à la figure 5.3 : soient  $A_0, B_0$  les positions initiales de chaque doigt au début du mouvement et  $A_1, B_1$  leurs positions finales. Soient  $C_0, C_1$  les milieux respectifs de  $[A_0, B_0]$  et  $[A_1, B_1]$ . Alors la nouvelle orientation de l'objet est définie par la combinaison de deux rotations :

1. une rotation autour de l'axe perpendiculaire à l'écran et passant par le centre de l'objet. L'amplitude de la rotation est définie par l'angle entre  $[A_0, B_0]$  et  $[A_1, B_1]$ .
2. une autre rotation autour de l'axe perpendiculaire à  $[C_0, C_1]$ , parallèle à l'écran, et passant par le centre de l'objet. L'amplitude de la rotation est définie comme dans l'implémentation de Scheurich et al : elle vaut 180 degrés lorsque le mouvement traverse entièrement l'écran de gauche à droite.

Nous fournissons dans la section 5.3 une analyse comparative des critères d'ergonomie satisfaits par l'interaction TAV+.

Comme cette interaction est conçue dans le but d'être utilisée conjointement à une interaction de peinture numérique sur un objet, nous avons choisi arbitrairement comme point

de pivot le centre de l'objet en question. Cependant, ce point de pivot est paramétrable. Le choix du point de pivot est discuté de manière plus approfondie en section 6.1.2.4.

## 5.2.2 Arcball+

### 5.2.2.1 Arcball : la technique originale

L'Arcball de Shoemake est basé sur une *sphère invisible* qui englobe l'objet contrôlé [144]. L'orientation de l'objet est liée à celle de la sphère, ce qui permet aux utilisateurs de manipuler uniquement celle-ci. L'Arcball est conçu pour un dispositif d'entrée à deux degrés de libertés (en l'occurrence, il s'agit de la souris), ce qui implique l'existence de modes pour pouvoir manipuler les trois degrés de libertés. L'interaction comprend deux modes qui sont déclenchés par une activation spatiale : un mode correspond à l'interaction *sur* la sphère, et l'autre mode correspond à l'interaction *en dehors* de la sphère.

La position de la souris ou du doigt sur l'écran est projetée sur la sphère. Soient  $P_0$  et  $P_1$  les projections des positions de début et de fin du dispositif d'entrée. Soit  $O$  la position du centre de la sphère. On définit alors l'axe de rotation comme le produit vectoriel de  $\overrightarrow{OP_0}$  et  $\overrightarrow{OP_1}$ . Cet axe est perpendiculaire au plan défini par  $\overrightarrow{OP_0}$  et  $\overrightarrow{OP_1}$ , et passe par  $O$ .

Quand la projection reste sur la sphère, l'axe de rotation est une combinaison de l'axe perpendiculaire à l'écran et d'un axe dans le plan de l'écran, comme on peut le voir à la figure 5.4a. Ainsi, l'Arcball permet le contrôle intégral des trois degrés de liberté.

Si la projection n'intersecte pas la sphère, on projète alors à la place la position du point de contact sur le plan parallèle à l'écran et passant par le centre de la sphère, comme on peut le voir à la figure 5.4b. L'axe de rotation est alors perpendiculaire à l'écran.

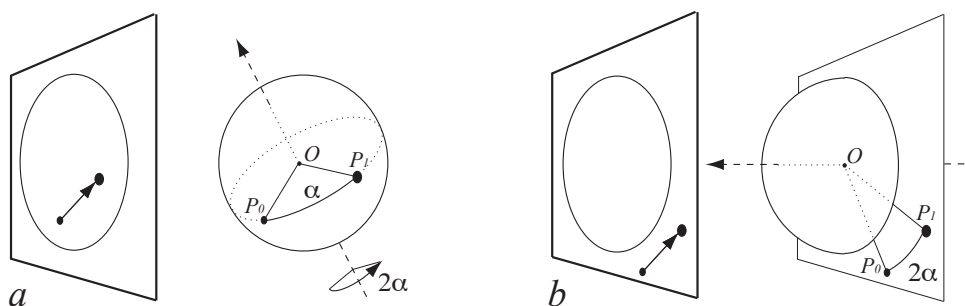


FIGURE 5.4 – Interaction Arcball : *a* : L'interaction à 1 doigt sur la sphère permet le contrôle intégral des 3 axes de rotation. *b* : En dehors de la sphère, le doigt contrôle une rotation autour de la normale de l'écran.



Le gain de l'Arcball est non paramétrable si l'on utilise l'implémentation proposée par Shoemake : en effet, cette implémentation utilise des quaternions pour le calcul des amplitudes, ce qui impose que l'amplitude de la rotation soit le double de l'angle entre  $\overrightarrow{O, P_0}$  et  $\overrightarrow{O, P_1}$ .

Le point de pivot est également non paramétrable, car il correspond au centre de la sphère.

À cause des deux différents modes (sur et en dehors de la sphère), la rotation est partiellement contrôlée de manière intégrale : elle est intégrale uniquement lorsque la projection reste sur la sphère. Une autre limitation de l'Arcball est qu'elle ne satisfait pas le critère de surjection, comme on a pu voir précédemment à la section 5.1.

### 5.2.2.2 Arcball+ : un Arcball intégral et surjectif

Nous avons conçu une version de l'Arcball entièrement intégrale et surjective, qui permet de paramétrer l'amplitude de la rotation autour de l'axe de contrôle. Nous appelons cette interaction *Arcball+*. Elle prend en entrée deux points de contacts, et nécessite par là l'usage permanent et simultané de deux doigts.

Soit  $C$  le milieu des deux points de contacts. Soit  $pC$  la projection de  $C$  sur la sphère. Alors  $pC$  est utilisé pour manipuler la sphère de la même manière que dans la technique originale Arcball.

De même que pour TAV+, la rotation des points de contacts fournit un nouveau degré de liberté qui est utilisé pour contrôler la rotation autour de l'Axe de Contrôle ( $O, pC$ ),

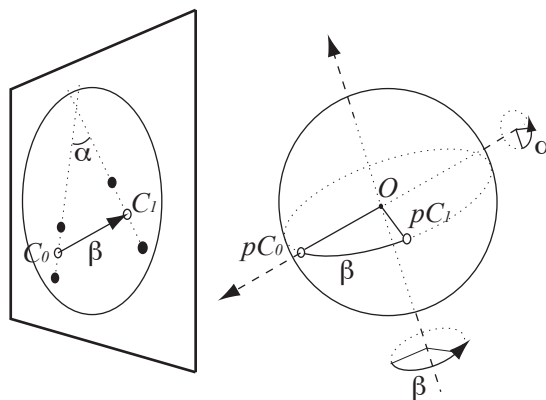


FIGURE 5.5 – Interaction Arcball+ : La projection du milieu des points de contact  $pC$  contrôle la sphère comme dans la technique Arcball originale. La rotation du vecteur défini par les deux points de contact contrôle la rotation autour de l'axe défini par  $O$  et  $pC_0$ .

comme on peut le voir à la figure 5.5. C'est la principale différence avec la technique originale : cela permet à Arcball+ d'être une technique d'interaction surjective.

Dans Arcball+, les amplitudes de rotation sont calculées à partir des mouvements des projections, mais sont paramétrables contrairement à l'Arcball, car l'implémentation est différente de celle proposée par Shoemake.

## 5.3 Analyse de critères

Nous analysons dans cette partie comment chacune des techniques décrites précédemment satisfait les critères d'ergonomie de Bade et al. [146], d'intégralité et de surjection. Les résultats de cette analyse sont synthétisés dans le tableau 5.1.

### 5.3.1 Ergonomie

Les critères d'ergonomie proposés par Bade et al. et présentés en section 2.1.1.2 sont les suivants :

- B1** Des actions similaires doivent provoquer des réactions similaires
- B2** La direction d'une rotation doit correspondre à la direction du mouvement du périphérique d'entrée
- B3** Les rotations 3D doivent être transitives.
- B4** Le gain contrôle/affichage doit être personnalisable.

Si l'on considère ces critères d'ergonomie, on constate que :

- B1 : TAV+ est la seule technique qui respecte le premier critère, car c'est la seule qui fournit le même résultat quelle que soit la position de départ sur l'écran de l'interaction. Avec les autres techniques, un même geste effectué depuis différentes positions de départ sur l'écran provoque des résultats différents.
- B2 : Toutes les techniques font correspondre la direction de pointage et la direction de la rotation.
- B3 : Les techniques TAVZ et TAV+ ne respectent pas le principe de transitivité. Ceci est du au principe de base de TAV, selon lequel les rotations sont toujours appliquées à un cadre global qui ne se déplace pas avec l'objet. Si l'on tourne un objet de  $+90^\circ x$ ,  $+90^\circ y$ ,  $-90^\circ x$ , puis de  $-90^\circ y$ , cela ne le ramène pas à l'orientation de départ, mais à une orientation de  $120^\circ$ .
- B4 : L'Arcball est la seule technique qui impose un gain contrôle/affichage. Cette contrainte a été supprimée dans la technique Arcball+, car le gain est paramétrable.

### 5.3.2 Intégralité

On note **I** le critère d'intégralité. On constate que l'Arcball respecte partiellement ce critère : le contrôle simultané des trois degrés de liberté est possible, mais uniquement quand le premier point de contact se projète sur la sphère. Ce critère est préservé dans Arcball+, mais est également étendu à toute la surface d'interaction : Arcball+ est donc complètement intégrale.

L'interaction TAVZ comprend deux modes, ce qui exclue le contrôle simultané des trois degrés de liberté. Ceci est corrigé dans TAV+, qui est donc une technique intégrale.

### 5.3.3 Surjection

On note **S** le critère de surjection. On constate que l'Arcball n'est pas surjective, car elle ne permet pas d'atteindre toutes les orientations depuis un point de contact quelconque : les seules orientations accessibles sont celles définies par les axes qui appartiennent au plan normal à l'axe de contrôle, donc les rotations autour de l'axe de contrôle sont impossible (voir la section 5.1).

La technique Arcball+ ajoute un nouveau degré de liberté en entrée à l'Arcball. Ce degré de liberté permet d'effectuer des rotations autour de l'axe de contrôle. Ainsi, chaque axe de rotation souhaité peut être décomposé en un axe accessible par l'Arcball et le nouvel axe de contrôle. À tout instant de l'interaction, il est donc possible de trouver une trajectoire pour atteindre n'importe quelle nouvelle orientation. Ainsi, la technique Arcball+ est surjective.

Comme la technique TAVZ comprend plusieurs modes nécessitant un nombre de contacts différent ainsi que des initialisations différentes, elle n'est pas surjective : les utilisateurs doivent en effet changer de mode, et donc décomposer leurs rotations pour atteindre leur objectif.

La technique TAV+ permet de manipuler les 3 axes de rotations en même temps, mais aussi de manipuler chacun d'entre eux séparément. Il est donc possible d'atteindre toute nouvelle orientation depuis l'orientation courante : TAV+ est surjective.

## 5.4 Évaluation expérimentale

Nous allons dans cette partie présenter une étude expérimentale dont l'objectif est d'évaluer l'efficacité des techniques d'interaction TAV+ et Arcball+ sur une tâche de

	<b>Arcball</b>	<b>Arcball+</b>	<b>TAVZ</b>	<b>TAV+</b>
<b>B1</b>	-	-	-	+
<b>B2</b>	+	+	+	+
<b>B3</b>	+	+	-	-
<b>B4</b>	-	+	+	+
<b>I</b>	-/+	+	-	+
<b>S</b>	-	+	-	+

TABLE 5.1 – Respect des critères d’ergonomie (**Bi**), d’intégralité (**I**) et de surjection (**S**) pour les techniques de rotation originales et améliorées.

docking en terme de temps de réalisation, de précision, et d’intégration du contrôle utilisateur. Nous allons également considérer l’effet du critère de surjection sur ces mêmes variables.

#### 5.4.1 Matériel

L’expérience a été effectuée sur un ordinateur portable comprenant un processeur Intel Core i7, 2,6Ghz, ainsi qu’une carte graphique NVidia GT 650M. L’expérience repose sur une application programmée en C++ à l’aide du framework Open Scene Graph. Nous avons utilisé un écran interactif de 24 pouces à sa résolution native de 1920x1200 pixels. Comme une configuration d’affichage verticale génère de la fatigue au niveau des bras dans le cadre de l’interaction tactile, nous avons réglé l’écran de manière à lui donner une inclinaison de 30° par rapport à l’horizontale.

#### 5.4.2 Tâches

Les participants ont interagi avec l’écran à l’aide d’un doigt ou bien de deux doigts. Ils ont également contrôlé les différentes étapes de l’expérimentation en pressant la barre espace et la touche entrée sur un clavier externe.

Les participants ont effectué une tâche de *docking* en rotation dans l’espace, donc à 3 degrés de liberté. Un cube ayant une orientation initiale aléatoire est présenté au participant. La taille du cube est d’environ 75% de la hauteur de l’écran. Le cube comprend sur l’une de ses faces un creux carré. Les participants doivent tourner le cube de manière à ce que le creux apparaisse à l’écran. Comme on peut le voir à la figure 5.6, nous avons utilisé les cotés du creux ainsi qu’un niveau pour fournir un retour visuel intuitif concernant la tolérance selon les trois axes de rotation. Le cube est dans l’orientation demandée avec la tolérance requise lorsque tous les cotés du creux sont visibles, et la ligne rouge du niveau est à l’intérieur de la zone blanche. La forme du creux ainsi que celle de la zone blanche ont été adaptées pour les différents niveaux de tolérance. Les participants

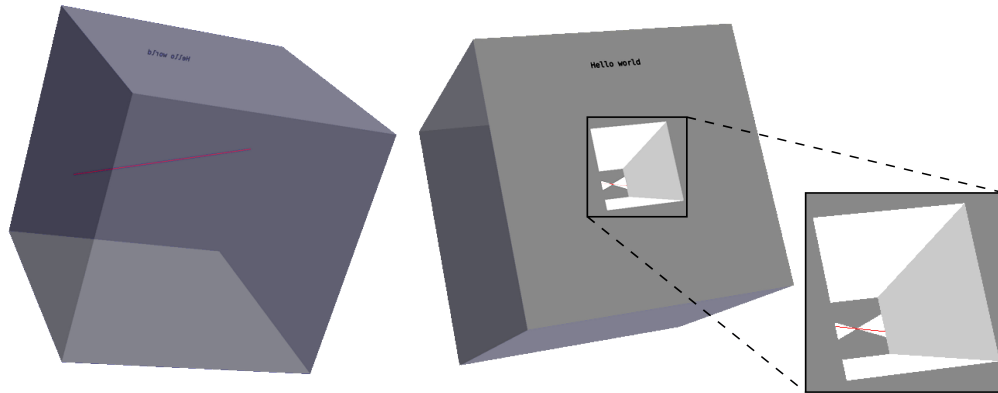


FIGURE 5.6 – Le Cube. À gauche : l'orientation initiale est aléatoire. La cible est sur la face arrière, et le texte est visible à l'envers grâce à la transparence. Au milieu : le creux fait face à l'écran, et le cube est dans une orientation incluse dans les  $20^\circ$  de tolérance pour l'axe Y (les cotés haut et bas du creux sont visibles) et pour l'axe Z (la ligne rouge est dans la zone blanche), mais pas pour l'axe X (le coté gauche du creux n'est pas visible). À droite : Gros plan sur le creux.

valident leurs rotations en pressant la barre espace sur le clavier. Une seconde pression sur la barre espace relance une nouvelle orientation initiale.

Nous avons considéré deux niveaux de tolérance dans cette expérience :  $2^\circ$  et  $20^\circ$ . Le niveau de tolérance à  $2^\circ$  a permis de modéliser une tâche de *haute précision* comme l'alignement précis d'un objet virtuel. Cependant, la principale motivation de ces travaux est d'étudier une tâche d'exploration, c'est pourquoi nous avons inclus une tâche de *faible précision* (niveau de tolérance à  $20^\circ$ ) dans l'expérience. Celle-ci comprend donc deux tâches distinctes.

### 5.4.3 Conditions

On compare les les deux nouvelles techniques, TAV+ et Arcball+, avec les techniques originales Arcball et TAVZ. Cependant, afin de rester dans le cadre d'une tâche d'observation et de mieux respecter les critères de Bade et al., Arcball et TAVZ ont été légèrement adaptées :

1. Nous avons adapté TAVZ en appliquant les rotations au centre de l'objet. Ceci évite les combinaisons de rotations et translations provoquées par le mode à deux doigts (voir la section 5.2.1.1). Nous appelons cette adaptation TAV<sup>2</sup>.
2. Nous avons ajouté un mode à deux doigts à l'Arcball. Avec un seul doigt, cette adaptation est équivalente à l'Arcball, mais quand un second point de contact apparaît, l'utilisateur peut contrôler la rotation autour de l'axe de contrôle, qui est constitué du centre de la sphère et de la projection du premier point de contact. Nous avons appelé cette adaptation Arcball<sup>2</sup>.

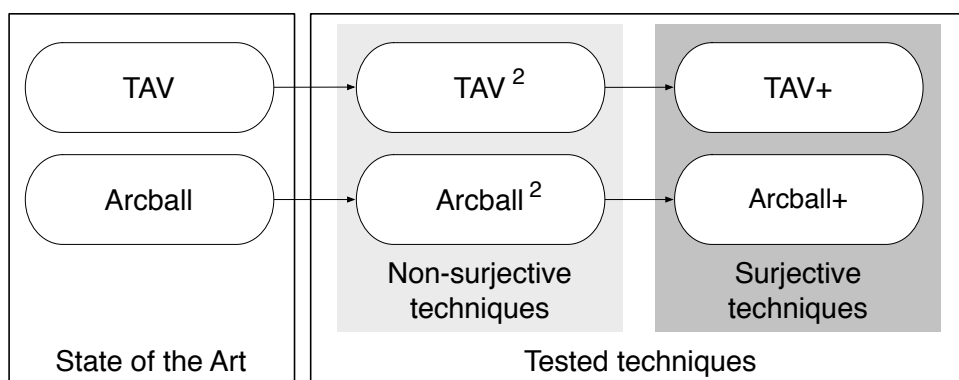


FIGURE 5.7 – Évolutions depuis les techniques Arcball et TAVZ. Dans le but de rester dans une tâche d’observation, les techniques Arcball et TAVZ ont été légèrement adaptées avant d’être comparées aux techniques Arcball+ et TAV+.

La figure 5.7 résume les différentes techniques que nous avons décrites dans ce chapitre, ainsi que les liens entre chacune d’entre elles.

#### 5.4.4 Protocole

Nous avons recruté 16 participants, âgés de 23 à 57 ans, dont 7 femmes et 9 hommes ; 2 sont gauchers, 13 droitiers, et l’un d’eux est ambidextre ; 7 sont des étudiants de l’université, ce qui n’est pas le cas des 9 autres participants. 14 participants ont l’habitude d’utiliser des dispositifs tactiles multipoints comme les Smartphones de manière quotidienne. 2 participants ont déjà eu l’occasion de manipuler une scène 3D, mais les 14 autres participants n’ont jamais manipulé de contenu 3D.

L’expérience a eu lieu dans une pièce silencieuse. Le participant est assis devant l’écran interactif, incliné à 30° de l’horizontale. Au début de chaque session, le chercheur explique l’ensemble de la procédure au participant et effectue une démonstration des techniques d’interaction. Le chercheur reste à côté du participant durant toute l’expérience.

Nous avons utilisé un design expérimental de type intra-sujet : chaque participant effectue les rotations à l’aide des quatre techniques d’interaction : TAV<sup>2</sup>, Arcball<sup>2</sup>, TAV+ et Arcball+. Chaque technique est testée pour les tâches de haute et faible précision, avec 20 répétitions chacune. Nous avons pu enregistrer au total :

16 (participants)  
 x 4 (techniques d’interaction)  
 x 2 (niveaux de précision)  
 x 20 (répétitions)  
 = 2560 essais.

Chaque nouvelle combinaison de technique et de niveau de précision commence par une phase d'entraînement durant laquelle le participant effectue un nombre illimité d'essais, jusqu'à se sentir à l'aise. Il peut alors presser la touche entrée du clavier pour commencer le test.

Afin de contrebalancer les effets d'apprentissage, l'ordre de présentation des niveaux de précision a été équilibré entre 2 groupes de 8 participants. À l'intérieur de chaque groupe, nous avons également équilibré l'ordre de présentation des techniques.

Les participants ont ensuite noté les techniques selon leur préférences durant un entretien post-expérimental. On a demandé aux participants de prendre en compte la vitesse, la précision, et leur préférence générale dans la notation (voir section 5.5.3).

#### 5.4.5 Mesures

Nous avons mesuré le temps de réalisation ainsi que la précision. Le temps de réalisation est défini entre le premier contact et le dernier contact avant la validation. La précision est l'angle de la rotation minimale nécessaire pour tourner le cube de la position finale produite par le participant à la position cible parfaitement alignée.

Nous avons également calculé une mesure de l'intégralité du contrôle de l'interaction. Nous avons utilisé la mesure *Magnitude of Degrees of freedom Separation* (MDS) définie par Veit et al. [210]. Dans cette mesure, une valeur de 0 indique que tous les degrés de liberté sont manipulés durant une fenêtre de temps donnée, tandis que une valeur de 1 indique qu'un seul degré de liberté y est manipulé.

### 5.5 Résultats

Les résultats de l'expérience sont présentés graphiquement à sur la figure 5.8, et les valeurs correspondantes sont listées dans le tableau 5.2. Nous avons rapporté l'effet de l'interaction individuellement pour chaque technique d'interaction, mais nous avons également rapporté l'effet du critère de surjection :  $TAV^2$  et  $Arcball^2$  constituent le groupe *non surjectif*, tandis que  $TAV+$  et  $Arcball+$  constituent le groupe *surjectif*. Les valeurs rapportées pour chaque groupe sont calculées comme la moyenne des mesures des techniques du groupe.

Les valeurs de la mesure MDS sont présentées à la figure 5.9. Cependant, la découverte tardive d'une erreur dans le code d'enregistrement des trajectoire nous a empêché de calculer la MDS pour tous les participants : les mesures sont uniquement calculées pour les 9 derniers participants.

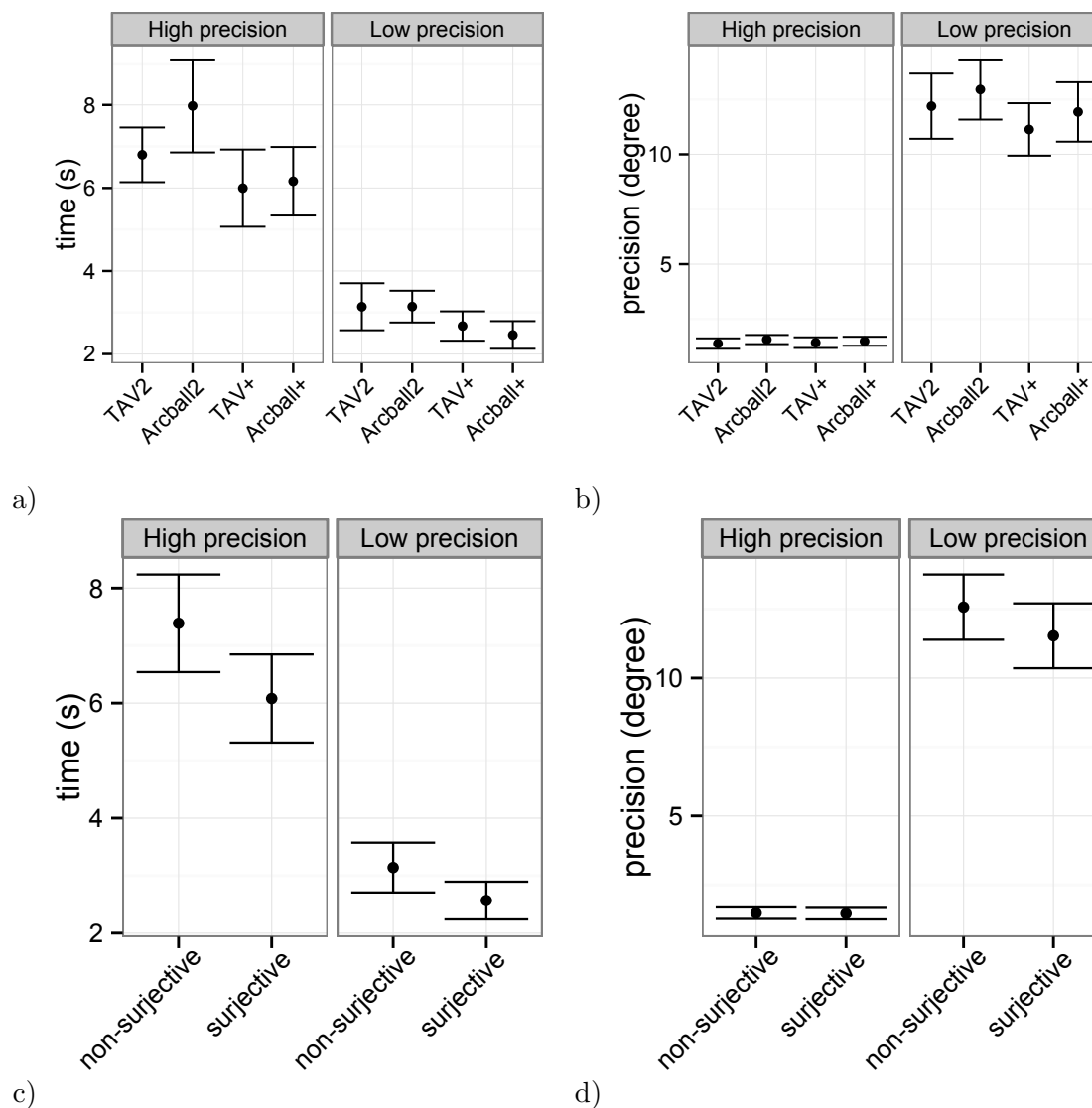


FIGURE 5.8 – Résultats. *a) et b)* : moyenne sur tous les participants du temps de réalisation de la tâche et de la précision, en fonction de la technique d'interaction, avec les intervalles de confiance à 95%. *c) et d)* : les mêmes mesures, agrégées par la moyenne selon les groupes de surjection. La tâche de haute précision correspond à une tolérance de 2° et celle de faible précision à une tolérance de 20°.

Précision demandée	Temps de réalisation (s.)		Précision réalisée (°)	
	haute (2°)	faible (20°)	haute (2°)	faible(20°)
<b>Technique d'interaction</b>				
TAV <sup>2</sup>	6,80	3,14	1,38	12,19
Arcball <sup>2</sup>	7,98	3,14	1,56	12,95
TAV+	6,00	2,67	1,42	11,13
Arcball+	6,16	2,46	1,48	11,93
<b>Surjection</b>				
non surjective	7,39	3,14	1,47	12,57
surjective	6,08	2,57	1,45	11,53

TABLE 5.2 – Résultats



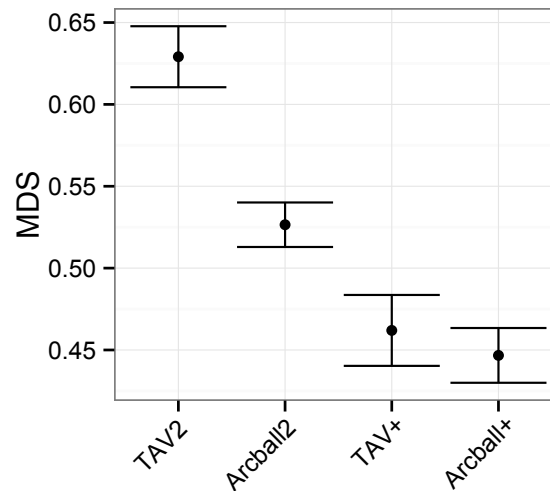


FIGURE 5.9 – Moyenne sur 9 participants de la mesure MDS de l'intégralité du contrôle (0 représente un contrôle entièrement intégral)

### 5.5.1 Effet de la technique d'interaction

Une ANOVA par mesures répétées à un facteur contrôlé révèle que la technique d'interaction a un effet sur le temps de réalisation, aussi bien pour la tâche de haute précision ( $F(3, 45) = 9.84, p < 10^{-4}$ ) que pour la tâche de faible précision ( $F(3, 45) = 8.47, p < 10^{-3}$ ).

Des comparaisons deux-à-deux à l'aide de t-test appariés bilatéraux avec correction de Bonferroni révèlent une différence de temps significative pour la tâche de haute précision entre Arcball<sup>2</sup> et TAV<sup>2</sup> ( $p < 0.05$ ) et entre Arcball<sup>2</sup> et TAV+ / Arcball+ ( $p < 0.01$ ). La technique Arcball<sup>2</sup> est la moins efficace.

Pour la tâche de faible précision, les seules différences de temps significatives sont entre TAV<sup>2</sup> et Arcball+ ( $p < 0.05$ ), entre Arcball<sup>2</sup> et TAV+ ( $p < 0.05$ ), et entre Arcball<sup>2</sup> et Arcball+ ( $p < 0.01$ ).

Une ANOVA par mesures répétées à un facteur contrôlé ne révèle pas d'effet de la technique d'interaction sur la précision, ni dans la tâche de haute précision, ni dans celle de faible précision. Ceci est illustré par les distributions similaires visibles à la figure 5.8b.

Une ANOVA par mesure répétées à un facteur contrôlé ne révèle pas d'effet de la technique d'interaction sur la mesure MDS de contrôle intégral. Cependant, comme indiqué plus haut, l'étude statistique de la mesure MDS porte uniquement sur 9 sujets au lieu des 16 participants. Par ailleurs, la probabilité de l'hypothèse nulle est

faible ( $F(3, 24) = 2.40, p = 0.0929$ ). Il est probable que l'effet soit avéré avec davantage de mesures.

Des comparaisons deux à deux à l'aide de t-test appariés bilatéraux avec correction de Bonferroni confirment ce qui est visible à la figure 5.9 : il y a une différence significative entre chaque paire de techniques d'interaction ( $p < 0.01$ ), sauf entre les techniques surjectives. Les participants ont exécuté avec les techniques surjectives 14% de plus de contrôle intégral par rapport à l'Arcball<sup>2</sup>, et 28% de plus par rapport à TAV<sup>2</sup>.

### 5.5.2 Effet de la surjection

Un t-test apparié bilatéral révèle que pour les tâches de haute et basse précision, la capacité d'intégration de la technique d'interaction a un effet sur le temps de réalisation de la tâche ( $t(15) > 4.4, p < 10^{-3}$ ). Pour les deux précisions, les techniques d'interactions surjectives sont environ 18% plus efficaces que les techniques non surjectives.

Un t-test apparié bilatéral ne révèle aucun effet significatif de la capacité d'intégration de l'interaction sur la précision, que ce soit pour la tâche de haute ou faible précision.

### 5.5.3 Évaluation subjective

Les participants ont donné leurs préférences sous la forme d'un score dans l'ensemble  $\{1, 2, 3, 4\}$  pour chaque interaction, 4 étant l'interaction préférée. Les moyennes sur les 16 sujets de ces notations qualitatives sont présentés à la figure 5.10.

Un test de la somme des rangs de Friedman effectué sur les données révèle un effet significatif de la technique d'interaction sur les préférences des participants, et ce pour les trois critères de notation (Friedman chi-squared  $> 15.9, df = 3, p < 0.01$ ).

Nous avons effectué des comparaisons deux-à-deux à l'aide de tests de la somme des rangs de Wilcoxon avec correction de Bonferroni, pour chaque critère. Concernant les préférences générales, Arcball+ est significativement plus appréciée que TAV<sup>2</sup> et Arcball<sup>2</sup> ( $p < 0.01$ ). Pour les préférences suivant le critère de précision, on a trouvé des différences significatives entre chacune des techniques surjectives (TAV+ et Arcball+) et chacune des techniques non-surjectives (TAV<sup>2</sup> avec  $p < 0.05$  et Arcball<sup>2</sup> avec  $p < 0.01$ ). Les participants expriment des préférences similaires pour le critère de vitesse, avec des différences significatives entre chaque technique surjective et chaque technique non surjective ( $p < 0.01$ ). Dans tous les cas de différences significative, les techniques surjectives sont préférées par rapport aux techniques non surjectives.

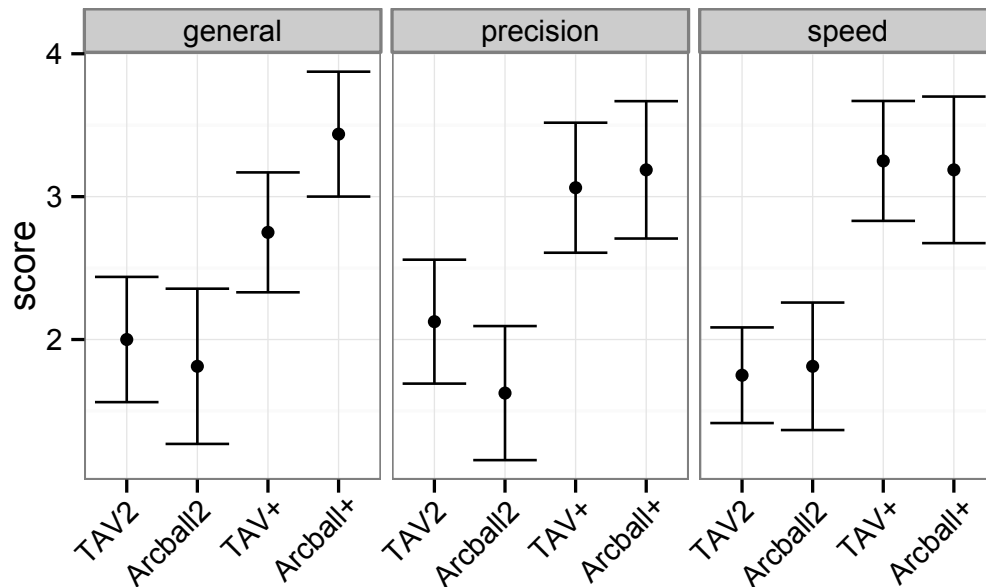


FIGURE 5.10 – Évaluation subjective de chaque technique d’interaction pour les 3 critères de notation. Les interactions présentant un plus haut score sont celles préférées par les participants.

## 5.6 Discussion

Le résultat principal de cette expérience est que les participants ont été en moyenne 18% plus rapides avec les techniques d’interaction surjectives et intégrales. L’amélioration du temps de réalisation ne semble pas avoir dégradé la précision des participants, puisque nous n’avons pas trouvé de différence significative à ce niveau là. Il faut souligner qu’aucun de nos participants n’avait d’expérience préalable en interaction 3D, et que la moitié d’entre eux ne travaillait pas dans le domaine de l’informatique. Par conséquent, cette expérience fournit une première mesure objective du bénéfice de la surjection et du contrôle intégral des trois axes de rotation pour des techniques de rotations sur surfaces planes tactiles multipoints, et de l’amélioration notable des performances.

Un second résultat de cette expérience concerne la capacité des participants à réaliser un contrôle intégral des rotations. En effet, offrir des techniques d’interactions intégrales ne signifie pas pour autant que les utilisateurs vont être capable d’exploiter cette possibilité. TAV<sup>2</sup> n’est pas une technique intégrale. Il est donc logique que sa valeur MDS soit éloignée de 0 (i.e. d’un contrôle entièrement intégral). Arcball et Arcball<sup>2</sup> ont été considérées comme des techniques partiellement intégrales, ce qui peut expliquer que les valeurs MDS soient proches mais légèrement supérieures à 0.5. Les meilleurs résultats de MDS sont obtenus avec les techniques TAV+ et Arcball+ (sans différence significative). Ainsi, avec ces techniques, le contrôle réalisé par les participant est plus intégral qu’avec les autres techniques, ce qui confirme nos hypothèses. Cependant, comme ces techniques

sont surjectives, et permettent d'ajuster de manière continue la trajectoire pour atteindre l'orientation cible, les utilisateurs ne manipulent parfois qu'un ou deux degrés de liberté. De plus, la tâche de docking peut être décomposée en deux phases successives : une navigation à gros grain dans un premier temps puis un ajustement de l'orientation finale, et nous supposons que cet ajustement précis nécessite moins d'intégralité que la première phase. Ces deux aspects peuvent expliquer pourquoi la valeur de MDS n'est pas plus proche de 0.

Finalement, les participants ont exprimé une préférence pour les techniques d'interaction surjectives. La technique Arcball+ a été particulièrement appréciée, dans la mesure où elle a été régulièrement notée comme meilleure technique. TAV+ a été presque aussi bien noté que Arcball+. Cependant, dans les préférences générales, sa notation n'est pas significativement meilleure que celle des techniques non-surjectives. Au delà de ces résultats, la surjection et l'intégration ont été clairement perçues et appréciées par les participants. Ils ont en effet trouvé plus facile d'effectuer la tâche "sans avoir à changer de mode". Les participants qui ont préféré l'Arcball+ à TAV+ ont expliqué que le mouvement de l'objet pendant la rotation était plus "naturel".

## Bilan

Dans ce chapitre nous avons introduit un nouveau critère pour les techniques d'interaction : la surjection. Nous avons utilisé la surjection, ainsi que l'intégralité et d'autres critères d'ergonomie, pour concevoir deux nouvelles techniques de rotations 3D : Arcball+ et TAV+.

Nous avons réalisé une étude utilisateur dans laquelle les résultats, qu'ils soient quantitatifs ou qualitatifs, ont démontré les bénéfices des techniques Arcball+ et TAV+ que nous proposons. Les participants ont pu effectuer un contrôle plus intégral des trois axes de rotation, et ainsi améliorer leurs performances par rapport aux techniques classiques. Comme les participants n'avaient pas d'expérience préalable en interaction 3D, ces résultats indiquent que ces techniques sont appropriées pour des applications destinées à des utilisateurs novices.

Dans le chapitre suivant, nous présentons notre troisième piste de recherche : l'évaluation expérimentale de différentes formes d'interaction couplant navigation et encrage pour la peinture numérique 3D.

## Chapitre 6

# Interactions bimanuelles et unimanuelles pour la peinture 3D

### Sommaire

---

<i>Introduction</i> . . . . .	110
<b>6.1 Choix des interactions</b> . . . . .	<b>110</b>
6.1.1 Peinture . . . . .	111
6.1.2 Navigation . . . . .	111
6.1.2.1 Interaction au stylet . . . . .	111
6.1.2.2 Interaction tactile . . . . .	112
6.1.2.3 Interaction 6dof . . . . .	113
6.1.2.4 Choix du point de pivot . . . . .	113
<b>6.2 Choix de la tâche</b> . . . . .	<b>113</b>
6.2.1 Un compromis entre tâche écologique et tâche expérimentale . . . . .	113
6.2.2 Des zones de difficultés variées . . . . .	114
<b>6.3 Contraintes d'implémentation</b> . . . . .	<b>116</b>
6.3.1 Dépliage UV . . . . .	116
6.3.2 Projection painting . . . . .	116
<b>6.4 Évaluation expérimentale</b> . . . . .	<b>118</b>
6.4.1 Matériel . . . . .	118
6.4.2 Tâche . . . . .	118
6.4.3 Conditions . . . . .	119
6.4.4 Protocole . . . . .	120
6.4.5 Mesures . . . . .	121
<b>6.5 Résultats</b> . . . . .	<b>121</b>
6.5.1 Temps . . . . .	122
6.5.2 Nombre de traits . . . . .	123

---

6.5.3	Vitesse de remplissage . . . . .	124
6.5.4	Évaluation subjective . . . . .	125
<b>6.6</b>	<b>Discussion . . . . .</b>	<b>126</b>
6.6.1	Analyse de la peinture . . . . .	126
6.6.2	Analyse de la navigation . . . . .	127
6.6.3	Analyse de la difficulté . . . . .	128
6.6.4	Synthèse . . . . .	128
	<b>Bilan . . . . .</b>	<b>129</b>

---

## Introduction

Les études décrites dans les chapitres précédents nous ont permis d'étudier séparément les interactions de peinture et de navigation pour les novices. Elles ont notamment mis en évidence que (1) l'interaction de peinture numérique directe est plus appropriée que l'interaction de peinture numérique indirecte et que (2) les techniques de rotation Arcball+ et TAV+ sont plus efficaces que celles de l'état de l'art et sont préférées par les participants. Cependant, les performances d'une interface de peinture numérique 3D dépendent directement du couplage des interactions de peinture et de navigation.

Dans ce chapitre, nous étudions différentes formes de couplages pour la peinture numérique 3D. Plus précisément, nous couplons la peinture sur écrans interactifs avec différentes techniques et dispositifs de navigations 3D, que nous avons sélectionnées d'après notre état de l'art et le chapitre précédent. Pour cela, nous avons choisi une tâche de peinture 3D suffisamment écologique pour être représentative des besoins courants en peinture 3D, mais également suffisamment contrainte pour permettre de comparer les différentes techniques d'interactions entre elles.

Cette étude expérimentale nous permet ainsi de mettre en évidence les formes d'interaction appropriées à un usage novice de la peinture numérique 3D.

### 6.1 Choix des interactions

Nous avons sélectionné durant l'état de l'art un ensemble de dispositifs d'entrée appropriés à la peinture 3D, dont la répartition est synthétisée dans le tableau 6.1. Afin de sélectionner les interactions que nous allons comparer dans notre étude expérimentale, nous allons considérer dans cette partie les interactions appropriées à ces dispositifs et à la tâche de peinture 3D. Nous étudierons dans un premier temps la sous-tâche de peinture numérique, puis nous passerons à la sous-tâche de navigation 3D.

	Bimanuel		Unimanuel
	<i>Main dominante</i>	<i>Main non dominante</i>	<i>Main dominante</i>
<b>Peinture</b>	Stylet	x	Stylet
<b>Navigation</b>	x	Tactile / 6dof	Stylet / Tactile

TABLE 6.1 – Dispositifs d’entrée à étudier pour la peinture numérique 3D

### 6.1.1 Peinture

Si l’on se réfère au tableau 6.1, le stylet est le dispositif d’entrée à utiliser pour la sous-tâche de peinture numérique. Le statu quo en terme de geste de peinture numérique au stylet consiste à appuyer la pointe du stylet sur la surface d’acquisition pour commencer un trait, puis à la faire *glisser* sur la surface pour déterminer la trajectoire du trait, et enfin à relever la pointe pour terminer le trait. Il est également possible de gommer en utilisant l’extrémité opposée du stylet de manière similaire.

L’interaction au stylet peut prendre deux formes principales, à savoir l’interaction directe ou l’interaction indirecte (cf. section 1.2.2). Cependant, l’interaction directe est particulièrement appropriée à la peinture numérique dans le cadre d’un usage novice (cf. chapitre 4). Nous allons donc dans cette étude considérer une interaction de peinture numérique au stylet sous forme directe, à l’aide d’un écran interactif.

Concernant le type d’interaction de peinture 3D, nous allons étudier l’interaction *projection painting* car elle est adaptée à un usage novice (cf. section 1.3.1.5). L’implémentation choisie pour cette interaction est détaillée ultérieurement à la section 6.3.

### 6.1.2 Navigation

Pour la sous-tâche de navigation 3D, le tableau 6.1 nous propose trois dispositifs d’entrée différents que nous allons détailler ici, à savoir le stylet, les surfaces tactiles, et les joysticks 6dof (cf. section 3.2.1). Nous allons également considérer les interactions de navigation de type navigation libre, qui permettent d’atteindre tous les placements possibles de la caméra (cf. chapitre 2).

#### 6.1.2.1 Interaction au stylet

Le statu quo que l’on retrouve dans les logiciels de graphisme pour la navigation au stylet consiste à manipuler la caméra suivant une interaction de type *Rotate/Pan/Zoom*

(cf. section 2.1.1). L'utilisateur a accès séparément à chacune de ces trois interactions via des modes activés par des boutons dans l'interface, des raccourcis claviers ou des boutons physiques.

Dans chaque mode, l'interaction est effectuée par survol de la tablette avec le stylet. Les mouvements décrits par celui-ci sont alors associés

- soit à des translations dans le plan de l'écran : les deux degrés de liberté du stylet et des translations sont alors en correspondance directe (cf. section 2.1.1.1)
- soit à une translation selon l'axe perpendiculaire à l'écran : les deux degrés de liberté du stylet sont agrégés en un seul pour contrôler la profondeur, à l'aide d'une fonction de type *distance* (cf. section 2.1.1.1)
- soit à des rotations, de type Arcball ou TAV (cf. section 2.1.1.2)

Concernant l'interaction de rotation, dans notre étude expérimentale notre choix se portera sur le TAV : cette interaction permet en effet de manipuler la caméra indépendamment de la position de départ du geste sur l'écran (cf. section 5.3). L'utilisateur a donc la possibilité d'éviter de placer sa main sur l'objet, contrairement à l'Arcball, et par conséquent d'avoir une meilleure visibilité sur celui-ci durant la navigation.

### 6.1.2.2 Interaction tactile

En interaction tactile multipoint, la métaphore *Rotate/Pan/Zoom* est également largement répandue pour la manipulation libre de la caméra (cf. section 2.1.1). L'implémentation du *Pan* et du *Zoom* est relativement standardisée :

- les translations dans le plan de l'écran sont contrôlées par un geste de *glisser* avec une correspondance directe des degrés de liberté du point de contrôle et des translations (cf. section 2.1.1.1).
- la translation suivant l'axe perpendiculaire à l'écran est contrôlée par la distance entre deux doigts lors d'un geste de *pincer* (cf. section 2.1.1.1).

Il existe de nombreuses interactions possibles pour les rotations (cf. sections 2.1.1.2 et 2.1.2). Celles que nous proposons au chapitre 5, à savoir TAV+ et Arcball+, sont particulièrement efficaces dans le cadre d'un usage novice (cf. chapitre 5). La technique TAV+ présente cependant l'avantage de permettre la manipulation de la caméra indépendamment de la position de départ du geste sur l'écran (cf. section 5.3), ce qui apporte une meilleure visibilité de l'objet à peindre. Nous allons donc utiliser cette interaction de rotation.



### 6.1.2.3 Interaction 6dof

Les *joystick 6dof* permettent de contrôler simultanément les six degrés de liberté de la caméra en les associant directement aux degrés de liberté du dispositif. Ware et al. explorent deux métaphores possibles pour cette association[202] :

**EyeBall in hand** Dans cette interaction, la position et l'orientation du dispositif d'entrée sont directement associées à la position et l'orientation de la caméra.

**Scene in hand** Il s'agit de l'interaction duale de la précédente : la position et l'orientation du dispositif d'entrée sont associées à celles de l'environnement.

Cette seconde métaphore est particulièrement appropriée au cas de l'observation d'un unique objet dans la scène, c'est pourquoi nous choisissons de l'implémenter.

### 6.1.2.4 Choix du point de pivot

La question du choix du point de pivot se pose pour les différentes techniques de rotations choisies. Une première possibilité est de prendre comme pivot le centre de l'objet, de manière à faire tourner la caméra autour de celui-ci. Cependant, lorsque l'utilisateur se rapproche de l'objet pour peindre des détails, une rotation autour du centre de l'objet peut amener la zone d'intérêt hors de l'écran.

Une solution est donc de définir dynamiquement le point de pivot de manière à le faire correspondre à la zone d'intérêt. Ainsi, les rotations effectuées permettent de changer de point de vue sur la zone d'intérêt, tout en restant centré sur celle-ci.

Il reste alors à définir comment choisir la zone d'intérêt. Dans notre étude, nous avons choisi d'implémenter le comportement présent dans le logiciel Zbrush, qui consiste à définir comme point de pivot le dernier point du modèle peint à l'aide du stylet.

## 6.2 Choix de la tâche

### 6.2.1 Un compromis entre tâche écologique et tâche expérimentale

Nous avons souhaité étudier une tâche de peinture sur modèle 3D qui soit écologique. Pour cela, nous avons souhaité reproduire le comportement d'un artiste qui applique des couleurs sur différentes zones d'un objet 3D, de manière précise. Nous avons donc considéré dans un premier temps une tâche où le participant doit peindre des zones de

l'objet, sélectionnées pour leurs caractéristiques topologiques (voir section 6.2.2), puis valider manuellement lorsqu'il considère la tâche comme terminée.

Nous avons testé cette tâche à l'aide d'une première étude pilote, avec 8 participants. Nous avons observé que le travail demandé n'était pas assez contraint, ce qui amenait les participants à adopter des comportements trop différents pour être comparables : certains ont peint les zones de manière méthodique et successive alors que d'autres ont peint les zones de manière erratique jusqu'à être satisfait. Ce deuxième type de comportement provoque une perte de temps nécessaire à la vérification des zones peintes, et conduit régulièrement à l'oubli de certaines zones. Les résultats obtenus à l'issue de cette étude pilote étaient donc inexploitable du point de vue quantitatif.

Nous avons donc considéré un autre type de tâche, plus contraint, où le participant se voit proposer successivement différentes zones à peindre. La validation des zones est faite automatiquement par le système, ce qui évite les pertes de temps parasites. La tâche est décrite en détail à la section 6.4.2.

### 6.2.2 Des zones de difficultés variées

Le choix du modèle 3D à peindre s'est porté sur un personnage de type souris-humanoïde. Ce choix nous a paru judicieux dans la mesure où ce type de modèle est représentatif de la peinture de personnages 3D, qui est une tâche répandue en peinture numérique 3D.

Les zones à peindre ont été choisies en fonction de leur *complexité* topologique. En effet, plus la topologie est complexe, plus la tâche va nécessiter de changements de points de vue et donc d'entrelacement entre la navigation et la peinture. Par exemple, le ventre du modèle est une zone plane, très peu complexe, qui peut être peinte depuis un unique point de vue, tandis que la main est complexe et nécessite beaucoup de points de vue différents (voir figure 6.1).

Nous avons considéré différentes zones, puis les avons classées par ordre de difficulté (fortement liée à la complexité) de manière empirique. Pour cela, nous avons réalisé une étude pilote impliquant 5 participants et nous avons mesuré les vitesses moyennes de remplissage pour chaque zone en pixel/seconde. Nous faisons l'hypothèse qu'une zone peinte à vitesse élevée correspond à une zone facile.

Parmi toutes les zones que nous avons considérées, nous avons alors choisi pour notre étude des zones de difficulté moyenne, nécessitant plusieurs points de vue pour être peintes, mais accessibles à des participants novices. Notre choix s'est donc porté sur le nez, l'extérieur d'une oreille, l'intérieur d'une autre oreille, et le sourcil (voir figure 6.2). Ces zones sont cependant de difficultés variables les unes par rapport aux autres. Par

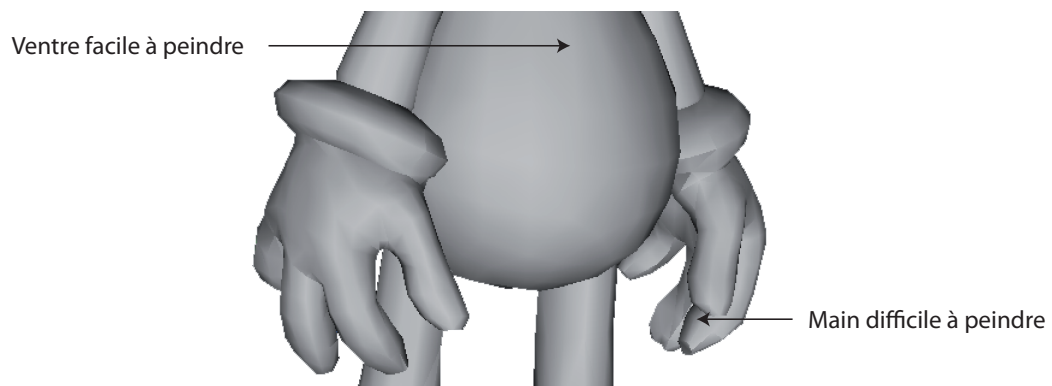


FIGURE 6.1 – Difficulté des zones : le ventre est une zone relativement plane, qui est facile à peindre car elle nécessite peu de changements de points de vue. Au contraire, la main est complexe et difficile à peindre : elle nécessite beaucoup de changements de point de vue.

exemple, le nez est plus facile à peindre que les oreilles, qui sont elles mêmes plus faciles que le sourcil :

**Nez** Cette zone est complètement convexe et régulière, ce qui rend ses faces faciles d'accès

**Extérieur de l'oreille** Cette zone comprend une partie convexe ainsi qu'un retour abrupt, elle est donc un peu plus difficile à peindre

**Intérieur de l'oreille** Cette zone est concave, elle comprend un recoin assez profond au niveau du conduit auditif, elle est d'une difficulté similaire à la zone précédente

**Sourcil** Cette zone est très fine avec des rebords abrupts, elle est plus difficile à peindre que les zones précédentes.

Ces niveaux de difficulté définis empiriquement à l'aide du pilote sont confirmés par les vitesses mesurées lors de l'étude expérimentale (cf. section 6.5.3).

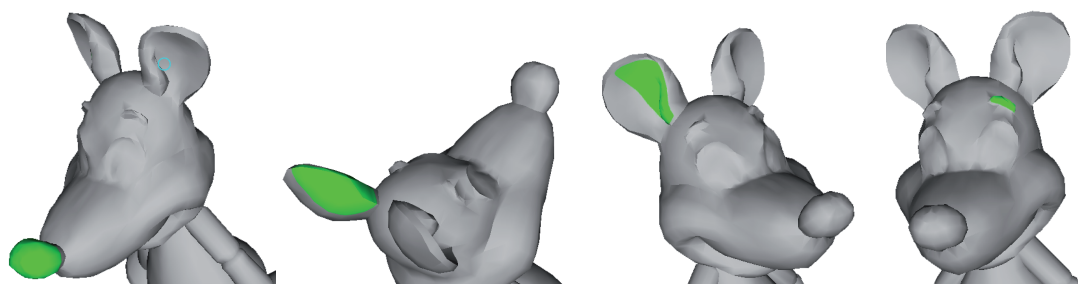


FIGURE 6.2 – Zones à peindre

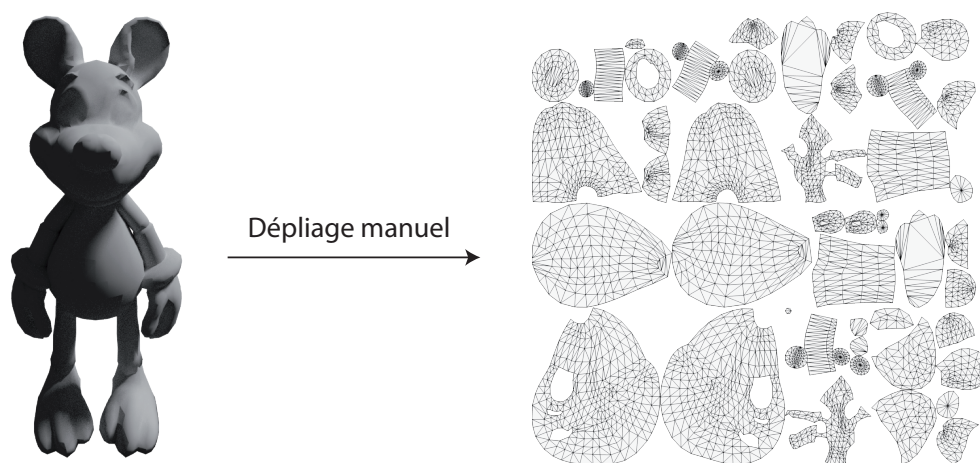


FIGURE 6.3 – Carte UV du modèle de l'étude expérimentale

## 6.3 Contraintes d'implémentation

Nous avons dû faire face à deux contraintes techniques majeures pour mener à bien notre étude expérimentale, à savoir l'implémentation du *dépliage UV* du modèle à peindre, et l'implémentation du *projection painting*. Nous allons détailler dans cette partie les solutions techniques choisies.

### 6.3.1 Dépliage UV

Notre tâche expérimentale nécessite un unique modèle représentant une souris. Nous avons donc décidé d'effectuer manuellement le dépliage nécessaire de ce modèle à l'aide du logiciel Blender. Un algorithme de dépliage n'est pas justifié pour un seul modèle, et de plus, les algorithmes de dépliage automatique ne produisent pas de résultat aussi bon qu'un dépliage manuel.

Nous avons défini les coutures manuellement, puis nous avons également ajusté manuellement *a posteriori* la distorsion du graphe planaire obtenu. La carte UV du modèle est présentée à la figure 6.3.

### 6.3.2 Projection painting

Nous avons décrit en section 1.3.3 plusieurs variantes de projections pour l'interaction *projection painting* :

- la *parameter-space brush*
- la *screen-space brush*

— la *tangent-space brush*

Ces techniques présentent différents niveaux de difficulté d'implémentation, à savoir que la *parameter-space brush* est plus simple que la *screen-space brush* qui est elle-même plus facile à implémenter que la *tangent-space brush*. Cependant, nous avons également vu que la *parameter-space brush* amène des problèmes de distorsion de la peinture tandis que la *screen-space brush* donne un effet de spray dû à sa projection pixel-à-pixel (cf. section 1.3.3). Afin de palier à ces problèmes, nous avons implémenté une combinaison de ces deux types de projections.

Nous projetons tout d'abord les pixels de l'empreinte de la brosse (sur l'écran) dans la texture, à la manière de *screen-space brush*. Ensuite, au lieu de colorer uniquement les projections, ce qui provoque l'apparition de vides et donne l'effet de spray, nous utilisons la technique *parameter-space brush* individuellement pour chaque projection, avec un rayon de quelques pixels seulement. Ainsi les zones colorées se recouvrent, et donnent l'apparence d'une empreinte connexe (figure 6.4).

Ceci nous apporte un compromis entre la simplicité d'implémentation et la qualité de l'interaction. En effet, nous avons pu constater que le comportement obtenu avec ce type de projection a un rendu identique à celui de certains logiciels de peinture 3D (Zbrush, Blender). Cependant les performances de cette projection sont limitées : elles sont suffisantes dans le cadre de notre étude, mais une utilisation sur de plus grandes textures conduit à accroître la latence de l'interaction.

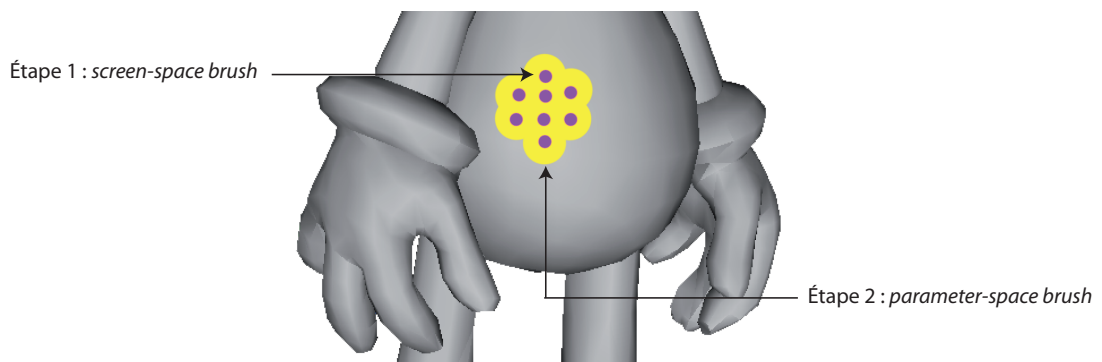


FIGURE 6.4 – Implémentation du *projection painting*

## 6.4 Évaluation expérimentale

### 6.4.1 Matériel

Le prototype expérimental est exécuté sur un ordinateur portable équipé d'un processeur IntelCore i7 (2.6Ghz) et d'une carte graphique NVidia GT 650M. L'application est programmée en C#, C++, objective-C et C dans Unity3D. Les participants utilisent un écran interactif tactile de 24 pouces à sa résolution native de 1920\*1200 pixels (Wacom Cintiq 24HD touch). La sensibilité à la pression du stylet est désactivée pendant l'expérience. Pour certaines conditions expérimentales, les utilisateurs manipulent également un *joystick 6dof*<sup>1</sup>.

### 6.4.2 Tâche

La tâche proposée aux participants consiste à peindre des zones successives sur un modèle 3D représentant une souris (figure 6.5). La zone à peindre apparaît en vert sur le modèle. Les participants doivent peindre cette zone à l'encre bleue à l'aide du stylet. S'ils dépassent en dehors de la zone verte, les dépassements apparaissent à l'encre rouge. Un taux d'erreur de remplissage est affiché à l'écran. Celui-ci est défini comme la somme des pixels manquants (verts) et des pixels de débordement (rouges) divisée par le nombre de pixels à peindre dans la texture sous-jacente :

$$\text{erreur} = \frac{\#\text{pixels manquants} + \#\text{pixels de débordement}}{\#\text{pixels à peindre}}$$

Nous demandons aux participants de faire descendre ce taux en dessous de 5%. Au passage de ce seuil la zone est validée automatiquement puis la zone suivante est affichée. Cette approche évite le problème observé pendant les études pilotes : les participants pouvaient passer à une autre zone du modèle sans se rendre compte qu'il leur restait à peindre une grande partie cachée de la zone.

Les participants doivent peindre quatre zones distinctes, choisies pour la complexité de leur topologie : le nez, un sourcil, l'extérieur d'une oreille, l'intérieur de l'autre oreille (figure 6.2). Chacune de ces zones nécessite de changer de point de vue plusieurs fois pour être peinte en intégralité.

---

1. 3Dconnexion SpaceNavigator

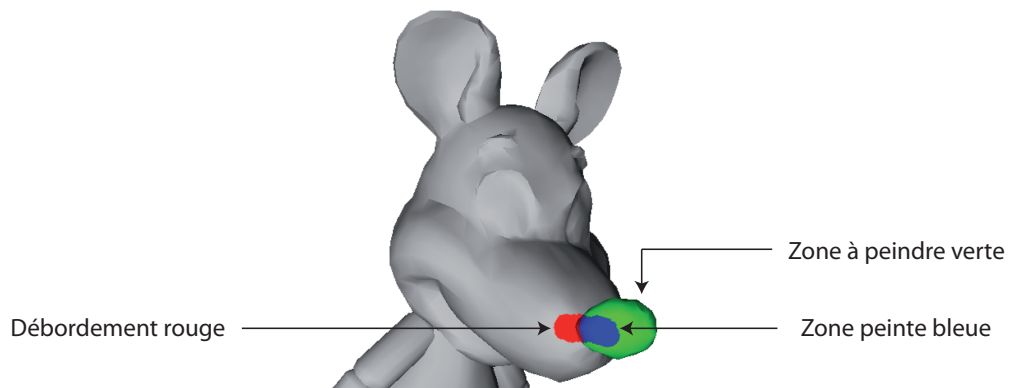


FIGURE 6.5 – Tâche

### 6.4.3 Conditions

L'interaction de peinture est restée la même tout au long de l'expérience : les participants peignent avec la pointe du stylet et peuvent gommer leurs débordements avec la gomme numérique présente de l'autre côté du stylet. Les participants effectuent la tâche de peinture 3D via quatre interactions de navigation distinctes :

**Stylet** L'interaction est unimanuelle modale. Le participant navigue en *survolant* la tablette avec la pointe du stylet. Il maintient enfoncé un bouton physique du stylet ou de l'écran interactif pour activer un *mode maintenu* de navigation. Le *bouton bas* du stylet active les translations dans le plan de l'écran. Un bouton sur le bord de l'écran interactif active la translation dans le plan perpendiculaire à l'écran : la caméra s'approche de l'objet lorsque le stylet survole l'écran vers le haut, et inversement. Le *bouton haut* du stylet active les rotations selon l'interaction TAV. Seules les rotations dans le plan de l'écran sont possibles. Cette combinaison d'interactions activées par modes maintenus est similaire à ce qui est offert dans les logiciels professionnels.

**T2M** *Tactile Deux Mains* : L'interaction est bimanuelle au stylet et au touché. La main non dominante manipule la caméra via l'interaction tactile. Les translations dans le plan de l'écran sont effectuées par un geste de *glisser* à un doigt. La translation selon l'axe perpendiculaire à l'écran correspond au geste de *pincer* à deux doigts. Les rotations suivent l'implémentation du TAV+ du chapitre précédent : les rotations suivant les axes dans le plan de l'écran correspondent à un geste de *glisser* à deux doigts tandis que la rotation suivant l'axe perpendiculaire à l'écran correspond à un geste de rotation à deux doigts.

**T1M** *Tactile Une Main* : C'est l'équivalent unimanuel modal de T2M. La navigation est effectuée de manière identique à T2M mais par la main dominante.

Le changement de mode s'effectue simplement en changeant la prise en main du stylet : en position de dessin ou en travers des doigts. Les phases de peinture et de navigation ne peuvent donc pas se superposer.

**6dof** L'interaction est bimanuelle. La main non dominante manipule la camera à l'aide du joystick 6dof. Les translations sont effectuées en appliquant une pression sur le dispositif selon les axes correspondants. Les rotations suivant les axes dans le plan de l'écran correspondent à une inclinaison du dispositif selon ces axes. La rotation suivant l'axe perpendiculaire à l'écran correspond à un mouvement de torsion du dispositif.

#### 6.4.4 Protocole

Nous avons recruté 16 participants âgés de 24.5 ans en moyenne ([19-32],  $\text{stddev}=3.73$ ). Ils sont tous expérimentés en interaction tactile (utilisation quotidienne du Smartphone). Trois d'entre eux dessinent régulièrement mais aucun n'utilise régulièrement une tablette graphique ou un écran interactif. Aucun n'est familier avec un *joysticks 6dof*, mais 12 participants évoluent dans des environnements 3D de manière régulière pour les jeux vidéo.

L'expérience à lieu dans une pièce silencieuse. Les participants sont assis devant l'écran interactif, incliné à 30° selon l'horizontale.

Nous utilisons un design intra-sujet : chaque participant doit peindre les quatre zones successivement dans chacune des quatre techniques, avec trois répétitions. Nous enregistrons en tout  $16$  (participants)  $\times$   $4$  (techniques)  $\times$   $4$  (zones)  $\times$   $3$  répétitions = 768 zones peintes.

Chaque session débute par la présentation de la procédure, suivie d'une phase d'entraînement générale, pendant laquelle les participants peuvent se familiariser avec chacune des techniques de navigation sur l'ensemble des zones à peindre.

Afin de contrebalancer les effets d'apprentissage, l'ordre de présentation des techniques est équilibré entre les participants. En revanche, les types de zones à peindre sont toujours présentés dans le même ordre de difficulté (nez, intérieur de l'oreille, extérieur de l'oreille, puis sourcil).

Les participants notent ensuite les techniques selon leurs préférences dans un entretien post-expérimental. Les notes sont comprises entre 1 et 4, où 4 est la meilleure note. Les participants peuvent également commenter librement les différentes interactions.



### 6.4.5 Mesures

Le temps de réalisation de la tâche et le taux d'erreurs sont les variables dépendantes généralement utilisées dans ce type d'expérience. Cependant, par construction de la tâche expérimentale, le taux d'erreur est fixe à 5%. Nous pouvons donc nous focaliser sur l'étude des temps de réalisation. Nous décomposons ces temps en fonction des différentes actions effectuées :

**Temps total** temps total pour peindre l'ensemble des quatre zones.

**Navigation** temps de navigation cumulé pour les quatre zones.

**Combinaison** temps passé à peindre et naviguer simultanément cumulé pour les quatre zones.

**Inactif** temps où le participant est inactif cumulé pour les quatre zones.

**Encrage** temps d'encrage (utilisation de la pointe du stylet) cumulé pour les quatre zones.

**Gommage** temps de gommage (utilisation de la gomme du stylet) cumulé pour les quatre zones.

**Usage du stylet** temps total d'utilisation du stylet (Encrage + Gommage) cumulé pour les quatre zones.

Nous avons également mesuré la vitesse de remplissage, afin d'avoir une mesure normalisée pour les différentes zones :

**Vitesse** vitesse de peinture pour chaque zone en pixel/s.

Enfin, nous avons compté le nombre de traces de stylets, c'est à dire de traits dessinés à l'écran, et nous avons raffiné l'analyse en séparant tracé pour l'encrage et tracé pour gommer.

**Encre** nombre de traits d'encrage cumulés pour les quatre zones.

**Gomme** nombre de traits de gomme cumulés pour les quatre zones.

**Stylet** nombre de traits global (encre+gomme) cumulés pour les quatre zones.

## 6.5 Résultats

Nous avons agrégé les données de chaque mesure en calculant la moyenne pour chaque combinaison *participant*  $\times$  *technique*. Nous avons ensuite mesuré l'effet de l'interaction sur les différentes mesures à l'aide d'ANOVA par mesures répétées à un facteur, et nous avons utilisé des t-test appariés bilatéraux pour les comparaisons post-hoc, avec une correction de Bonferroni.

### 6.5.1 Temps

Nous présentons dans cette partie les résultats concernant les différentes mesures de temps effectuées durant l'étude expérimentale.

**Temps total** Le temps total est représenté à la figure 6.6a. L'ANOVA indique un effet global significatif des techniques ( $p < 0,001$ ). Les t-test montrent des différences significatives pour les couples : STYLET vs. T1M ( $p < 0.005$ ,  $-17\%$ ), STYLET vs. T2M ( $p < 0.001$ ,  $-21\%$ ), 6DOF vs. T1M ( $p < 0.005$ ,  $-19\%$ ) et 6DOF vs. T2M ( $p < 0.0001$ ,  $-23\%$ ).

**Navigation** Le temps de navigation est représenté à la figure 6.6b. L'ANOVA indique un effet global significatif des techniques ( $p < 0,001$ ). Les t-test montrent

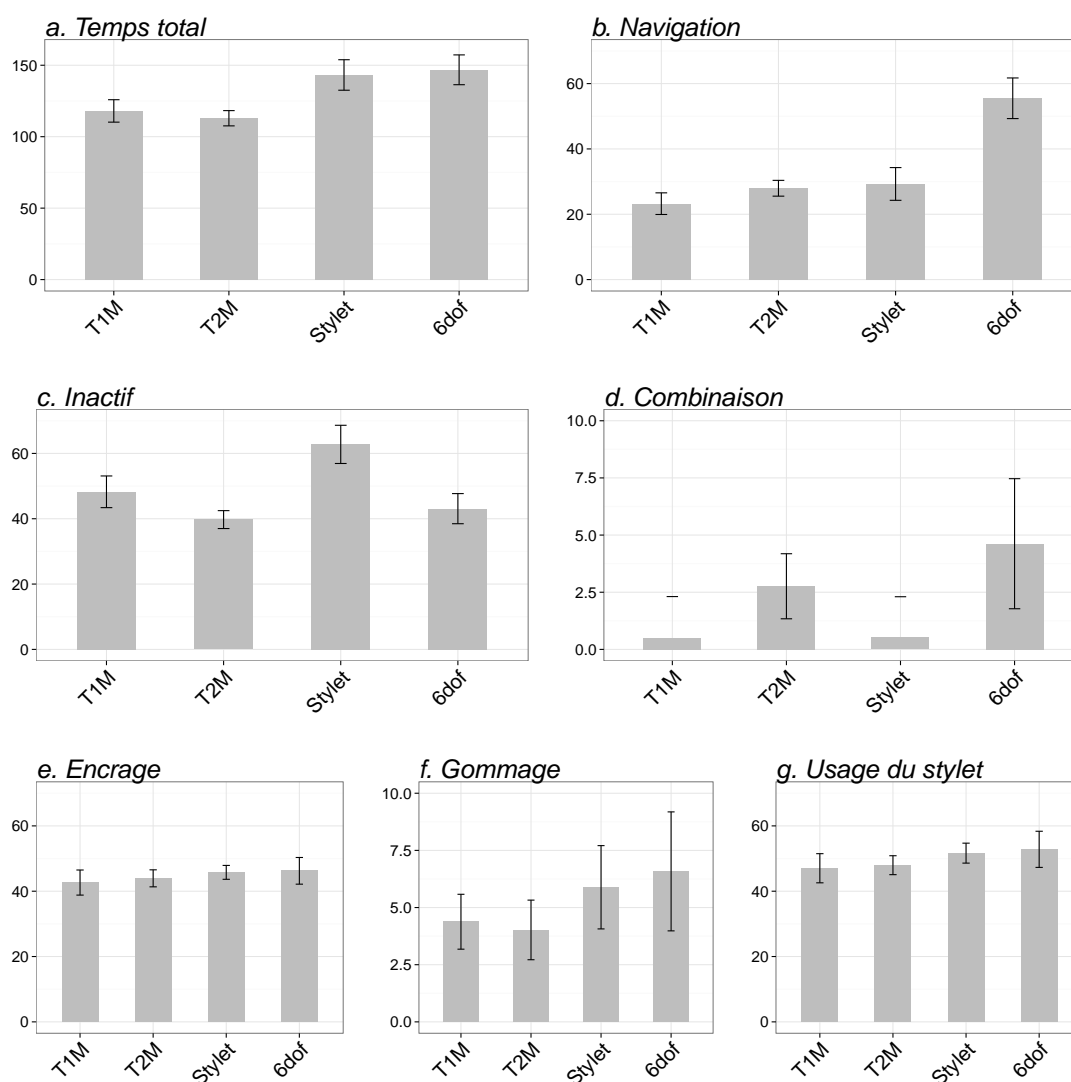


FIGURE 6.6 – Résultats des mesures de temps en secondes : moyennes sur tous les participants et intervalles de confiance à 95%

des différences significatives pour les couples : T2M vs. T1M ( $p < 0.05$ ,  $-17\%$ ), STYLET vs. T1M ( $p < 0.05$ ,  $-20\%$ ), 6DOF vs. STYLET ( $p < 0.0001$ ,  $-47\%$ ), 6DOF vs. T1M ( $p < 0.0001$ ,  $-58\%$ ) et 6DOF vs. T2M ( $p < 0.0001$ ,  $-49\%$ ).

**Inactif** Le temps d'inaction est représenté à la figure 6.6c. L'ANOVA indique un effet global significatif des techniques ( $p < 0,001$ ). Les t-test montrent des différences significatives pour les couples : T1M vs. T2M ( $p < 0.001$ ,  $-17\%$ ), STYLET vs. T1M ( $p < 0.05$ ,  $-23\%$ ), STYLET vs. T2M ( $p < 0.001$ ,  $-36\%$ ), STYLET vs. 6DOF ( $p < 0.0001$ ,  $-31\%$ ).

**Combinaison** Le temps d'interaction combinée est représenté à la figure 6.6d. L'ANOVA indique un effet global significatif des techniques ( $p < 0,01$ ). Les t-test montrent des différences significatives pour les couple T2M vs. T1M ( $p < 0.01$ ,  $-100\%$ ), T2M vs. STYLET ( $p < 0.01$ ,  $-100\%$ ), 6DOF vs. T1M ( $p < 0.001$ ,  $-100\%$ ), 6DOF vs. STYLET ( $p < 0.001$ ,  $-100\%$ ). Ces résultats sont triviaux dans la mesure où il n'y a pas de combinaison en interaction unimanuelle.

**Encrage** Le temps d'encrage est représenté à la figure 6.7a. L'ANOVA n'indique pas d'effet global significatif des techniques ( $p > 0.05$ ).

**Gommage** Le temps de gommage est représenté à la figure 6.7b. L'ANOVA n'indique pas d'effet global significatif des techniques ( $p > 0.05$ ).

**Usage du stylet** Le temps total d'utilisation du stylet est représenté à la figure 6.7c. L'ANOVA n'indique pas d'effet global significatif des techniques ( $p > 0.05$ ).

### 6.5.2 Nombre de traits

Nous présentons dans cette partie les résultats concernant les nombres de traits mesurés durant l'étude expérimentale.

**Encre** Le nombre de traits d'encrage est représenté à la figure 6.7a. L'ANOVA indique un effet global significatif des techniques ( $p < 0,0001$ ). Les t-test montrent des différences significatives pour les couples : 6DOF/T1M ( $p < 0.01$ ,  $-19\%$ ) et 6DOF/T2M ( $p < 0.001$ ,  $-21\%$ ).

**Gomme** Le nombre de traits de gomme est représenté à la figure 6.7b. L'ANOVA indique un effet global significatif des techniques ( $p < 0,005$ ). Les t-test montrent des différences significatives pour les couples : 6DOF/T1M ( $p < 0.05$ ,  $-21\%$ ) et 6DOF/T2M ( $p < 0.05$ ,  $-26\%$ ).

**Stylet** Le nombre total de traits est représenté à la figure 6.7c. L'ANOVA indique un effet global significatif des techniques ( $p < 0,0001$ ). Les t-test montrent des différences significatives pour les couples : 6DOF/T1M ( $p < 0.01$ ,  $-22\%$ )

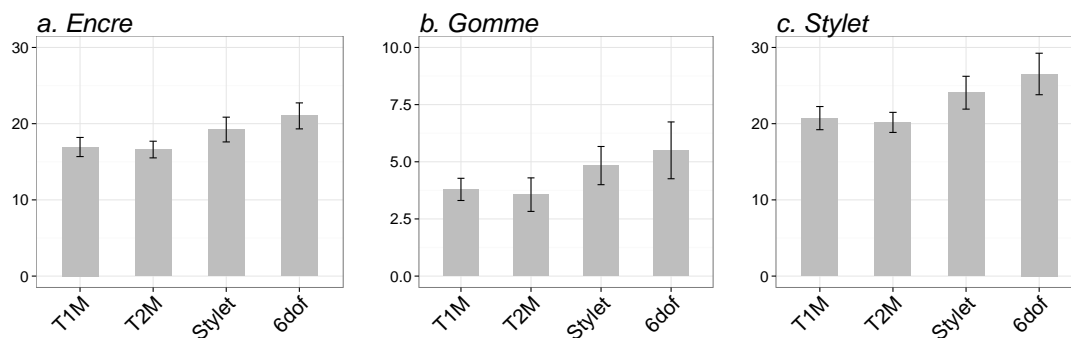


FIGURE 6.7 – Résultats des mesures de nombres de traits : moyennes sur tous les participants et intervalles de confiance à 95%

et 6DOF/T2M ( $p < 0.005$ ,  $-24\%$ ), STYLET/T1M ( $p < 0.01$ ,  $-14\%$ ) et STYLET/T2M ( $p < 0.005$ ,  $-16\%$ ).

### 6.5.3 Vitesse de remplissage

Nous présentons dans cette partie les résultats concernant la vitesse de remplissage mesurée durant l'étude expérimentale. La vitesse de remplissage est représentée par zone et par technique à la figure 6.8.

Nous avons mesuré l'effet de la technique d'interaction et de la zone peinte sur la vitesse de remplissage à l'aide d'une ANOVA par mesures répétées à deux facteurs. Celle-ci indique un effet de la technique d'interaction ( $p < 0.001$ ) et un effet de la zone peinte ( $p < 0.001$ ). Cependant, elle ne permet pas de mettre en évidence un effet de l'interaction des deux facteurs ( $p > 0.5$ ). Nous avons utilisé des t-test appariés bilatéraux pour les comparaisons post-hoc, avec une correction de Bonferroni.

Concernant l'effet des zones peintes, les t-test montrent des différences significatives pour les couples : Oreille intérieure/Sourcil ( $p < 0.001$ ,  $-52\%$ ), Nez/Sourcil ( $p < 0.001$ ,  $-69\%$ ), Oreille extérieure/Sourcil ( $p < 0.001$ ,  $-60\%$ ), Nez/Oreille intérieure ( $p < 0.001$ ,  $-35\%$ ), Oreille extérieure/Oreille intérieure ( $p < 0.001$ ,  $-17\%$ ) et Nez/Oreille extérieure ( $p < 0.01$ ,  $-22\%$ ).

Concernant l'effet de la technique d'interaction, Les t-test montrent des différences significatives pour les couples : T1M/STYLET ( $p < 0.01$ ,  $-13\%$ ), T2M/STYLET ( $p < 0.001$ ,  $-16\%$ ), T1M/6DOF ( $p < 0.01$ ,  $-15\%$ ) et T2M/6DOF ( $p < 0.001$ ,  $-19\%$ ).

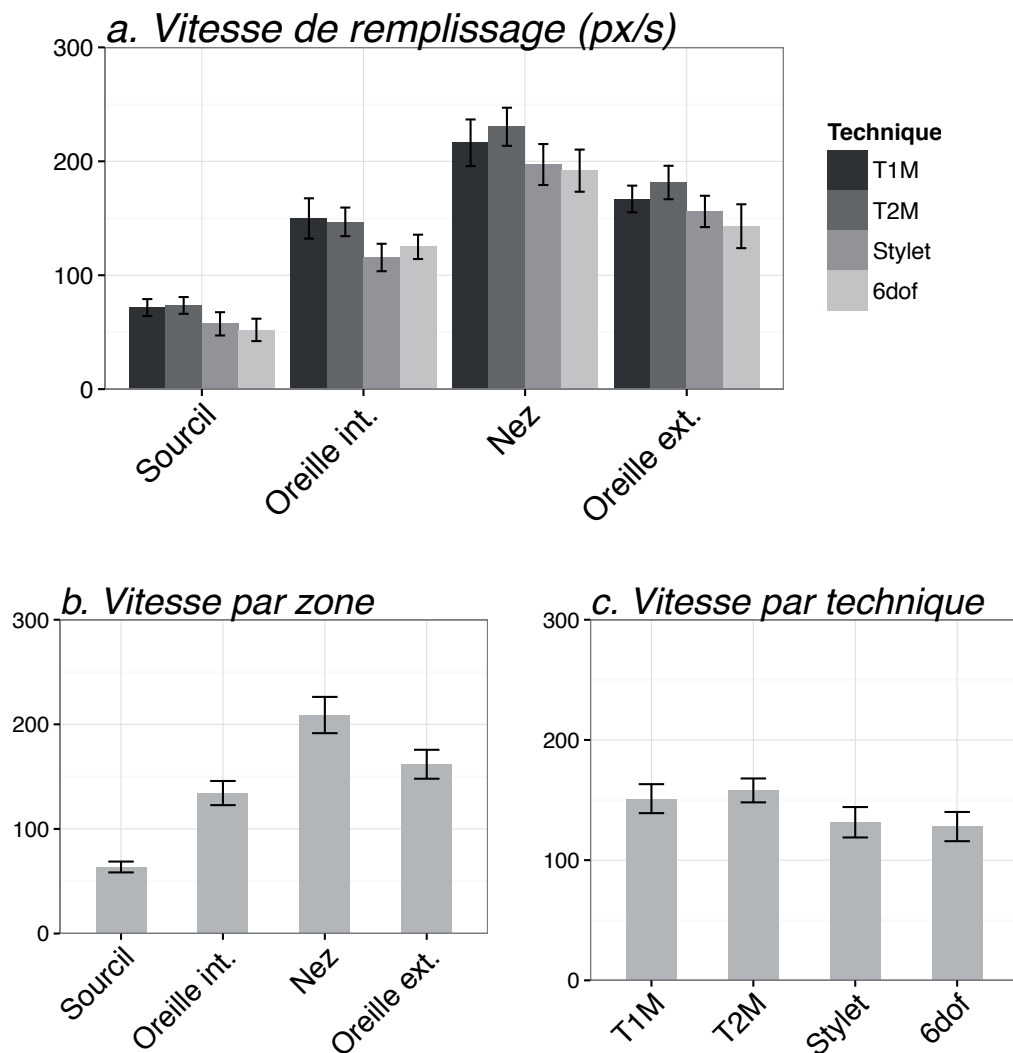


FIGURE 6.8 – Résultats des mesures de vitesse de remplissage : *a.* moyennes sur tous les participants par zones et par techniques. *b.* moyennes sur tous les participants et techniques par zones. *c.* moyennes sur tous les participants et zones par techniques. Les barres d'erreurs représentent les intervalles de confiance à 95%.

#### 6.5.4 Évaluation subjective

Les participants ont noté leurs préférences par un score allant de 1 à 4, la meilleure note étant 4. Les résultats sont présentés dans la figure 6.9. Un test de la somme des rangs de Friedman montre un effet significatif des techniques sur les notations des participants ( $p < 0.001$ ). Nous avons effectués des tests de la somme des rangs de Wilcoxon avec correction de Bonferroni. Ces derniers montrent que T2M a été significativement mieux notée que toutes les autres techniques ( $p < 0.05$ ). Ils montrent également que T1M a été mieux notée que STYLET et 6DOF ( $p < 0.01$ ). Nous n'avons pas trouvé de différence significative de notation entre STYLET et 6DOF. Cependant, nous avons remarqué que STYLET était globalement mieux noté (médiane plus élevée), alors que 6DOF était soit

mal notée, soit mieux notée que *STYLET*. Les participants nous ont donné des retours disparates quant à 6DOF : certains ont trouvé la technique “inutilisable” ou “difficile” alors que d’autres au contraire l’ont trouvée “pratique” et “intuitive”. Les retours concernant *STYLET* ont été unanimement négatifs mais de manière moins radicale que pour 6DOF.

## 6.6 Discussion

### 6.6.1 Analyse de la peinture

Nous n’observons pas de différence selon les techniques sur les temps passés à encrer et à gommer (et donc sur le temps total de peinture). Cela peut s’expliquer par le fait que la tâche et la technique de peinture restent identiques tout au long de l’expérience. Cela semble indiquer que la technique de navigation n’a pas d’influence quantitative sur l’exécution de la peinture.

Cependant, d’un point de vue qualitatif, les participants nous ont indiqué durant l’expérience que les techniques non-tactiles offraient un “confort d’exécution plus faible”. Avec ces techniques, ils préféraient peindre avec un point de vue “approximatif” de la caméra plutôt que de perdre du temps à manipuler la caméra et obtenir le point de vue idéal. Ce n’était pas le cas avec les techniques tactiles qui, selon les participants, permettaient d’atteindre les points de vue souhaités plus facilement et rapidement. En corrélant cette remarque avec les résultats obtenus sur les nombre de coups de stylet (pinceau et gomme confondus), nous pouvons supposer que les points de vue “approximatifs” ont induit davantage de modification de position de la main dominante pour s’adapter au mauvais

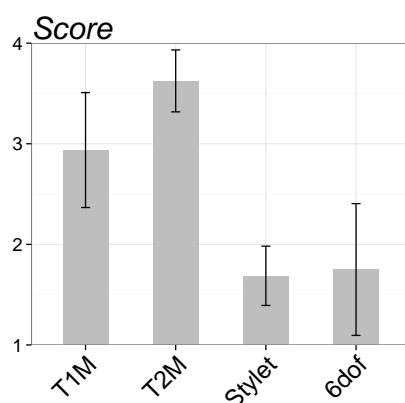


FIGURE 6.9 – Évaluation subjective de chaque technique d’interaction. Les interactions préférées par les participants ont un score élevé.

point de vue, et donc davantage de coups de stylet. Les techniques non-tactiles ont ainsi induit un confort d'exécution plus faible que les techniques tactiles.

### 6.6.2 Analyse de la navigation

On constate que les temps de navigation sont plus faibles avec les interactions tactiles. Nous l'expliquons par l'usage répandu des Smartphones multi-tactile. Nos participants, bien que novices en peinture numérique, possèdent tous un Smartphone alors qu'ils n'ont pas d'expérience avec le stylet ou le joystick 6dof. Les techniques tactiles ont sans doute bénéficié de l'expérience de nos participants en interaction au touché. Une autre interprétation est cependant possible : si les participants éprouvent des difficultés avec la navigation au touché, ils préféreront éviter de naviguer, ce qui amènera à un temps de navigation plus faible. Cependant, les notations subjectives de ces techniques sont meilleures. Cette hypothèse est donc peu probable.

On remarque que le temps de navigation en T1M est plus faible que celui e T2M. C'est un résultat attendu dans la mesure où la main dominante est plus rapide et précise que la main non dominante : les angle de vue souhaités sont atteints plus facilement lorsque la main dominante manipule la caméra.

Le joystick 6dof en revanche amène à de moins bonnes performances en terme de temps de navigation. C'est un dispositif difficile à appréhender par des utilisateurs novices. Plusieurs participants ont indiqué avoir eu des difficultés à l'utiliser. L'ensemble des participants a considéré que les performances auraient été améliorées s'ils avaient eu un entraînement plus important avec ce dispositif. C'est une hypothèse crédible qui nécessiterait d'autres investigations, cependant nous sortirions alors du cadre des utilisateurs novices.

Le temps de combinaison, où le participant navigue et peint simultanément, ne peut être que nul pour les techniques unimanuelles qui imposent un entrelacement dans le temps des deux modes. Cependant, il est resté faible même pour les techniques bimanuelles. Nous avons tout de même observé des disparités de comportement selon les participants. Généralement, ils n'ont pas réussi à paralléliser la peinture et la navigation car la charge cognitive semble trop forte. Mais certains participants ont indiqué qu'ils auraient aimé pouvoir paralléliser pour gagner en efficacité, et qu'ils pensaient y arriver avec de l'entraînement. Deux participants ont même rapidement réussi a paralléliser peinture et navigation sur les zones faciles à peindre (nez, partie convexe de l'oreille extérieure). Cette difficulté générale à paralléliser la peinture et la navigation n'est pas surprenante : Masliah et al. montrent que les utilisateurs n'arrivent pas à coordonner le contrôle de six degrés de liberté dans une tâche de positionnement et orientation dont la structure

est intégrale [154]. Il paraît donc encore plus délicat de coordonner le contrôle des six degrés de liberté de la navigation et des deux degrés de liberté de la peinture.

Le temps d'inaction est plus faible avec les techniques bimanuelles qu'avec les techniques unimanuelles. Ce résultat est attendu dans la mesure où les techniques unimanuelles comptabilisent un temps supplémentaire correspondant au changement de mode. Cependant on peut également proposer l'interprétation suivante : le temps d'inaction comprend également un temps de réflexion quand à la prochaine interaction à effectuer. On peut supposer que ce temps de réflexion est dans une certaine mesure parallélisé avec les techniques bimanuelles, et contribue avec l'absence de changement de mode à diminuer le temps d'inaction.

### 6.6.3 Analyse de la difficulté

On n'observe pas d'interaction des effets de la technique d'interaction et de la difficulté de la tâche sur la vitesse de remplissage (qui est la version normalisée du temps total d'interaction). Ce résultat indique que les résultats précédents sont généraux et ne dépendent pas de la difficulté des zones à peindre.

### 6.6.4 Synthèse

Si l'on considère le temps total d'interaction pour la tâche de peinture, on constate que les techniques tactiles sont les plus efficaces, ce qui est corroboré par les préférences des participants. Plus précisément T2M permet de minimiser le temps de changement de mode entre navigation et peinture par rapport à T1M, ce qui rend ses performances globales meilleures. Ceci est cohérent avec les notations des participants, qui placent T2M en meilleure position.

Le joystick 6dof est au même niveau que le stylet en terme de performances. Ceci est dû à la difficulté d'appréhender le joystick 6dof qui augmente le temps de navigation. Le temps de combinaison et la réduction du temps d'inaction ne parviennent pas à compenser l'accroissement du temps de navigation avec cette interaction.

Le temps de navigation au stylet est proche de celui des interactions tactiles, mais son temps d'inaction est plus élevé, ce qui dégrade ses performances globales. Ce temps d'inaction est représentatif de la difficulté de changement de mode : les participants ont en effet indiqué devoir systématiquement réfléchir au bouton à actionner pour effectuer une action donnée.



## Bilan

Dans ce chapitre, nous avons présenté une étude expérimentale qui nous permet de tirer trois conclusions principales concernant la peinture numérique sur modèle 3D.

Tout d'abord, l'usage de l'interaction tactile pour la navigation est particulièrement approprié dans le cadre d'utilisateurs novices. Elle est plus efficace que la navigation au stylet. Ceci étend à l'interaction 3D les résultats de l'état de l'art indiquant que dans le cadre de l'interaction 2D, le tactile est approprié à la gestion des commandes et à la navigation (cf. section 3.2.2.2).

Nous avons également montré qu'une répartition bimanuelle de l'interaction est plus efficace qu'une répartition unimanuelle modale de celle-ci, ce qui reste dans la logique des travaux précédents, que ce soit en interaction 2D ou 3D (cf. section 3.1.3).

Enfin, nous avons montré que la technique TAV+ développée au chapitre 5 est plus efficace que l'utilisation d'un dispositif d'entrée isométrique à 6 degrés de liberté.



# Bilan des contributions

Nous avons apporté dans cette partie trois contributions concernant la peinture numérique sur modèle 3D à destination des utilisateurs novices. Ces contributions s'organisent suivant trois axes principaux, à savoir la peinture numérique, la navigation 3D, et la combinaison de la peinture avec la navigation.

Concernant la peinture numérique, nous avons montré que l'interaction directe améliore les performances de dessin et conduit à des erreurs d'orientation plus faibles et une moindre variabilité comparée à l'interaction indirecte. Nous en avons déduit que l'usage d'écrans interactifs est plus adapté que les tablettes graphiques pour la peinture numérique auprès d'un public novice.

Nous avons ensuite apporté une contribution concernant la navigation 3D sur écran interactif en proposant des techniques de rotations 3D adaptées à un usage novice. Pour cela, nous avons introduit un nouveau critère pour les techniques d'interaction, la surjection, et nous avons élaboré nos techniques de manière à respecter ce critère.

Enfin, nous avons évalué différentes formes d'interaction de peinture numérique 3D combinant peinture et navigation. Nous avons montré que l'interaction bi-manuelle consistant à peindre avec la main dominante, et à naviguer avec la main non dominante via une interaction tactile basée sur une des nouvelles techniques de rotations que nous proposons, est la plus performante dans le cadre d'un usage novice.



# Conclusion

## Bilan général

Durant cette thèse, nous avons étudié le problème de la peinture numérique sur modèles 3D pour utilisateurs novices. Nous avons vu que les novices peuvent rencontrer des difficultés à deux niveaux : (1) au niveau des interactions nécessaires à chaque composante de la tâche : peinture numérique et navigation ; (2) au niveau de l'utilisation conjointe de ces interactions. Dans l'objectif de comprendre et réduire ces difficultés, nous avons suivi trois axes de recherche : la peinture numérique, la navigation 3D, et la combinaison des techniques de peinture et de navigation.

Concernant la peinture numérique, nous sommes partis du constat que l'interaction indirecte est souvent estimée plus difficile que l'interaction directe. Afin d'objectiver ce phénomène et d'en comprendre les raisons, nous avons effectué une étude expérimentale qui nous a permis de recommander l'usage des écrans interactifs pour les novices.

Concernant la navigation, nous avons vu dans l'état de l'art que les interactions de type *Rotate/Pan/Zoom* sont particulièrement adaptées à la tâche de peinture numérique 3D, mais que les interactions de contrôle de rotation sont difficiles à maîtriser. Nous avons donc proposé des techniques de rotations 3D adaptées à un usage novice (TAV+ et Arcball+), afin d'obtenir une interaction de type *Rotate/Pan/Zoom* accessible à cette catégorie d'utilisateurs.

Enfin, en nous appuyant sur l'état de l'art et les contributions précédentes, nous avons sélectionné un ensemble de dispositifs d'entrée et d'interactions susceptibles d'être bien adaptés à la peinture numérique 3D en usage récréatif. Nous avons évalué ces différentes interactions de manière à mettre en évidence la plus efficace. L'étude a révélé que la meilleure combinaison consiste à peindre au stylet avec la main dominante, et à contrôler le point de vue de la caméra par une interaction tactile exécutée par la main non-dominante. L'interaction de navigation est de type *Rotate/Pan/Zoom* avec un contrôle TAV+ pour les rotations.

L'objectif principal des travaux présentés dans ce mémoire est de simplifier l'interaction avec les scènes numériques 3D afin d'en démocratiser l'usage. Nous avons focalisé sur une tâche particulière : la peinture des modèles. Même si cette tâche ne représente qu'une petite partie des activités nécessaire à la création de scène 3D, nous pensons qu'elle peut être une *porte d'entrée* accessible et ludique, à l'instar de la peinture de modèles physiques. Si l'utilisateur novice est satisfait de son expérience de peinture de modèles, nous pensons qu'il sera davantage enclin à faire les efforts nécessaires pour apprendre les interactions plus complexes pour des tâches plus complexes. Nos travaux, vus en parallèle d'autres efforts qui visent à simplifier les tâches de modélisation [211–213], indiquent que la création numérique 3D accessible au plus grand nombre n'est plus une utopie.

Les résultats obtenus dans le cadre de ce mémoire ont fait l'objet des publications suivantes :

- Élisabeth Rousset, François Bérard, and Michaël Ortega. Two-finger 3d rotations for novice users: surjective and integral interactions. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, pages 217–224. ACM, 2014
- Élisabeth Rousset, François Bérard, and Michaël Ortega. Study of the effect of the directness of the interaction on novice users when drawing straight lines. In *27ème conférence francophone sur l'Interaction Homme-Machine.*, page a14. ACM, 2015

## Perspectives de recherche

Les travaux présentés dans ce mémoire ouvrent plusieurs pistes de recherches :

- Nous avons montré la supériorité d'une interaction directe, par rapport à une interaction indirecte, dans le cas du dessin de lignes droites par des utilisateurs novices. Mais dessin et peinture ne peuvent se réduire aux lignes droites. Il sera nécessaire de conduire d'autres expérimentations pour vérifier si ce résultat se généralise aux courbes. L'étude de Reina et al. concernant la coordination oeil-main dans ce cadre [34] donne bon espoir sur la généralité de ce résultat.
- En dépit du bénéfice sur la qualité du tracé, les écrans interactifs directs n'offrent pas le confort d'utilisation des tablettes tactiles. Nous envisageons d'étudier des

approches visant à conserver le meilleur des deux mondes. Par exemple, de nouvelles techniques d'interactions au stylet indirect pourraient permettre de diminuer l'erreur d'orientation et la variabilité du trait. Une piste de recherche serait d'afficher en temps réel une *prédiction* de la trajectoire.

- D'un point de vue plus général, nous avons focalisé nos travaux sur un contexte d'usage spécifique : les utilisateurs novices qui se servent du système dans un but récréatif. Mais il est possible que certains résultats produits dans ce mémoire se généralisent à d'autres contextes. En particulier, nous souhaitons étudier si le bénéfice des techniques d'interaction proposées ici se transpose à une utilisation quotidienne par des utilisateurs professionnels.





# Annexe A

## Revue technologique

### Sommaire

---

<b>Introduction</b> . . . . .	<b>138</b>
<b>A.1 Technologies tactiles</b> . . . . .	<b>138</b>
A.1.1 Technologies capacitives . . . . .	138
A.1.1.1 Capacité projetée . . . . .	139
A.1.1.2 Capacité de surface . . . . .	140
A.1.2 Technologies résistives . . . . .	141
A.1.2.1 Technologie résistive analogique . . . . .	141
A.1.2.2 Technologie résistive numérique . . . . .	142
A.1.3 Technologies acoustiques . . . . .	143
A.1.3.1 Ondes acoustiques de surface . . . . .	143
A.1.3.2 Ondes acoustiques de profondeur . . . . .	143
A.1.4 Technologies optiques . . . . .	144
A.1.4.1 Technologie infrarouge . . . . .	144
Infrarouge traditionnel . . . . .	144
Variantes . . . . .	145
A.1.4.2 Technologie à caméras . . . . .	146
A.1.4.3 Réflexion interne totale . . . . .	147
Frustrated Total Internal Reflection . . . . .	147
Planar Scatter Detection . . . . .	148
A.1.5 Technologies à détection de pression . . . . .	148
Force sensing . . . . .	148
Sensel . . . . .	148
A.1.6 Technologies In-Cell . . . . .	149
<b>A.2 Stylets numériques</b> . . . . .	<b>149</b>
A.2.1 Résonance électro-magnétique . . . . .	150
A.2.2 Technologies capacitives . . . . .	150

A.2.2.1	Technologie passive . . . . .	150
A.2.2.2	Technologie active . . . . .	151
A.2.2.3	Technologie semi-active . . . . .	152
A.2.3	Technologie résistive . . . . .	152
A.2.4	Technologies optiques . . . . .	152
A.2.4.1	Technologie à caméra . . . . .	152
A.2.4.2	Technologie à infrarouges . . . . .	152
Intupaint	. . . . .	152
I/O brush	. . . . .	153
A.2.5	Stylet magnétique . . . . .	154
<b>Bilan</b>	. . . . .	<b>155</b>

---

## Introduction

Nous allons effectuer dans cette annexe une revue des technologies tactiles et des technologies de stylets numériques existantes, en considérant leurs avantages et inconvénients du point de vue d'une tâche de peinture. Nous en tirerons une synthèse concernant les technologies les mieux adaptées à la tâche de peinture numérique.

### A.1 Technologies tactiles

Plusieurs technologies tactiles ont été développées parallèlement au cours des dernières décennies[216–218]. On peut regrouper ces technologies dans différentes catégories (capacitive, résistive, acoustique, optique, à pression) qui présentent chacune des avantages et des inconvénients.

#### A.1.1 Technologies capacitives

Le premier dispositif équipé de capteurs capacitifs tactile est un synthétiseur développé par Le Caine en 1955[219]. Le premier écran capacitif est mis au point dix ans plus tard par Johnson[220, 221]. Le premier écran capacitif multipoint est développé par Boie et al en 1984[222], et le premier touchpad en 1985 par Buxton et al[223]. Il existe deux types de technologies capacitives, à savoir la *capacité projetée* (*projected capacitance*) et la *capacité de surface* (*surface capacitance*). Nous présentons dans la suite le cas où le dispositif tactile est combiné avec un écran, bien qu'il puisse être utilisé indépendamment (touchpad).

### A.1.1.1 Capacité projetée

Comme on peut le voir à la figure A.1, l'écran LCD est placé derrière une matrice de lignes et colonnes en matériau conducteur transparent (Oxyde d'étain-indium) séparées par des couches isolantes (verre). Chaque intersection définit alors une électrode. Comme le corps humain est conducteur, un contact avec l'écran provoque une déformation du champ électrostatique de la matrice. Cette déformation peut être mesurée comme un changement de capacité des électrodes.

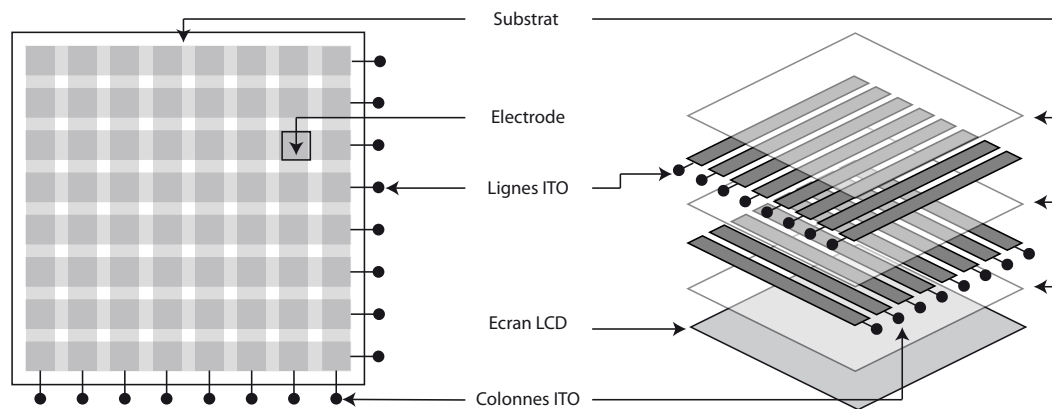


FIGURE A.1 – Capacité projetée

On parle *d'auto capacité* lorsque l'on mesure la capacité d'une unique électrode par rapport à la masse (cf Figure A.2a). Dans ce cas, lorsqu'un doigt s'approche de l'électrode, la capacité du corps humain augmente celle de cette dernière[224]. Cette technique permet en partie d'effectuer des entrées multipoint. Cependant, les points de contact séparés diagonalement provoquent l'apparition de contacts fantômes (cf Figure A.3). En effet, la détection s'effectuant par ligne et par colonne, il existe quatre emplacements possibles pour deux contacts simultanés.

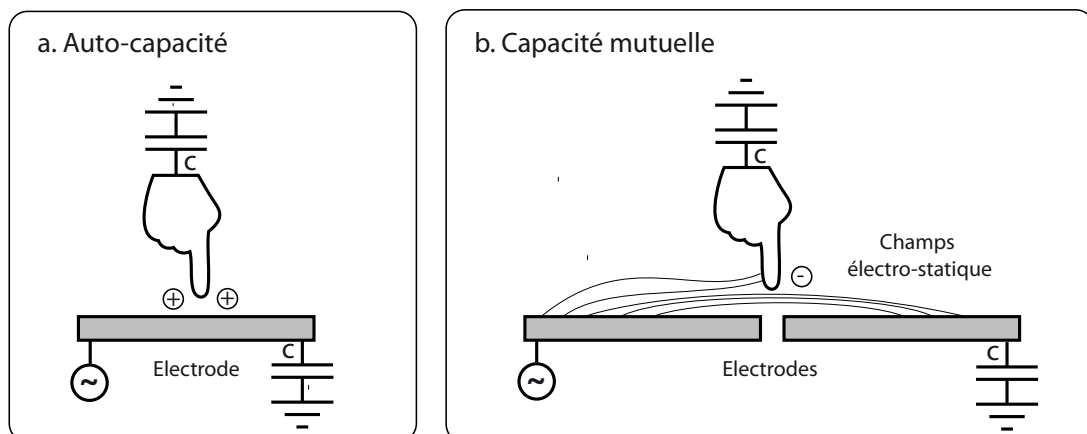


FIGURE A.2 – Auto-capacité et capacité mutuelle

Pour remédier à ce problème, la *capacitance mutuelle* mesure les changements de capacité entre les électrodes deux à deux (cf Figure A.2b). Dans ce cas, lorsqu'un doigt s'approche d'une paire d'électrodes, la capacité du corps humain par rapport à la masse prend une partie de la charge entre les électrodes, et donc diminue leur capacité[224]. Les différentes intersections sont alors mesurées indépendamment les unes des autres, ce qui évite les contacts fantômes.

La capacité projetée est plus durable, sensible et précise que les technologies résistives que l'on décrira plus loin[216]. Cependant, elle ne permet pas de détecter la pression, à moins d'être associée à un dispositif spécifique (cf. section A.2.2). Comme cette technologie est composée de plusieurs couches, son épaisseur peut amener à des problèmes de parallaxe.

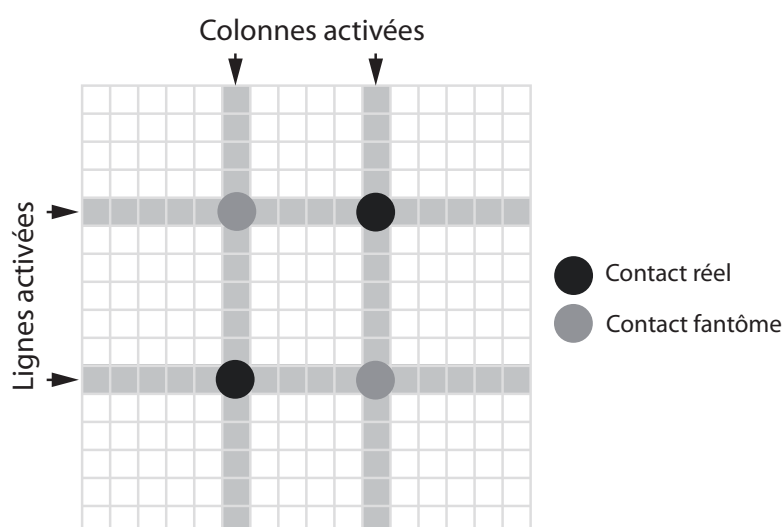


FIGURE A.3 – Contacts fantômes

#### A.1.1.2 Capacité de surface

Cette technologie a été brevetée par MicroTouch[225, 226]. Elle consiste en une plaque isolante (verre) associée à un conducteur transparent (Indium Tin Oxyde). Des électrodes sont placées sur le contour de la couche conductrice (figure A.4). Ces derniers permettent de mesurer les variations du champ électrostatique de l'écran lors d'un contact avec celui-ci.

Cette technologie est plus simple et moins couteuse à mettre en oeuvre que la capacité projetée. Cependant elle ne permet pas les entrées multipoints. De plus, elle nécessite une masse très stable, ce qui la rend inexploitable pour les terminaux mobiles[216]. Cette technologie est sujette à des problèmes de drift et présente une qualité visuelle réduite (elle ne restitue que 85% de la lumière de l'écran) ce qui la rend peu appropriée pour un usage artistique de précision [216].

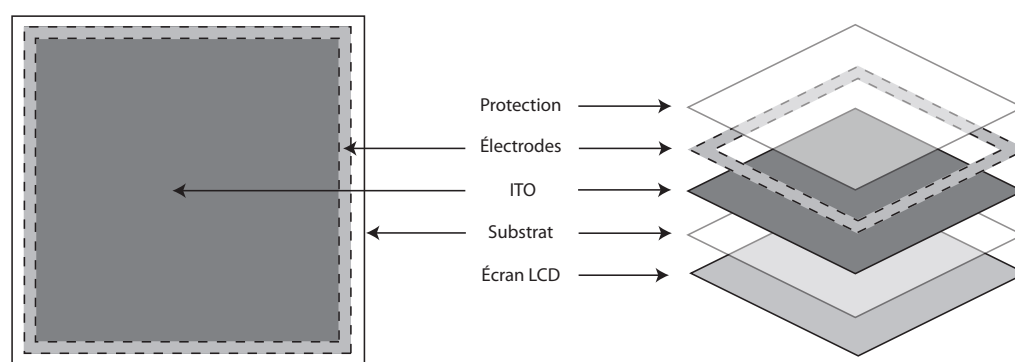


FIGURE A.4 – Capacité de surface

## A.1.2 Technologies résistives

### A.1.2.1 Technologie résistive analogique

Le premier écran résistif a été développé par Elographics en 1971[227], puis breveté en 1975 par son fondateur G. Hurst[228]. Cette technologie, appelée *écran résistif analogique*, est composée de deux couches conductrices transparentes séparées par un mince espace vide. Les couches conductrices sont parcourues par un courant électrique. La pression exercée sur l'écran crée un point de contact entre elles, ce qui modifie leur champ électrique (cf. figure A.5). Les coordonnées du point de contact sont déterminées grâce à la variation de ce champ électrique.

Cette technologie est sensible à l'usure (à cause de la variation de la conductivité des surfaces conductrices) et nécessite une re-calibration régulière. De plus, elle ne permet la détection que d'un seul point de contact et possède une mauvaise qualité optique (10% à 20% de réduction de la luminosité[217]). Elle est donc assez peu adaptée à un usage artistique. Cependant, cette technologie est relativement répandue car elle est peu chère et fonctionne avec n'importe quelle source de pression (doigt, gant, stylet, etc.), mais est utile essentiellement pour du pointage.

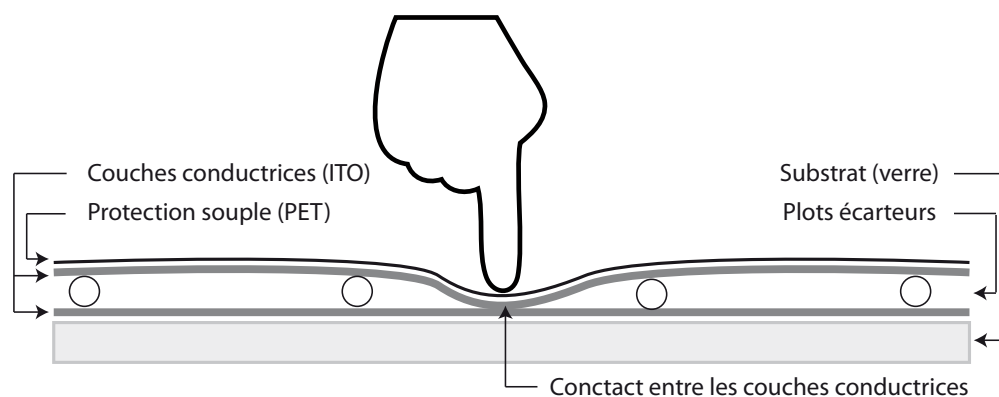


FIGURE A.5 – Écran résistif analogique

### A.1.2.2 Technologie résistive numérique

Une variante appelée *technologie résistive numérique (digital resistive)* permet de détecter plusieurs points de contact. Pour cela, l'écran est divisé en une matrice dont chaque case se comporte comme un écran résistif analogique indépendant (cf figure A.6). Cependant, cette technologie ne permet pas de détecter deux points de contacts dans la même case, c'est pourquoi la matrice doit être la plus fine possible. Cela a pour conséquence de rendre le procédé de fabrication aussi coûteux que celui des écrans capacitifs. De plus, cette technologie souffre des mêmes inconvénients que son pendant analogique concernant une utilisation artistique.

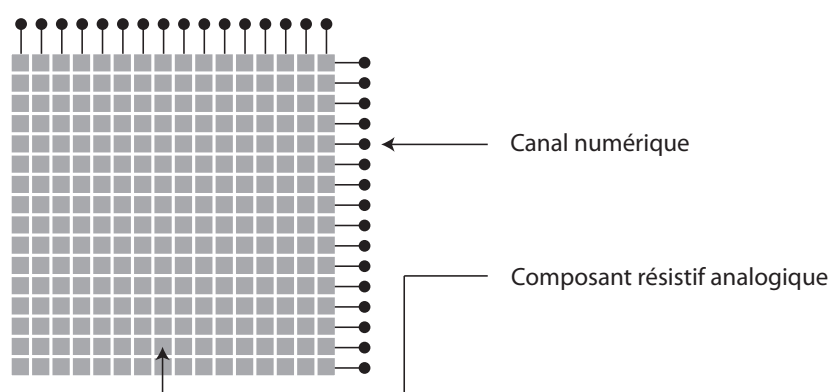


FIGURE A.6 – Écran résistif numérique

### A.1.3 Technologies acoustiques

#### A.1.3.1 Ondes acoustiques de surface

Il existe plusieurs technologies tactiles basées sur les ondes acoustiques. La technologie d'*onde acoustique de surface* ou *surface acoustic waves* a été inventée par R. Adler en 1985 [229]. Elle consiste en une plaque de verre, quatre transducteurs piézo-électriques et quatre réflecteurs organisés suivant le schéma de la figure A.7. Les ondes ultrasoniques se déplacent à la surface du verre depuis un transducteur émetteur, sont réfléchies par les réflecteurs puis récupérées par le transducteur récepteur. Lorsqu'un doigt touche l'écran, il absorbe une partie des ondes. La position du point de contact est déterminée en combinant le temps de trajet et l'amplitude des ondes reçues. Cette technologie permet une bonne qualité optique puisqu'elle ne nécessite qu'une seule couche. De plus, elle peut être utilisée avec n'importe quel objet absorbant le son, comme des gants ou un stylet souple. Par contre elle est fragile car la moindre rayure ou particule à la surface du verre peut altérer la précision des résultats, ce qui peut la rendre délicate pour un usage grand public.

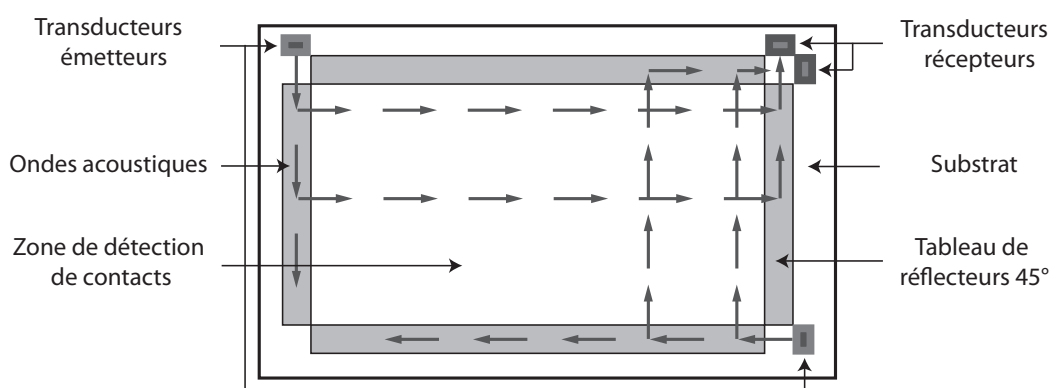


FIGURE A.7 – Écran à ondes acoustiques de surface

#### A.1.3.2 Ondes acoustiques de profondeur

Dans le cas des technologies d'*ondes acoustiques de profondeur* ou *bending waves*, les ondes acoustiques se propagent à l'intérieur du substrat. Lorsqu'un objet touche le verre, le contact génère des ondes qui se dispersent dans celui-ci à des vitesses différentes selon leur longueur d'onde (figure A.8). Le résultat de cette dispersion est récupéré par des capteurs le long des bords de l'écran. Il y a ensuite deux types de technologies propriétaires qui traitent ce résultat.

La technique *Acoustic Pulse Recognition* de l'entreprise Elo TouchSystem consiste à caractériser en avance le substrat en frappant ce dernier à plusieurs milliers d'endroits et en enregistrant le résultat obtenu[230]. Par la suite, lorsqu'un contact est détecté, sa signature est comparée à la base d'échantillons pour déterminer sa position. L'autre technique, appartenant à l'entreprise 3M, se nomme *Dispersive Signal Technology*[231]. Au lieu d'utiliser une banque d'échantillons, cette technique analyse la propagation des ondes en temps réel pour déterminer l'origine de celles-ci.

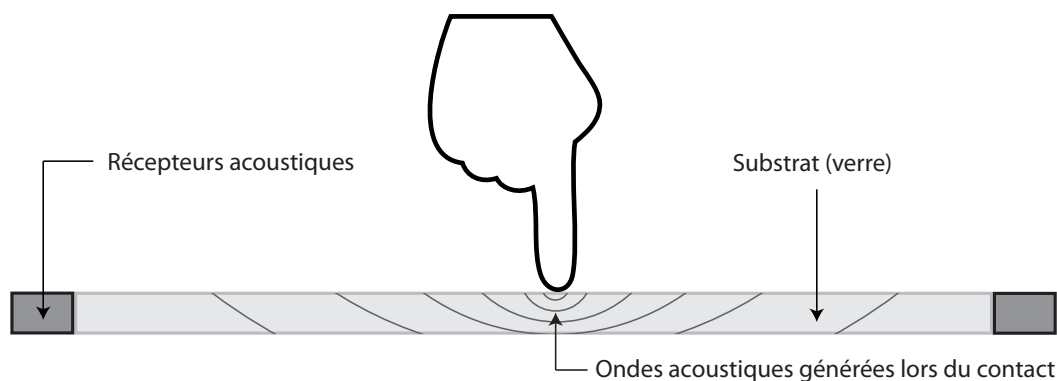


FIGURE A.8 – Ondes acoustiques de profondeur

Ces technologies permettent de détecter le contact de n'importe quel type d'objet. Cependant elles ne permettent pas les entrées multipoint. Elles nécessitent une certaine vitesse du contact pour générer les ondes, ainsi qu'un calibrage conséquent pour APR. De plus elles ne détectent pas les événements de type *touch and hold*, ce qui les rend inutilisables pour une tâche de tracé.

## A.1.4 Technologies optiques

### A.1.4.1 Technologie infrarouge

**Infrarouge traditionnel** Le premier écran infrarouge a été breveté en 1973 par Ebeling et al. [232]. Cette technologie consiste à ajouter au dessus de l'écran un cadre comprenant un ensemble de diodes infrarouges sur deux cotés adjacents et un ensemble de récepteurs infrarouges sur les deux autres cotés (figure A.9). Les diodes composent alors une grille de lumière infrarouge. Un contact est détecté lorsqu'un objet opaque à ce type de lumière bloque des rayons.

Cette technologie permet les entrées multipoints de manière assez limitée car elle est sujette aux contacts fantômes, tout comme l'auto-capacité qui est elle aussi basée sur un modèle de détection par lignes et colonnes (cf. Figure A.3). De part sa construction, elle



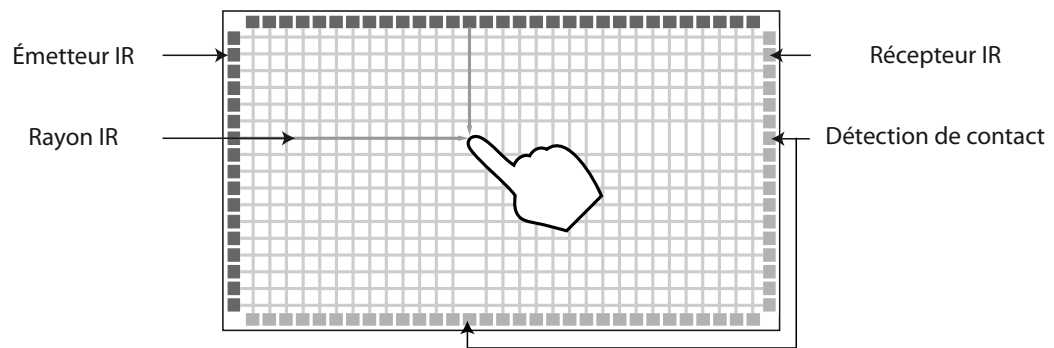


FIGURE A.9 – Écrans infrarouges

nécessite une épaisseur supplémentaire non-négligeable pour inclure le cadre et les diodes. La résolution est déterminée par le nombre de diodes et reste relativement faible, ce qui est peu utilisable pour un usage de tracé précis. De plus, elle détecte les "contacts" avant que ceux-ci aient réellement lieu, ce qui pose le problème de détection de faux contact pendant le survol.

**Variantes** L'entreprise RPO a développé une variante de cette technologie intitulée *Digital Waveguide Touch*. Cette dernière utilise uniquement deux diodes infrarouges (une pour chaque axe) dont la lumière est distribuée sous forme de grille par des canaux de polymère (cf figure A.10a). Cette technique permet d'atteindre des résolutions plus élevées que les écrans infrarouges classiques, mais elle nécessite l'emploi d'un encadrement supplémentaire pour protéger les canaux de polymère.

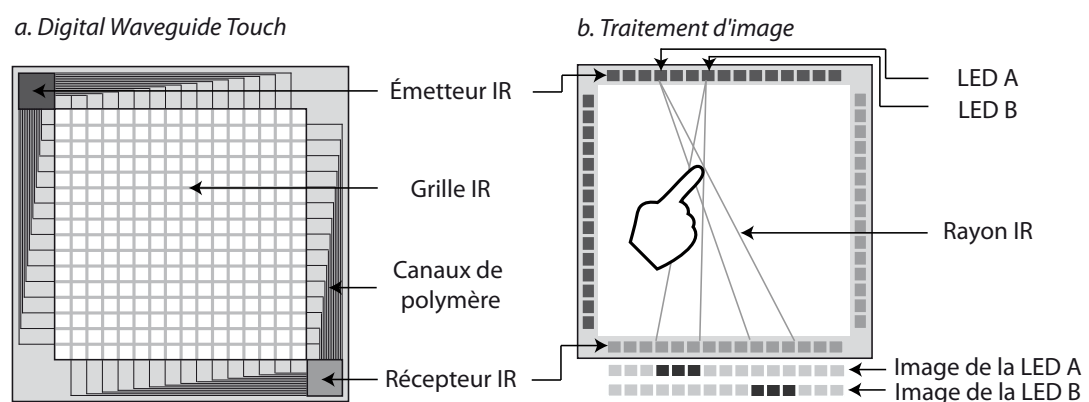


FIGURE A.10 – Variantes infrarouges

Il existe également une variante multipoints des écrans infrarouges basée sur une approche de traitement d'image [218] : chaque diode émet successivement un flash qui est

enregistré par plusieurs récepteurs sur le coté opposé. La combinaison des niveaux d'intensité lumineuse des différents couples flash/récepteur fournit une image qui permet de localiser les points de contacts (figure A.10b).

#### A.1.4.2 Technologie à caméras

La détection de points de contact à l'aide de caméras est une technologie brevetée en 1979 par l'entreprise Sperry Rand corp[233]. Elle consiste éclairer l'écran avec une lumière infrarouge depuis les coins de celui-ci. La lumière est réfléchi au niveau des bords de l'écran par des rétro-rélecteurs (un matériau qui renvoie la lumière dans la direction d'où elle vient quel que soit son angle d'incidence) puis captée par des caméras positionnée au niveau des émetteurs. Le contact d'un objet sur l'écran génère des ombres sur les caméras qui permettent d'obtenir les coordonnées du point de contact par triangulation (figure A.11).

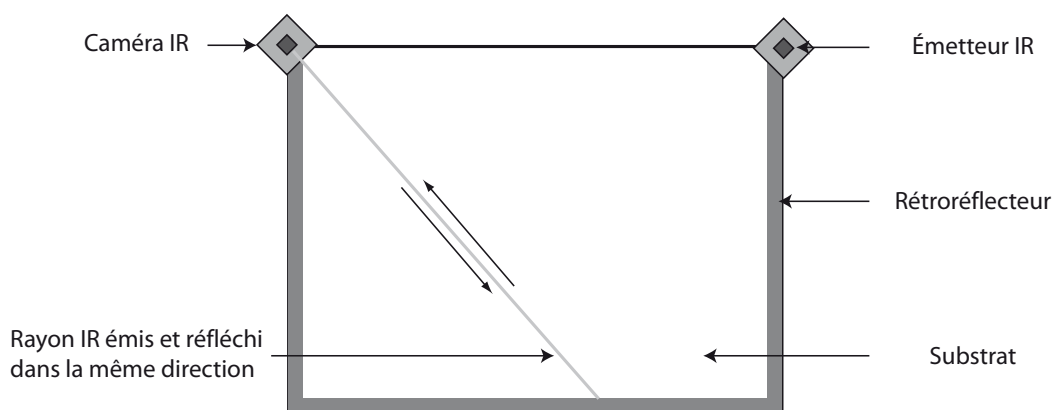


FIGURE A.11 – Écrans à caméras

Cette technologie gère assez mal les entrées multipoint car elle est sujette aux contacts fantômes (figure A.12). Cependant ces derniers peuvent être réduits à l'aide d'algorithmes complexes[234].

Cette technologie permet de détecter tout type de contact produit par un objet opaque au rayonnement infrarouge (doigts, gants, stylet, etc.) et de permettre également la reconnaissance d'objet via la forme des ombres. Cependant elle est sensible à la lumière ambiante ainsi qu'aux débris et irrégularités de l'écran. Elle est également sujette aux faux-contacts lors du survol et est donc peu appropriée pour un usage de tracé de précision.

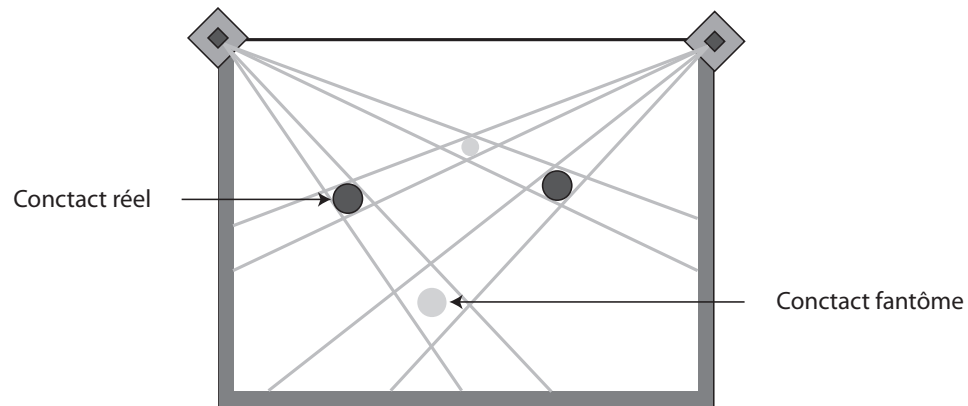


FIGURE A.12 – Contacts fantomes sur les écrans à caméras

#### A.1.4.3 Réflexion interne totale

**Frustrated Total Internal Reflection** Une première technologie utilisant la réflexion interne pour détecter des points de contact est brevetée en 1984[235]. Cependant sa forme actuelle est présentée par Han en 2005[236]. Comme indiqué à la figure A.13, une lumière infrarouge est injectée dans substrat placé au-dessus de l'écran depuis les bords de celui-ci. La lumière reste prisonnière du substrat à cause de l'angle d'incidence de départ, à moins qu'un objet entre en contact avec celui-ci. La lumière est alors diffusée vers l'intérieur, phénomène appelé *frustrated total internal reflection*. Cette lumière est détectée par une caméra placée derrière l'écran, ce qui permet de localiser le point de contact.

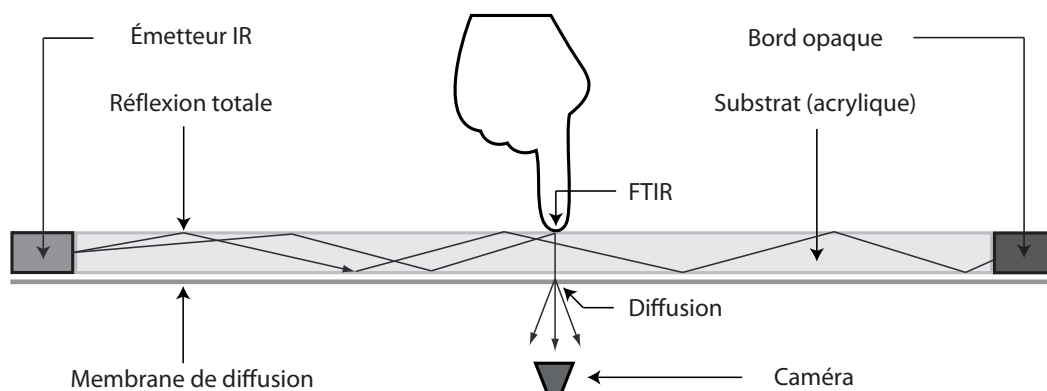


FIGURE A.13 – Réflexion interne totale

Cette technologie permet de détecter un nombre illimité de points de contacts simultanément et de reconnaître différents types d'objets. Elle est suffisamment précise pour un usage artistique et fait l'objet d'adaptations pour l'interaction au stylet (cf. section A.2.4.2).

**Planar Scatter Detection** Une variante de cette technologie a été brevetée par l'entreprise Flatfrog sous le nom de *Planar Scatter Detection*[237, 238]. Cette fois-ci ce n'est pas la lumière diffusée qui est mesurée, mais la diminution d'intensité de la lumière à l'intérieur du substrat (diminution due à la diffusion de lumière au niveau du point de contact) comme on peut le voir à la figure A.14 .

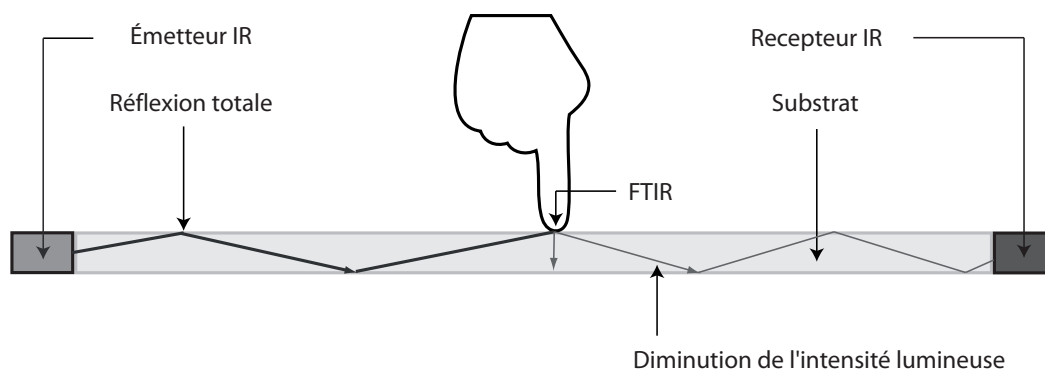


FIGURE A.14 – Planar Scatter Detection

### A.1.5 Technologies à détection de pression

**Force sensing** Le premier écran à détection de pression ou *force sensing* a été développé par IBM en 1971 sous le nom de *TouchSelect*[218]. L'objectif de cette technologie est de détecter directement la pression physique appliquée à l'écran. Pour cela l'écran est soutenu par des capteurs de pression comme indiqué à la figure A.15. Chaque contact induit une pression sur l'écran qui est détectée par les capteurs. Cette technologie a l'avantage de ne pas nécessiter de couche supplémentaire entre l'utilisateur et l'écran, et de détecter n'importe quel type d'objet. Cependant elle ne permet pas les entrées multipoints, et nécessite de dépasser un certain seuil de pression pour que le contact soit détecté. Elle est donc peu appropriée à un usage de peinture.

**Sensel** Rosenberg et al. ont développé une version plus évoluée de cette technologie intitulée *Sensel*[239, 240] : ils proposent d'équiper une surface d'une grille de micro-capteurs de pression (figure A.16). La résolution est de 20000 capteurs pour une surface 23cm\*13cm. Cette technologie est suffisamment précise pour détecter un nombre quelconque de points de contacts. Elle permet de mesurer la pression appliquée et la forme des points de contacts. Elle est donc particulièrement adaptée à un usage artistique. Cette technologie a été brevetée en 2011 et fait actuellement l'objet d'une levée de fonds afin d'être commercialisée sous forme d'une surface indépendante de l'écran[241, 242].

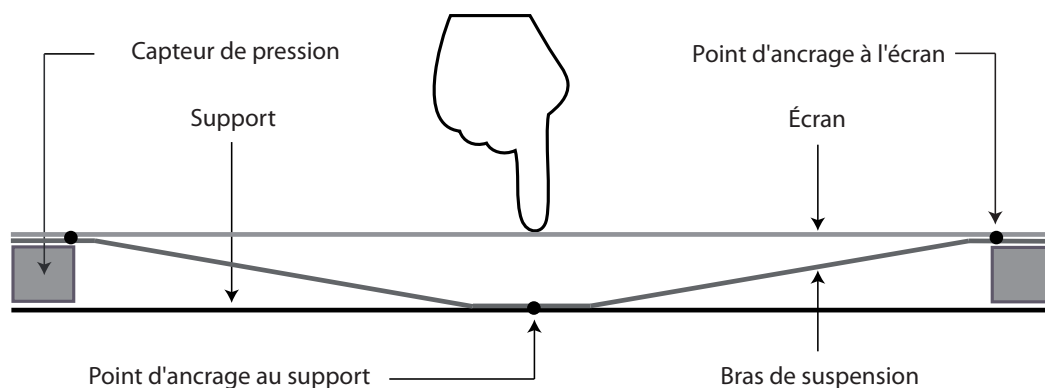


FIGURE A.15 – Force Sensing

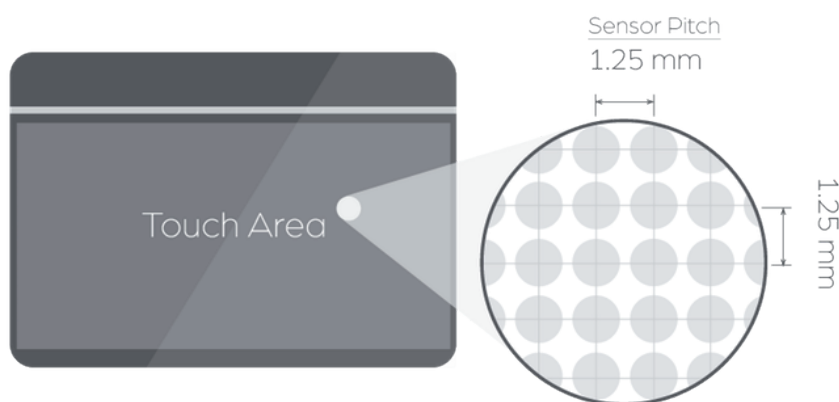


FIGURE A.16 – Sensel Morph

### A.1.6 Technologies In-Cell

Certaines technologies de détection de points de contact que nous venons de passer en revue se déclinent aussi en versions *In Cell* : les capteurs sont miniaturisés et directement intégrés aux cellules de l'écran LCD. C'est le cas des technologies *Light-sensing in cell*, *Voltage-sensing in cell* et *Capacitive-sensing in cell*[216].

## A.2 Stylets numériques

Il existe plusieurs types de technologies de stylets numériques. Certaines, comme les stylets capacitifs, font usage des technologies existantes en terme d'écrans tactiles. D'autres, comme les stylets à résonance électro-magnétique, sont complètement originales.

## A.2.1 Résonance électro-magnétique

Cette technologie fonctionne selon un principe d'absorption/émission d'énergie électro-magnétique (ondes radio basses fréquences). Une grille formée de boucles de cuivre est disposée derrière l'écran LCD (figure A.17). Elle communique par ondes radio avec un stylet comprenant un couple induction/capacité de la manière suivante : (1) la grille émet une quantité d'énergie électro-magnétique à destination du stylet puis passe en mode récepteur (2) le stylet absorbe l'énergie, puis la renvoie en y encodant des informations sur son état (boutons, pression, etc) (3) la grille réabsorbe l'énergie émise par le stylet et permet de localiser l'origine de celle-ci[216].

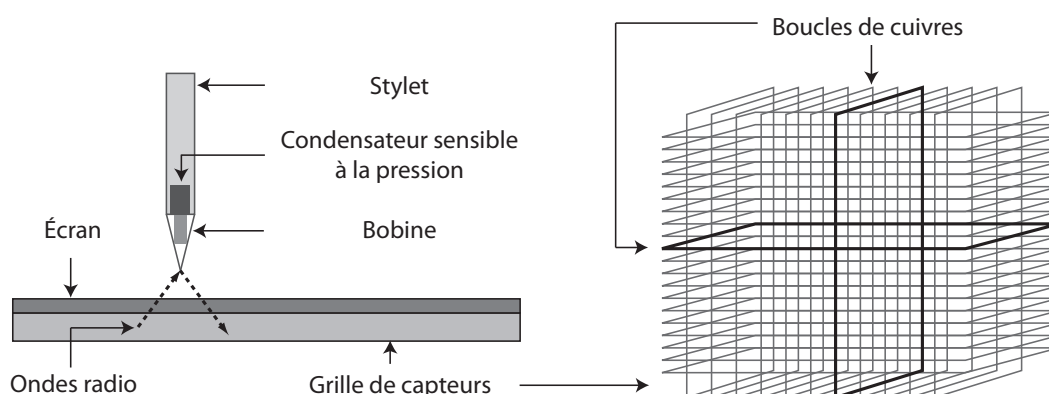


FIGURE A.17 – Résonance électro-magnétique

Cette technologie est principalement développée et brevetée par l'industriel Wacom[243–247]. Elle est très précise (1000dpi et plus) et permet de détecter le stylet à distance de l'écran (survol). Elle fournit des informations sur la pression appliquée au stylet. De plus elle n'influe pas sur la qualité visuelle de l'écran car la grille est placée derrière celui-ci. Elle est également insensible aux contacts de la paume de la main (*palm rejection*). Cependant, cette technologie est assez coûteuse.

## A.2.2 Technologies capacitives

### A.2.2.1 Technologie passive

Un stylet conçu en matériau conducteur peut être utilisé avec n'importe quel écran capacitif. En effet, il transmet les charges électriques du corps humain et remplace le doigt de l'utilisateur. Certains écrans capacitifs peuvent fonctionner avec des stylets de seulement 2mm d'épaisseur, grâce à des algorithmes de réduction de bruit[217]. Cette technologie est peu coûteuse et ne nécessite pas d'embarquer de matériel électronique

ni de batterie dans le stylet. De plus elle fonctionne avec tous les écrans capacitifs. Cependant elle est assez peu précise et ne permet pas de détecter le survol de l'écran.

### A.2.2.2 Technologie active

Dans cette technologie le stylet est alimenté par une batterie. Il génère des charges au niveau de la pointe, qui sont captées par une grille capacitive standard (figure A.18a). Celle-ci permet d'obtenir des informations sur la localisation du stylet. Cela permet au stylet d'être plus fin qu'un stylet passif. De plus, le stylet transmet à la grille capacitive des informations supplémentaires (pression, id, etc) via des ondes radio basse fréquence. Un contrôleur spécifique est associé à la grille capacitive pour décoder ces informations [217].

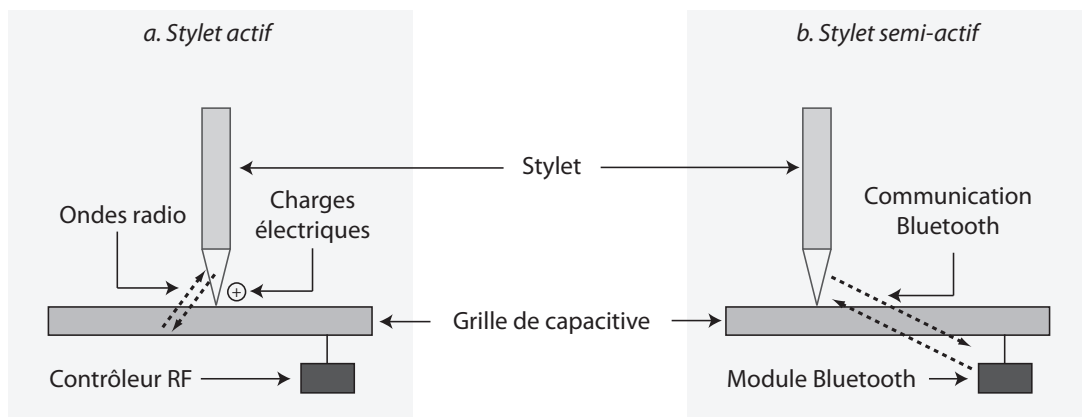


FIGURE A.18 – Stylets capacitifs

Il existe deux variantes concernant la transmission par onde radio. NTrig propose une communication à sens unique, du stylet vers la grille [217, 248–250]. Atmel propose un mode de communication où le stylet reçoit un signal émis par le contrôleur, l'amplifie et y encode des informations supplémentaires puis le retransmet à la grille capacitive [217, 251, 252].

Cette technologie est assez précise. Elle permet de détecter le survol du stylet. Elle ne nécessite pas l'inclusion de matériel électronique dans l'écran car elle utilise la grille capacitive déjà présente. Cependant elle implique l'utilisation d'une batterie dans le stylet. De plus les différentes marques de stylets actifs et de contrôleurs ne sont pas compatibles entre elles.

### A.2.2.3 Technologie semi-active

A mi-chemin entre les deux technologies précédentes, on trouve les stylets semi-actifs. Cette technologie consiste à munir un stylet capacitif passif d'un émetteur bluetooth (figure A.18b). Il permet d'appairer le stylet avec un appareil tactile disposant de la fonction bluetooth. Le stylet peut alors transmettre des informations supplémentaires à l'appareil via ce canal (la pression notamment). Cette technologie est principalement développée pour les terminaux mobiles, car ces derniers ont généralement un écran capacitif et un émetteur/récepteur bluetooth.

Cette technologie est en théorie utilisable avec n'importe quel dispositif respectant les conditions ci-dessus. En pratique, elle nécessite un support logiciel afin de décoder les informations transmises via bluetooth et de calculer le mécanisme de *palm rejection*. Cependant elle nécessite l'intégration d'une batterie dans le stylet et est plus coûteuse qu'un stylet passif. Elle est également moins précise qu'un stylet actif.

### A.2.3 Technologie résistive

Comme on l'a vu précédemment, la technologie tactile résistive permet également la détection d'un stylet de matériau quelconque. Cependant ses mauvaises qualités optiques la rendent peu utilisable dans un cadre artistique. De plus, cette technologie ne détecte pas le survol du stylet.

### A.2.4 Technologies optiques

#### A.2.4.1 Technologie à caméra

Les stylets optiques à caméra sont principalement développés et brevetés par l'entreprise Anoto[253–258]. Cette technologie consiste à insérer sur l'écran un motif irrégulier constitué de points et lignes microscopiques (Figure A.19). Ce motif est invisible à l'oeil nu car il est imprimé avec une encre carbone visible uniquement à la lumière infrarouge. Le motif est éclairé et filmé par une caméra infrarouge intégrée au stylet lorsque celui-ci est utilisé. Cela permet au stylet de calculer sa position par rapport à l'écran, puis de la transmettre par bluetooth.

#### A.2.4.2 Technologie à infrarouges

**Intupaint** Vandoren et al. ont développé une technologie de pinceau infrarouge dédiée à la peinture numérique[259, 260]. Cette technologie, intitulée *IntuPaint*, comprend une



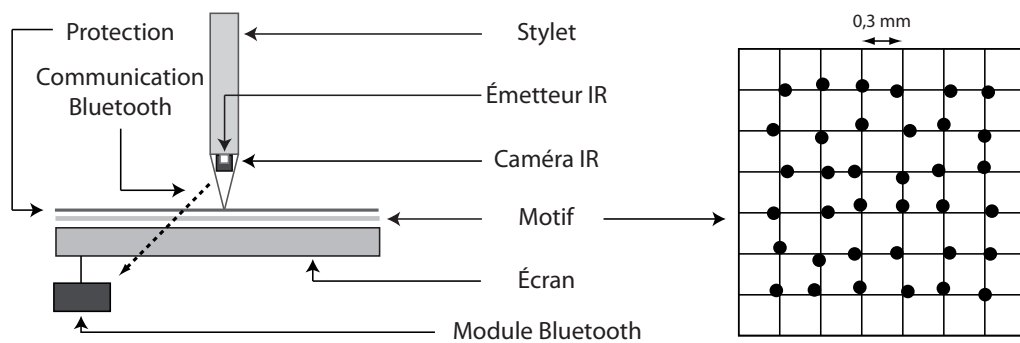


FIGURE A.19 – Stylet optique

surface de détection dont le fonctionnement est basé sur la technologie FTIR (cf. section A.1.4.3) ainsi qu'un pinceau de fibre optique. Les poils en fibre optique conduisent chacun un faisceau lumineux infrarouge qui est détecté par la surface (figure A.20). La forme de l'empreinte permet de distinguer les contacts du pinceau des contacts tactiles et de déterminer la pression et l'inclinaison utilisés.

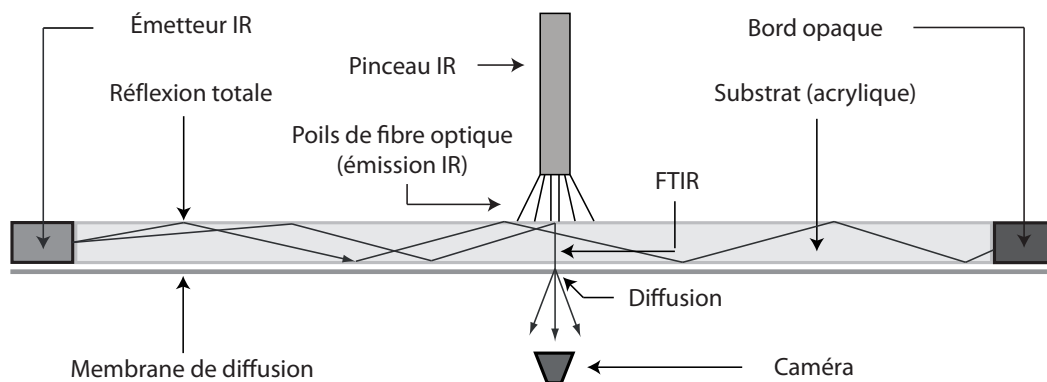


FIGURE A.20 – Intupaint[259, 260]

**I/O brush** Ryokai et al. ont mis au point un dispositif de peinture numérique adapté aux enfants en bas âge intitulé *I/O brush*[261, 262]. Cette technologie permet de définir un encre numérique à partir d'objets de l'environnement (figure A.21). Pour cela, le pinceau est équipé d'une caméra et de lampes LED afin de scanner l'objet souhaité. Ensuite l'enfant peut peindre avec l'encre obtenu sur un écran à technologie infrarouge qui détecte le contact du pinceau. Les poils du pinceau sont équipé de capteurs de pression et le corps du pinceau comprend un inclinomètre.

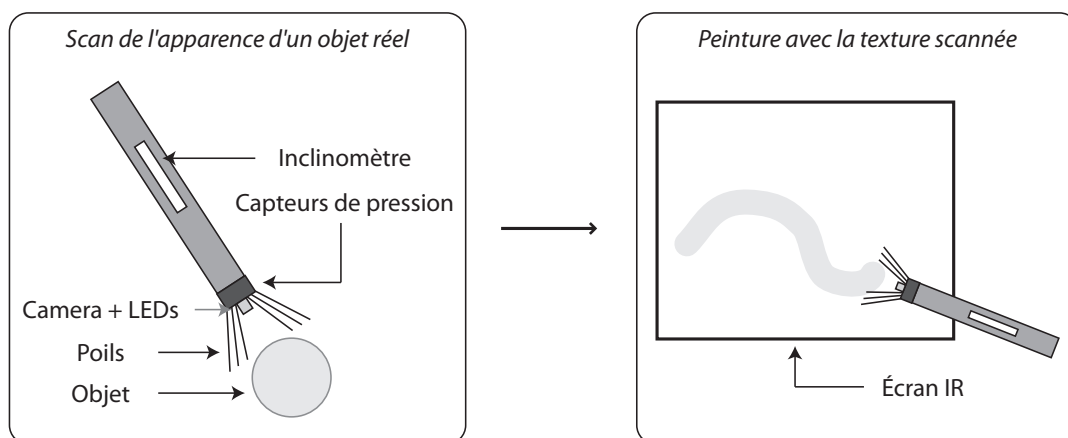


FIGURE A.21 – I/O brush [262]

### A.2.5 Stylet magnétique

La start-up grenobloise ISketchNote a développé et breveté une technologie magnétique de détection de stylet[263–269]. Cette technique combine un stylet équipé à l'aide d'une bague aimantée avec une surface comprenant un réseau de magnétomètre. Comme indiqué à la figure A.22, les magnétomètres permettent de détecter la position et l'orientation de la bague aimantée et donc du stylet. Cette technologie est actuellement commercialisée comme une surface indépendante de l'écran. Elle ne permet donc qu'une interaction de type indirect.

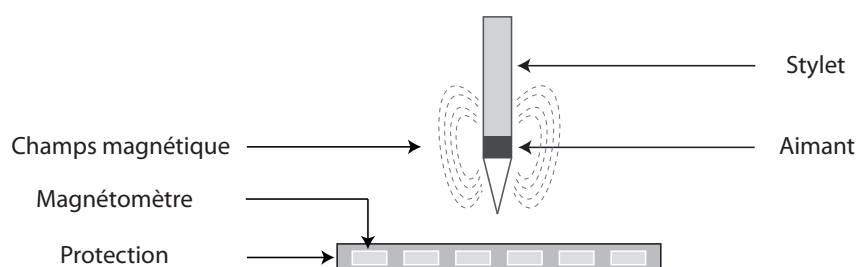


FIGURE A.22 – Stylet magnétique

Une technologie similaire a été brevetée par I. Rosenberg, le fondateur de Sensel[270] (cf. section A.1.5).

## Bilan

Les avantages et inconvénients des différentes technologies de stylet sont synthétisées dans la table A.1. Comme on peut le voir, les technologies les plus avantageuses sont la résonance électromagnétique et la réflexion interne totale. En effet, elles sont précises, permettent une bonne qualité optique, et fournissent des informations sur la pression et l'inclinaison du stylet. La technologie FTIR permet également d'utiliser un pinceau numérique et de détecter simultanément les contacts tactiles sans couche supplémentaire sur l'écran, ce qui évite d'aggraver les problèmes de parallaxe. À l'inverse, l'EMR permet uniquement l'utilisation d'un stylet et ne détecte pas les contacts tactiles. Cependant elle fournit des informations concernant le survol du stylet, ce qui rend cette technologie utilisable aussi bien en direct qu'en indirect.

L'EMR peut être combinée avec une couche de technologie tactile pour détecter les contacts des doigts. Il s'agit usuellement de la capacité mutuelle. En effet, cette technologie permet un bon rendu optique, est suffisamment précise pour une interaction tactile et permet la détection multipoint. De plus, elle ne provoque pas de faux contacts ou de contacts fantômes. Enfin, la gestion séparée des entrées tactiles et stylet permet d'implémenter la palm rejection de manière logicielle.

Il apparaît donc que le meilleur compromis correspond à la combinaison EMR/capacité mutuelle. Les expériences effectuées dans cette thèse ont donc été menées à l'aide de ces technologies, via un écran interactif Wacom 24HD Touch.

Technologie	Précision	Pression	Inclinaison	Survol	Indirect	Direct	Qualité optique	Parallaxe	Tactile simultané	Palm rejection	Solidité	Absence de faux contacts
EMR	++++	✓	✓	✓	✓	✓	+++	+++	x		+++	✓
Cap. passive	+	x	x	x	x	✓	++	++	✓	x	++	✓
Cap. active	+++	✓	x	✓	✓	✓	++	++	✓	✓	++	✓
Cap. Semi-active	++	✓	x	x	x	✓	++	++	✓	✓	++	✓
Résistive	+	x	x	x	x	✓	+	+	✓	x	+	✓
Caméra	+++	✓	x	✓	✓	✓	++	++	x		++	✓
FTIR	+++	✓	✓	x	x	✓	++	++	✓	✓	++	✓
Magnétique	++	x	✓	✓	✓	x			x		+++	x
Capteurs de pression	+++	✓	✓	x	✓	x			✓	✓	+++	✓
Infrarouge	+	✓	x	x	x	✓	++	++	x	✓	+	x

TABLE A.1 – Synthèse comparative des technologies tactiles et de stylets numériques

# Bibliographie

- [1] BlenderArtists. Blenderartists, 2015. URL <http://blenderartists.org/>.
- [2] CGSociety. Cgsociety, 2015. URL <http://www.cgsociety.org/>.
- [3] DeviantArt. Deviantart, 2015. URL <http://www.deviantart.com/>.
- [4] SketchFab. Sketchfab, 2015. URL <http://sketchfab.com/>.
- [5] Autodesk. 123d suite, 2015. URL <http://www.123dapp.com/>.
- [6] Pixologic. Sculptris, 2015. URL <http://pixologic.com/sculptris/>.
- [7] SketchUp. Sketchup, 2015. URL <http://www.sketchup.com/>.
- [8] Blender. Blender, 2015. URL <http://www.blender.org/>.
- [9] Universalis. Art numérique, 2015. URL <http://www.universalis.fr/encyclopedie/art-numerique/>.
- [10] Wikipédia. Digital art, 2015. URL [https://en.wikipedia.org/wiki/Digital\\_art](https://en.wikipedia.org/wiki/Digital_art).
- [11] DeviantArt. Digital art, 2015. URL <http://www.deviantart.com/browse/all/digitalart/>.
- [12] Alvy Ray Smith. Digital paint systems : An anecdotal and historical overview. *IEEE Annals of the History of Computing*, (2) :4–30, 2001.
- [13] Katiedesousa. Katiedesousa’s brush set, 2009. URL <http://www.deviantart.com/art/23-Brushes-for-Photoshop-117421513>.
- [14] William John Kubitz. *A tricolor cartograph*. PhD thesis, University of Illinois at Urbana-Champaign, 1968.
- [15] Richard G Shoup. Some experiments in television graphics and animation using a digital image memory. In *SMPTE Conferences*, volume 1979, pages 88–98. Society of Motion Picture and Television Engineers, 1979.

- [16] Alvy Ray Smith. Alpha and the history of digital compositing. URL : [http://www.alvyray.com/Memos/7\\_alpha.pdf](http://www.alvyray.com/Memos/7_alpha.pdf), zuletzt abgerufen am, 24 :2010, 1995.
- [17] Thomas Porter. Picture coding and the paint system. *Lucasfilm memo, San Rafael, Calif*, 13, 1981.
- [18] Thomas Porter. Picture handling using staging areas. *Lucasfilm memo, San Rafael, Calif*, 17, 1981.
- [19] Thomas Porter. The paint system design. Technical report, 1st Pass Technical Memo, 1981.
- [20] Ravin Balakrishnan and I Scott MacKenzie. Performance differences in the fingers, wrist, and forearm in computer input control. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 303–310. ACM, 1997.
- [21] François Bérard and Amélie Rochet-Capellan. The transfer of learning as hci similarity : Towards an objective assessment of the sensory-motor basis of naturalness. In *Proceedings of ACM CHI, CHI'15*, pages 1315–1324. ACM, 2015. ISBN 978-1-4503-3145-6. doi: 10.1145/2702123.2702359. URL <http://doi.acm.org.gate6.inist.fr/10.1145/2702123.2702359>.
- [22] Jonathan Deber, Ricardo Jota, Clifton Forlines, and Daniel Wigdor. How much faster is fast enough? : User perception of latency & latency improvements in direct and indirect touch. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI'15*, pages 1827–1836. ACM, 2015.
- [23] CGSociety. Intuos vs. cintiq, 2015. URL <http://forums.cgsociety.org/archive/index.php?t-1140969.html>.
- [24] DeviantArt. Intuos vs. cintiq, 2015. URL <http://forum.deviantart.com/art/digital/1884749/>.
- [25] Wacom. Eye-hand coordination exercises, 2015. URL [http://www.wacom.eu/\\_bib\\_user/dealer/man\\_onebywacom\\_en.pdf](http://www.wacom.eu/_bib_user/dealer/man_onebywacom_en.pdf).
- [26] Dominik Schmidt, Florian Block, and Hans Gellersen. A comparison of direct and indirect multi-touch input for large surfaces. INTERACT'09, pages 582–594. Springer, 2009. ISBN 978-3-642-03654-5. doi: 10.1007/978-3-642-03655-2\_65. URL [http://dx.doi.org/10.1007/978-3-642-03655-2\\_65](http://dx.doi.org/10.1007/978-3-642-03655-2_65).
- [27] Knoedel, Hachet, and Martin. Multi-touch rst in 2d and 3d spaces : Studying the impact of directness on user performance. In *3D User Interfaces (3DUI), 2011 IEEE Symposium on, 3DUI'11*, pages 75–78. IEE, March 2011. doi: 10.1109/3DUI.2011.5759220.

- [28] Clifton Forlines and Ravin Balakrishnan. Evaluating tactile feedback and direct vs. indirect stylus input in pointing and crossing selection tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'08*, pages 1563–1572, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-011-1. doi: 10.1145/1357054.1357299. URL <http://doi.acm.org/10.1145/1357054.1357299>.
- [29] Jérémie Gilliot, Géry Casiez, and Nicolas Roussel. Impact of form factors and input conditions on absolute indirect-touch pointing tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'14*, pages 723–732, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2473-1. doi: 10.1145/2556288.2556997. URL <http://doi.acm.org/10.1145/2556288.2556997>.
- [30] J L Vercher, G M Gauthier, J Cole, and J Blouin. Role of arm proprioception in calibrating the arm-eye temporal coordination. *Neuroscience letters*, 237(2-3) : 109–12, November 1997. ISSN 0304-3940. URL <http://www.ncbi.nlm.nih.gov/pubmed/9453227>.
- [31] C.J. Ketcham, Natalia V. Dounskaia, and George E. Stelmach. The role of vision in the control of continuous multijoint movements. *Journal of Motor Behavior*, 38(1) :29–44, 2006. doi: 10.3200/JMBR.38.1.29-44. URL <http://dx.doi.org/10.3200/JMBR.38.1.29-44>.
- [32] Emma Gowen and R Chris Miall. Eye-hand interactions in tracing and drawing tasks. *Human movement science*, 25(4-5) :568–85, October 2006. ISSN 0167-9457. doi: 10.1016/j.humov.2006.06.005. URL <http://www.ncbi.nlm.nih.gov/pubmed/16891021>.
- [33] John Tchalenko. Eye movements in drawing simple lines. *Perception*, 36(8) : 1152–1167, 2007. ISSN 0301-0066. doi: 10.1068/p5544. URL <http://www.perceptionweb.com/abstract.cgi?id=p5544>.
- [34] G. Anthony Reina and Andrew B. Schwartz. Eye–hand coupling during closed-loop drawing : Evidence of shared motor planning? *Human Movement Science*, 22(2) : 137–152, April 2003. ISSN 01679457. doi: 10.1016/S0167-9457(02)00156-2. URL <http://linkinghub.elsevier.com/retrieve/pii/S0167945702001562>.
- [35] Takeo Igarashi and Dennis Cosgrove. Adaptive unwrapping for interactive texture painting. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 209–216. ACM, 2001.
- [36] Johannes Schmid, Martin Sebastian Senn, Markus Gross, and Robert W Sumner. Overcoat : an implicit canvas for 3d painting. In *ACM Transactions on Graphics (TOG)*, volume 30, page 28. ACM, 2011.

- [37] Sriram Karthik Badam, Senthil Chandrasegaran, Niklas Elmqvist, and Karthik Ramani. Tracing and sketching performance using blunt-tipped styli on direct-touch tablets. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, pages 193–200. ACM, 2014.
- [38] Daniel Dixon, Manoj Prasad, and Tracy Hammond. icandraw : using sketch recognition and corrective feedback to assist a user in drawing human faces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 897–906. ACM, 2010.
- [39] Johnny Accot and Shumin Zhai. Beyond fitts’ law : Models for trajectory-based hci tasks. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, CHI’97*, pages 295–302, New York, NY, USA, 1997. ACM. ISBN 0-89791-802-9. doi: 10.1145/258549.258760. URL <http://doi.acm.org/10.1145/258549.258760>.
- [40] Helmut Alt and Michael Godau. Computing the fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 05(01n02) :75–91, 1995. doi: 10.1142/S0218195995000064. URL <http://www.worldscientific.com/doi/abs/10.1142/S0218195995000064>.
- [41] E. Belogay, C. Cabrelli, U. Molter, and R. Shonkwiler. Calculating the hausdorff distance between curves. *Information Processing Letters*, 64(1) :17 – 22, 1997. ISSN 0020-0190. doi: [http://dx.doi.org/10.1016/S0020-0190\(97\)00140-3](http://dx.doi.org/10.1016/S0020-0190(97)00140-3). URL <http://www.sciencedirect.com/science/article/pii/S0020019097001403>.
- [42] Russell Owen, Gordon Kurtenbach, George Fitzmaurice, Thomas Baudel, and Bill Buxton. When it gets more difficult, use both hands : Exploring bimanual curve manipulation. In *Proceedings of Graphics Interface 2005, GI’05*, pages 17–24, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society. ISBN 1-56881-265-5. URL <http://dl.acm.org/citation.cfm?id=1089508.1089512>.
- [43] Y. Shon and S. McMains. Evaluation of drawing on 3d surfaces with haptics. *Computer Graphics and Applications*, 24(6) :40–50, Nov 2004. ISSN 0272-1716. doi: 10.1109/MCG.2004.49.
- [44] I. Scott MacKenzie, Tatu Kauppinen, and Miika Silfverberg. Accuracy measures for evaluating computer pointing devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI’01*, pages 9–16, New York, NY, USA, 2001. ACM. ISBN 1-58113-327-8. doi: 10.1145/365024.365028. URL <http://doi.acm.org/10.1145/365024.365028>.



- [45] D.F. Keefe, R.C. Zeleznik, and D.H. Laidlaw. Drawing on air : Input techniques for controlled 3d line illustration. *Visualization and Computer Graphics*, 13(5) : 1067–1081, Sept 2007. ISSN 1077-2626. doi: 10.1109/TVCG.2007.1060.
- [46] Blender. Manual : Modeling, 2015. URL <http://www.blender.org/manual/modeling/introduction.html>.
- [47] Pixologic. Zbrush, 2015. URL <http://pixologic.com/>.
- [48] Blender. Manual : Lighting, 2015. URL [http://www.blender.org/manual/render/blender\\_render/lighting/introduction.html](http://www.blender.org/manual/render/blender_render/lighting/introduction.html).
- [49] Blender. Manual : Animation, 2015. URL <http://www.blender.org/manual/animation/introduction.html>.
- [50] Blender. Manual : Rendering, 2015. URL <http://www.blender.org/manual/render/introduction.html>.
- [51] Blender. Manual : Compositing, 2015. URL [http://www.blender.org/manual/composite\\_nodes/introduction.html](http://www.blender.org/manual/composite_nodes/introduction.html).
- [52] James F Blinn and Martin E Newell. Texture and reflection in computer generated images. *Communications of the ACM*, 19(10) :542–547, 1976.
- [53] P.S. Heckbert. Survey of texture mapping. *Computer Graphics and Applications, IEEE*, 6(11) :56–67, Nov 1986. ISSN 0272-1716. doi: 10.1109/MCG.1986.276672. URL <https://www.cs.cmu.edu/~ph/texsurv.pdf>.
- [54] Jonas Gomes. *Warping and morphing of graphical objects*, volume 1. Morgan Kaufmann, 1999.
- [55] D.S. Ebert. *Texturing & Modeling : A Procedural Approach*. Morgan Kaufmann series in computer graphics and geometric modeling. Morgan Kaufmann, 2003. ISBN 9781558608481. URL <https://books.google.fr/books?id=F4t-5zSsqR4C>.
- [56] Paul S Heckbert. Fundamentals of texture mapping and image warping. Master’s thesis, Citeseer, 1989.
- [57] Edwin Catmull. A subdivision algorithm for computer display of curved surfaces. Technical report, DTIC Document, 1974.
- [58] James Frederick Blinn. *Computer display of curved surfaces*. PhD thesis, 1978.
- [59] Geoffrey Y Gardner. Visual simulation of clouds. In *ACM Siggraph Computer Graphics*, volume 19, pages 297–304. ACM, 1985.

- [60] James F Blinn. Simulation of wrinkled surfaces. In *ACM SIGGRAPH computer graphics*, volume 12, pages 286–292. ACM, 1978.
- [61] Charles Richard Francis Maunder. *Algebraic topology*. Courier Corporation, 1996.
- [62] William T Tutte. How to draw a graph. *Proc. London Math. Soc.*, 13(3) :743–768, 1963.
- [63] Ed Catmull and Alvy Ray Smith. 3-d transformations of images in scanline order. In *ACM SIGGRAPH Computer Graphics*, volume 14, pages 279–285. ACM, 1980.
- [64] Eugene Fiume, Alain Fournier, and V Canale. Conformal texture mapping. In *Proceedings of Eurographics 87*, pages 53–64, 1987.
- [65] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 173–182. ACM, 1995.
- [66] Michael S Floater. Parametrization and smooth approximation of surface triangulations. *Computer aided geometric design*, 14(3) :231–250, 1997.
- [67] Kai Hormann and Günther Greiner. Mips : An efficient global parametrization method. Technical report, DTIC Document, 2000.
- [68] Alla Sheffer and Eric De Sturler. Surface parameterization for meshing by triangulation flattening. In *Proc. 9th International Meshing Roundtable*. Citeseer, 2000.
- [69] Mathieu Desbrun, Mark Meyer, and Pierre Alliez. Intrinsic parameterizations of surface meshes. In *Computer Graphics Forum*, volume 21, pages 209–218. Wiley Online Library, 2002.
- [70] Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérôme Maillot. Least squares conformal maps for automatic texture atlas generation. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 362–371. ACM, 2002.
- [71] Michael S Floater. Mean value coordinates. *Computer aided geometric design*, 20(1) :19–27, 2003.
- [72] Alla Sheffer, Bruno Lévy, Maxim Mogilnitsky, and Alexander Bogomyakov. Abf++ : fast and robust angle based flattening. *ACM Transactions on Graphics (TOG)*, 24(2) :311–330, 2005.

- [73] Liliya Kharevych, Boris Springborn, and Peter Schröder. Discrete conformal mappings via circle patterns. *ACM Transactions on Graphics (TOG)*, 25(2) :412–438, 2006.
- [74] Pedro V Sander, John Snyder, Steven J Gortler, and Hugues Hoppe. Texture mapping progressive meshes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 409–416. ACM, 2001.
- [75] G. Zigelman, R. Kimmel, and N. Kiryati. Texture mapping using surface flattening via multidimensional scaling. *Visualization and Computer Graphics, IEEE Transactions on*, 8(2) :198–207, Apr 2002. ISSN 1077-2626. doi: 10.1109/2945.998671.
- [76] Patrick Degener, Jan Meseth, and Reinhard Klein. An adaptable surface parameterization method. *IMR*, 3 :201–213, 2003.
- [77] Michael Floater and Kai Hormann. Parameterization of triangulations and unorganized points. In *Tutorials on Multiresolution in Geometric Modelling*, pages 287–316. Springer, 2002.
- [78] MS Floater and K Hormann. Surface parameterization : a tutorial and survey. *advances in multiresolution for geometric modelling 2005*, 2005.
- [79] Alla Sheffer, Emil Praun, and Kenneth Rose. Mesh parameterization methods and their applications. *Foundations and Trends® in Computer Graphics and Vision*, 2(2) :105–171, 2006.
- [80] Pierre Alliez, Mark Meyer, and Mathieu Desbrun. Interactive geometry remeshing. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 347–354. ACM, 2002.
- [81] Jeff Erickson and Sarel Har-Peled. Optimally cutting a surface into a disk. *Discrete & Computational Geometry*, 31(1) :37–59, 2004.
- [82] Xianfeng Gu, Steven J Gortler, and Hugues Hoppe. Geometry images. *ACM Transactions on Graphics (TOG)*, 21(3) :355–361, 2002.
- [83] Xianfeng Gu and Shing-Tung Yau. Computing conformal structure of surfaces. *arXiv preprint cs/0212043*, 2002.
- [84] Xianfeng Gu and Shing-Tung Yau. Global conformal surface parameterization. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 127–137. Eurographics Association, 2003.
- [85] J McCartney, BK Hinds, and BL Seow. The flattening of triangulated surfaces incorporating darts and gussets. *Computer-Aided Design*, 31(4) :249–260, 1999.

- [86] Dan Piponi and George Borshukov. Seamless texture mapping of subdivision surfaces by model pelting and texture blending. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 471–478. ACM Press/Addison-Wesley Publishing Co., 2000.
- [87] Alla Sheffer and John C Hart. Seamster : inconspicuous low-distortion texture seam layout. In *Visualization, 2002. VIS 2002. IEEE*, pages 291–298. IEEE, 2002.
- [88] Olga Sorkine, Daniel Cohen-Or, Rony Goldenthal, and Dani Lischinski. Bounded-distortion piecewise mesh parameterization. In *Proceedings of the conference on Visualization'02*, pages 355–362. IEEE Computer Society, 2002.
- [89] Chakib Bennis, Jean-Marc Vézien, and Gérard Iglésias. Piecewise surface flattening for non-distorted texture mapping. In *ACM SIGGRAPH computer graphics*, volume 25, pages 237–246. ACM, 1991.
- [90] Eric Bier, Kenneth R Sloan Jr, et al. Two-part texture mappings. *Computer Graphics and Applications, IEEE*, 6(9) :40–53, 1986.
- [91] Aaron WF Lee, Wim Sweldens, Peter Schröder, Lawrence Cowsar, and David Dobkin. Maps : Multiresolution adaptive parameterization of surfaces. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 95–104. ACM, 1998.
- [92] John Snyder, Pedro V Sander, Zoe J Wood, Steven Gortler, and Hugues Hoppe. Multi-chart geometry images. 2003.
- [93] Kun Zhou, John Snyder, Baining Guo, and Heung-Yeung Shum. Iso-charts : stretch-driven mesh parameterization using spectral analysis. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 45–54. ACM, 2004.
- [94] Eugene Zhang, Konstantin Mischaikow, and Greg Turk. Feature-based surface parameterization and texture mapping. *ACM Transactions on Graphics (TOG)*, 24(1) :1–27, 2005.
- [95] Vladislav Kraevoy, Alla Sheffer, and Craig Gotsman. *Matchmaker : constructing constrained texture maps*, volume 22. ACM, 2003.
- [96] Kun Zhou, Xi Wang, Yiying Tong, Mathieu Desbrun, Baining Guo, and Heung-Yeung Shum. Texturemontage. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 1148–1155. ACM, 2005.

- [97] Ilya Eckstein, Vitaly Surazhsky, and Craig Gotsman. Texture mapping with hard constraints. In *Computer Graphics Forum*, volume 20, pages 95–104. Wiley Online Library, 2001.
- [98] Bruno Lévy. Constrained texture mapping for polygonal meshes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM, 2001.
- [99] John Schreiner, Arul Asirvatham, Emil Praun, and Hugues Hoppe. Inter-surface mapping. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 870–877. ACM, 2004.
- [100] Demetri Terzopoulos and Manuela Vasilescu. Sampling and reconstruction with adaptive meshes. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 70–75. IEEE, 1991.
- [101] Peter-Pike J Sloan, David M Weinstein, and J. Dean Brederson. Importance driven texture coordinate optimization. In *Computer Graphics Forum*, volume 17, pages 97–104, 1998.
- [102] Adam Hunter and Jonathan D Cohen. Uniform frequency images : adding geometry to images to produce space-efficient textures. In *Visualization 2000. Proceedings*, pages 243–250. IEEE, 2000.
- [103] Laurent Balmelli, Gabriel Taubin, and Fausto Bernardini. Space-optimized texture maps. In *Computer Graphics Forum*, volume 21, pages 411–420. Wiley Online Library, 2002.
- [104] Pedro V Sander, Hugues Hoppe, Steven Gortler, and John Snyder. Signal-specialized parameterization. 2002.
- [105] Jérôme Maillot, Hussein Yahia, and Anne Verroust. Interactive texture mapping. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 27–34. ACM, 1993.
- [106] Qian Sun, Long Zhang, Minqi Zhang, Xiang Ying, Shi-Qing Xin, Jiazhi Xia, and Ying He. Texture brush : an interactive surface texturing interface. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 153–160. ACM, 2013.
- [107] Pat Hanrahan and Paul Haeberli. Direct wysiwyg painting and texturing on 3d shapes. In *ACM SIGGRAPH Computer Graphics*, volume 24, pages 215–223. ACM, 1990.
- [108] Ergun Akleman. Implicit surface painting. In *Implicit Surfaces 98*, volume 2, 1998.

- [109] Eric Daniels. Deep canvas in disney's tarzan. In *ACM SIGGRAPH 99 Conference abstracts and applications*, page 200. ACM, 1999.
- [110] David Benson and Joel Davis. Octree textures. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 785–790. ACM, 2002.
- [111] David (grue) DeBry, Jonathan Gibbs, Devorah DeLeon Petty, and Nate Robins. Painting and rendering textures on unparameterized models. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '02*, pages 763–768, New York, NY, USA, 2002. ACM. ISBN 1-58113-521-1. doi: 10.1145/566570.566649. URL <http://doi.acm.org/10.1145/566570.566649>.
- [112] Richard Szeliski and David Tonnesen. *Surface modeling with oriented particle systems*, volume 26. ACM, 1992.
- [113] Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross. Surfels : Surface elements as rendering primitives. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 335–342. ACM Press/Addison-Wesley Publishing Co., 2000.
- [114] Matthias Zwicker, Mark Pauly, Oliver Knoll, and Markus Gross. Pointshop 3d : an interactive system for point-based surface editing. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 322–329. ACM, 2002.
- [115] Bart Adams, Martin Wicke, Philip Dutré, Markus Gross, Mark Pauly, and Matthias Teschner. Interactive 3d painting on point-sampled objects. In *Eurographics Symposium on Point-Based Graphics*, pages 2–4. Citeseer, 2004.
- [116] David Johnson, Thomas V Thompson, Matthew Kaplan, Donald Nelson, Elaine Cohen, et al. Painting textures with a haptic interface. In *Virtual Reality, 1999. Proceedings., IEEE*, pages 282–285. IEEE, 1999.
- [117] Bill Baxter, Vincent Scheib, Ming C Lin, and Dinesh Manocha. Dab : interactive haptic painting with 3d virtual brushes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 461–468. ACM, 2001.
- [118] Nelson SH Chu and Chiew-Lan Tai. An efficient brush model for physically-based 3d painting. In *Computer Graphics and Applications, 2002. Proceedings. 10th Pacific Conference on*, pages 413–421. IEEE, 2002.
- [119] Songhua Xu, Min Tang, Francis Lau, and Yunhe Pan. A solid model based virtual hairy brush. In *Computer Graphics Forum*, volume 21, pages 299–308. Wiley Online Library, 2002.

- [120] Songhua Xu, Francis Lau, Feng Tang, and Yunhe Pan. Advanced design for a realistic virtual brush. In *Computer Graphics Forum*, volume 22, pages 533–542. Wiley Online Library, 2003.
- [121] William V Baxter and Ming C Lin. A versatile interactive 3d brush model. In *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on*, pages 319–328. IEEE, 2004.
- [122] Randolph Schärfig and Kai Hormann. Hardware accelerated 3d mesh painting. In *VMV*, pages 211–218, 2010.
- [123] Julie Daily and Kenneth Kiss. 3d painting : Paradigms for painting in a new dimension. In *Conference companion on Human factors in computing systems*, pages 296–297. ACM, 1995.
- [124] Arthur D Gregory, Stephen Ehmann, Ming C Lin, et al. intouch : Interactive multiresolution modeling and 3d painting with a haptic interface. In *Virtual Reality, 2000. Proceedings. IEEE*, pages 45–52. IEEE, 2000.
- [125] Mark Foskey, Miguel Otaduy, Ming C Lin, et al. Artnova : Touch-enabled 3d model design. In *Virtual Reality, 2002. Proceedings. IEEE*, pages 119–126. IEEE, 2002.
- [126] Maneesh Agrawala, Andrew C Beers, and Marc Levoy. 3d painting on scanned surfaces. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 145–ff. ACM, 1995.
- [127] Robert D Kalnins, Lee Markosian, Barbara J Meier, Michael A Kowalski, Joseph C Lee, Philip L Davidson, Matthew Webb, John F Hughes, and Adam Finkelstein. Wysiwyg npr : Drawing strokes directly on 3d models. *ACM Transactions on Graphics (TOG)*, 21(3) :755–762, 2002.
- [128] Daniel F Keefe, Daniel Acevedo Feliz, Tomer Moscovich, David H Laidlaw, and Joseph J LaViola Jr. Cavepainting : a fully immersive 3d artistic medium and interactive experience. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 85–93. ACM, 2001.
- [129] Alexis Clay, Jean-Christophe Lombardo, Nadine Couture, and Julien Conan. Bimanual 3d painting : an interaction paradigm for augmented reality live performance. In *HCITOUCH-Human-Computer Interaction, Tourism and Cultural Heritage-2012*, 2012.
- [130] Desney S Tan, George G Robertson, and Mary Czerwinski. Exploring 3d navigation : combining speed-coupled flying with orbiting. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 418–425. ACM, 2001.

- [131] Marc Christie, Patrick Olivier, and Jean-Marie Normand. Camera control in computer graphics. In *Computer Graphics Forum*, volume 27, pages 2197–2218. Wiley Online Library, 2008.
- [132] Shumin Zhai. *Human performance in six degree of freedom input control*. PhD thesis, University of Toronto, 1995.
- [133] Bernd Froehlich, Jan Hochstrate, Verena Skuk, and Anke Huckauf. The globefish and the globemouse : two new six degree of freedom input devices for graphics applications. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 191–199. ACM, 2006.
- [134] Damien Marchal, Clément Moerman, Géry Casiez, and Nicolas Roussel. Designing intuitive multi-touch 3d navigation techniques. In *Human-Computer Interaction—INTERACT 2013*, pages 19–36. Springer, 2013.
- [135] Mark Hancock, Thomas Ten Cate, and Sheelagh Carpendale. Sticky tools : full 6dof force-based interaction for multi-touch tables. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 133–140. ACM, 2009.
- [136] Jeffrey S Pierce, Andrew S Forsberg, Matthew J Conway, Seung Hong, Robert C Zeleznik, and Mark R Mine. Image plane interaction techniques in 3d immersive environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 39–ff. ACM, 1997.
- [137] Jason L Reisman, Philip L Davidson, and Jefferson Y Han. A screen-space formulation for 2d and 3d direct manipulation. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 69–78. ACM, 2009.
- [138] Anthony Martinet, Géry Casiez, and Laurent Grisoni. The effect of dof separation in 3d manipulation tasks with multi-touch displays. In *Proc. of VRST*, pages pp 111–118. IEEE, 2010. ISBN 978-1-4503-0441-2. doi: 10.1145/1889863.1889888. URL <http://doi.acm.org/10.1145/1889863.1889888>.
- [139] Oscar Kin-Chung Au, Chiew-Lan Tai, and Hongbo Fu. Multitouch Gestures for Constrained Transformation of 3D Objects. *Computer Graphics Forum*, 31(2pt3) : 651–660, May 2012. ISSN 01677055. doi: 10.1111/j.1467-8659.2012.03044.x. URL <http://doi.wiley.com/10.1111/j.1467-8659.2012.03044.x>.
- [140] Jingbo Liu, Oscar Kin-Chung Au, Hongbo Fu, and Chiew-Lan Tai. Two-Finger Gestures for 6DOF Manipulation of 3D Objects. *Computer Graphics Forum*, 31(7) :pp 2047–2055, September 2012. ISSN 01677055. doi: 10.1111/



- j.1467-8659.2012.03197.x. URL <http://doi.wiley.com/10.1111/j.1467-8659.2012.03197.x>.
- [141] Doug Scheurich and Wolfgang Stuerzlinger. A one-handed multi-touch mating method for 3d rotations. In *Proc. of CHI, Ext. Abstract*, page pp 1623, 2013. doi: 10.1145/2468356.2468646. URL <http://dl.acm.org/citation.cfm?doid=2468356.2468646>.
- [142] Gordon Kurtenbach, George Fitzmaurice, Thomas Baudel, and Bill Buxton. The design of a gui paradigm based on tablets, two-hands, and transparency. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 35–42. ACM, 1997.
- [143] Michael Chen, S Joy Mountford, and Abigail Sellen. A study in interactive 3-d rotation using 2-d control devices. In *ACM SIGGRAPH Computer Graphics*, volume 22, pages 121–129. ACM, 1988.
- [144] Ken Shoemake. Arcball : a user interface for specifying three-dimensional orientation using a mouse. In *Graphics Interface*, volume 92, pages 151–156, 1992.
- [145] Ken Hinckley, Joe Tullio, Randy Pausch, Dennis Proffitt, and Neal Kassell. Usability analysis of 3d rotation techniques. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 1–10. ACM, 1997.
- [146] Ragnar Bade, Felix Ritter, and Bernhard Preim. Usability comparison of mouse-based interaction techniques for predictable 3d rotation. *Smart graphics*, 2005. URL [http://link.springer.com/chapter/10.1007/11536482\\_12](http://link.springer.com/chapter/10.1007/11536482_12).
- [147] Yao Jun Zhao, D. Shuralyov, and W. Stuerzlinger. Comparison of multiple 3d rotation methods. In *Proc. of VECIMS*, pages pp 1–5, 2011. doi: 10.1109/VECIMS.2011.6053855.
- [148] Michael Gleicher and Andrew Witkin. Through-the-lens camera control. In *ACM SIGGRAPH Computer Graphics*, volume 26, pages 331–340. ACM, 1992.
- [149] Anthony Martinet, Gery Casiez, and Laurent Grisoni. The design and evaluation of 3d positioning techniques for multi-touch displays. In *3D User Interfaces (3DUI), 2010 IEEE Symposium on*, pages 115–118. IEEE, 2010.
- [150] Wendell R Garner. *The processing of information and structure*. Psychology Press, 2014.
- [151] Robert JK Jacob, Linda E Sibert, Daniel C McFarlane, and M Preston Mullen Jr. Integrality and separability of input devices. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 1(1) :3–26, 1994.

- [152] Ivan Poupyrev, Suzanne Weghorst, and Sidney Fels. Non-isomorphic 3d rotational techniques. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 540–547. ACM, 2000.
- [153] Yanqing Wang, Christine L. MacKenzie, Valerie A. Summers, and Kellogg S. Booth. The structure of object transportation and orientation in human-computer interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '98, pages 312–319, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-30987-4. doi: 10.1145/274644.274688. URL <http://dx.doi.org/10.1145/274644.274688>.
- [154] Maurice R. Masliah and Paul Milgram. Measuring the allocation of control in a 6 degree-of-freedom docking experiment. In *Proc. of CHI*, pages pp 25–32. ACM, 2000. ISBN 1-58113-216-6.
- [155] Jock D Mackinlay, Stuart K Card, and George G Robertson. Rapid controlled movement through a virtual 3d workspace. In *ACM SIGGRAPH Computer Graphics*, volume 24, pages 171–176. ACM, 1990.
- [156] Cary B Phillips, Norman I Badler, and John Granieri. Automatic viewing control for 3d direct manipulation. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 71–74. ACM, 1992.
- [157] Robert Zeleznik and Andrew Forsberg. Unicam : 2d gestural camera controls for 3d environments. In *Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 169–173. ACM, 1999.
- [158] Azam Khan, Ben Komalo, Jos Stam, George Fitzmaurice, and Gordon Kurtenbach. Hovercam : interactive 3d navigation for proximal object inspection. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games*, pages 73–80. ACM, 2005.
- [159] Fabrice Declec, Martin Hachet, and P. Guitton. Tech-note : Scruticam : Camera manipulation technique for 3d objects inspection. In *3D User Interfaces, 2009. 3DUI 2009. IEEE Symposium on*, pages 19–22, March 2009. doi: 10.1109/3DUI.2009.4811200.
- [160] Michaël Ortega and Thomas Vincent. Direct drawing on 3d shapes with automated camera control. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2047–2050. ACM, 2014.
- [161] Jiazhi Xia, Shenghui Liao, and Juncong Lin. Rotation-aware layerpaint system. In *Advances in Visual Computing*, pages 60–68. Springer, 2013.

- [162] Michael Ortega, Wolfgang Stuerzlinger, and Doug Scheurich. Shocam : A 3d orbiting algorithm. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 119–128. ACM, 2015.
- [163] Kenneth Flowers. Handedness and controlled movement. *British Journal of Psychology*, 66(1) :39–52, 1975.
- [164] Richard A Schmidt, Howard Zelaznik, Brian Hawkins, James S Frank, and John T Quinn Jr. Motor-output variability : a theory for the accuracy of rapid motor acts. *Psychological review*, 86(5) :415, 1979.
- [165] John I Todor and Thomas Doane. Handedness and hemispheric asymmetry in the control of movements. *Journal of Motor Behavior*, 10(4) :295–300, 1978.
- [166] Michael Peters. Constraints in the performance of bimanual tasks and their expression in unskilled and skilled subjects. *The Quarterly Journal of Experimental Psychology*, 37(2) :171–196, 1985.
- [167] Jeffery J Summers and Tricia M Kennedy. Strategies in the production of a 5 : 3 polyrhythm. *Human Movement Science*, 11(1) :101–112, 1992.
- [168] Ravin Balakrishnan and Ken Hinckley. Symmetric bimanual interaction. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 33–40. ACM, 2000.
- [169] Yves Guiard. Asymmetric division of labor in human skilled bimanual action : The kinematic chain as a model. *Journal of motor behavior*, 19(4) :486–517, 1987.
- [170] Ken Hinckley, Randy Pausch, Dennis Proffitt, James Patten, and Neal Kassell. Cooperative bimanual action. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 27–34. ACM, 1997.
- [171] Ravin Balakrishnan and Ken Hinckley. The role of kinesthetic reference frames in two-handed input performance. In *Proceedings of the 12th annual ACM symposium on User interface software and technology*, pages 171–178. ACM, 1999.
- [172] Eric A Bier, Maureen C Stone, Ken Pier, William Buxton, and Tony D DeRose. Toolglass and magic lenses : the see-through interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 73–80. ACM, 1993.
- [173] C Wickens and J Hollands. Chapter 3 : Attention in perception and display space. *Engineering psychology and human performance*, pages 74–115, 1992.

- [174] Ken Hinckley, Randy Pausch, and Dennis Proffitt. Attention and visual feedback : the bimanual frame of reference. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 121–ff. ACM, 1997.
- [175] Colin G Butler and Robert St Amant. Habilisdraw dt : a bimanual tool-based direct manipulation drawing environment. In *CHI'04 extended abstracts on human factors in computing systems*, pages 1301–1304. ACM, 2004.
- [176] Stéphane Chatty. Extending a graphical toolkit for two-handed interaction. In *Proceedings of the 7th annual ACM symposium on User interface software and technology*, pages 195–204. ACM, 1994.
- [177] Lawrence D Cutler, Bernd Fröhlich, and Pat Hanrahan. Two-handed direct manipulation on the responsive workbench. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 107–114. ACM, 1997.
- [178] George W Fitzmaurice, Hiroshi Ishii, and William AS Buxton. Bricks : laying the foundations for graspable user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 442–449. ACM Press/Addison-Wesley Publishing Co., 1995.
- [179] Ken Hinckley, Mary Czerwinski, and Mike Sinclair. Interaction and modeling techniques for desktop two-handed input. In *Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 49–58. ACM, 1998.
- [180] Paul Kabbash, William Buxton, and Abigail Sellen. Two-handed input in a compound task. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 417–423. ACM, 1994.
- [181] Gordon Kurtenbach, George Fitzmaurice, Thomas Baudel, and Bill Buxton. The design of a gui paradigm based on tablets, two-hands, and transparency. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 35–42. ACM, 1997.
- [182] Mark R Mine, Frederick P Brooks Jr, and Carlo H Sequin. Moving objects in space : exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 19–26. ACM Press/Addison-Wesley Publishing Co., 1997.
- [183] Roope Raisamo and Kari-Jouko Rähkä. *Techniques for aligning objects in drawing programs*. University of Tampere, Department of Computer Science, 1996.
- [184] Chris D Shaw and Mark Green. Thred : a two-handed design system. *Multimedia Systems*, 5(2) :126–139, 1997.

- [185] Ravin Balakrishnan and Gordon Kurtenbach. Exploring bimanual camera control and object manipulation in 3d graphics interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 56–62. ACM, 1999.
- [186] William Buxton and Brad Myers. A study in two-handed input. In *ACM SIGCHI Bulletin*, volume 17, pages 321–326. ACM, 1986.
- [187] Kenrick Kin, Maneesh Agrawala, and Tony DeRose. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In *Proceedings of Graphics interface 2009*, pages 119–124. Canadian Information Processing Society, 2009.
- [188] Andrea Leganchuk, Shumin Zhai, and William Buxton. Manual and cognitive benefits of two-handed input : an experimental study. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 5(4) :326–359, 1998.
- [189] Ken Hinckley, Randy Pausch, Dennis Proffitt, and Neal F Kassell. Two-handed virtual manipulation. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 5(3) :260–302, 1998.
- [190] Ken Hinckley, Randy Pausch, John C Goble, and Neal F Kassell. Passive real-world interface props for neurosurgical visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 452–458. ACM, 1994.
- [191] Emanuel Sachs, Andrew Roberts, and David Stoops. 3-draw : A tool for designing 3d shapes. *IEEE Computer Graphics and Applications*, 11(6) :18–26, 1991.
- [192] Pedro Lopes, Daniel Mendes, Bruno Araújo, and Joaquim A Jorge. Combining bimanual manipulation and pen-based input for 3d modelling. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*, pages 15–22. ACM, 2011.
- [193] Luis Serra, Ng Hern, Chua Beng Choon, and Timothy Poston. Interactive vessel tracing in volume data. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 131–ff. ACM, 1997.
- [194] Chris Shaw and Mark Green. Two-handed polygonal surface design. In *Proceedings of the 7th annual ACM symposium on User interface software and technology*, pages 205–212. ACM, 1994.
- [195] Richard Stoakley, Matthew J Conway, and Randy Pausch. Virtual reality on a wim : interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 265–272. ACM Press/Addison-Wesley Publishing Co., 1995.

- [196] Kiyoshi Kiyokawa, Haruo Takemura, Yoshiaki Katayama, Hidehiko Iwasa, and Naokazu Yokoya. Vlego : A simple two-handed modeling environment based on toy blocks. In *Proc. of ACM Simpo. on Virtual Reality Software and Technology*, pages 27–34, 1996.
- [197] Kiyoshi Kiyokawa, Haruo Takemura, and Naokazu Yokoya. Manipulation aid for two-handed 3-d designing within a shared virtual environment. In *HCI (2)*, pages 937–940, 1997.
- [198] Maarten W Gribnau and James M Hennessey. Comparing single-and two-handed 3d input for a 3d object assembly task. In *CHI 98 Cconference Summary on Human Factors in Computing Systems*, pages 233–234. ACM, 1998.
- [199] Fabrice Boyer, 2015. URL <https://play.google.com/store/apps/details?id=truesculpt.main>.
- [200] Duke Dev, 2015. URL <https://play.google.com/store/apps/details?id=com.dukedev.hdmodelviewer>.
- [201] François Bérard, Jessica Ip, Mitchel Benovoy, Dalia El-Shimy, Jeffrey R Blum, and Jeremy R Cooperstock. Did "minority report" get it wrong? superiority of the mouse over 3d input devices in a 3d placement task. In *Human-Computer Interaction-INTERACT 2009*, pages 400–414. Springer, 2009.
- [202] Colin Ware and Steven Osborne. Exploration and virtual camera control in virtual three dimensional environments. In *ACM SIGGRAPH Computer Graphics*, volume 24, pages 175–183. ACM, 1990.
- [203] Peter Brandl, Clifton Forlines, Daniel Wigdor, Michael Haller, and Chia Shen. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. In *Proceedings of the working conference on Advanced visual interfaces*, pages 154–161. ACM, 2008.
- [204] Ken Hinckley, Michel Pahud, and Bill Buxton. 38.2 : Direct display interaction via simultaneous pen+ multi-touch input. In *SID Symposium Digest of Technical Papers*, volume 41, pages 537–540. Wiley Online Library, 2010.
- [205] Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, and Bill Buxton. Manual deskterity : an exploration of simultaneous pen+ touch direct input. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, pages 2793–2802. ACM, 2010.
- [206] Michelle Annett, Anoop Gupta, and Walter F Bischof. Exploring and understanding unintended touch during direct pen interaction. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 21(5) :28, 2014.

- [207] Chi-Wing Fu, Jiazhi Xia, and Ying He. Layerpaint : a multi-layer interactive 3d painting interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 811–820. ACM, 2010.
- [208] Eli Brenner, Jeroen Smeets, and H. C. M. Remijnse-Tamerius. Curvature in hand movements as a result of visual misjudgements of direction. *Spatial Vision*, 15(4) :393–414, 2002. URL <http://booksandjournals.brillonline.com/content/journals/10.1163/156856802320401883>.
- [209] B. Shneiderman. Direct manipulation : A step beyond programming languages. *Computer*, 16(8) :pp 57 –69, aug. 1983. ISSN 0018-9162.
- [210] Manuel Veit, Antonio Capobianco, and Dominique Bechmann. Influence of degrees of freedom’s manipulation on performances during orientation tasks in virtual reality environments. In *Proc. of VRST*, pages pp 51–58. ACM, 2009. ISBN 978-1-60558-869-8. doi: 10.1145/1643928.1643942. URL <http://doi.acm.org/10.1145/1643928.1643942>.
- [211] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy : a sketching interface for 3d freeform design. In *ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. ACM, 2007. doi: <http://doi.acm.org/10.1145/1281500.1281532>. URL <http://doi.acm.org/10.1145/1281500.1281532>.
- [212] Even Entem, Loic Barthe, Marie-Paule Cani, Frederic Cordier, and Michiel van de Panne. Modeling 3d animals from a side-view sketch. *Computers & Graphics*, 46 :221 – 230, 2015. ISSN 0097-8493. doi: <http://dx.doi.org/10.1016/j.cag.2014.09.037>. URL <http://www.sciencedirect.com/science/article/pii/S0097849314001253>. Shape Modeling International 2014.
- [213] O. Palombi, A. Pihuit, and M.-P. Cani. 3d modeling of branching vessels from anatomical sketches : towards a new interactive teaching of anatomy. *Surgical and Radiologic Anatomy*, 33(7) :631–636, 2011. ISSN 0930-1038. doi: 10.1007/s00276-011-0827-5. URL <http://dx.doi.org/10.1007/s00276-011-0827-5>.
- [214] Élisabeth Rousset, François Bérard, and Michaël Ortega. Two-finger 3d rotations for novice users : surjective and integral interactions. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, pages 217–224. ACM, 2014.
- [215] Élisabeth Rousset, François Bérard, and Michaël Ortega. Study of the effect of the directness of the interaction on novice users when drawing straight lines. In *27ème conférence francophone sur l’Interaction Homme-Machine.*, page a14. ACM, 2015.

- [216] Geoff Walker. A review of technologies for sensing contact location on the surface of a display. *Journal of the Society for Information Display*, 20(8) :413–440, 2012.
- [217] Geoff Walker. Touch technologies tutorial. In *SID Display Week*, Display Week'14, 2014.
- [218] M. Kriss. *Handbook of Digital Imaging*. Wiley, 2015. ISBN 9780470510599. URL <https://books.google.fr/books?id=bh1wCAAAQBAJ>.
- [219] Hugh Le Caine. Touch-sensitive organ based on an electrostatic coupling device. *The Journal of the Acoustical Society of America*, 27(4) :781–786, 1955.
- [220] E Johnson. Touch display—a novel input/output device for computers. *Electronics Letters*, 1(8) :219–220, 1965.
- [221] E Johnson. Touch displays : A programmed man-machine interface. *Ergonomics*, 10(2) :271–277, 1967. doi: 10.1080/00140136708930868. URL <http://dx.doi.org/10.1080/00140136708930868>.
- [222] Bill Buxton. Multi-touch systems that i have known and loved, 2007. URL <http://www.billbuxton.com/multitouchOverview.html>.
- [223] William Buxton, Ralph Hill, and Peter Rowley. Issues and techniques in touch-sensitive tablet input. *ACM SIGGRAPH Computer Graphics*, 19(3) :215–224, 1985.
- [224] Gary Barrett and Ryomei Omote. Projected-capacitive touch technology. *Information Display*, 26(3) :16–21, 2010.
- [225] F. Bottari. Touch screen with an applied edge electrode pattern, October 25 2001. URL <https://www.google.fr/patents/US20010032698>. US Patent App. 09/886,452.
- [226] P.J. Richter, M.J. Kardauskas, and F.J. Bottari. Method of manufacturing a touch screen panel, January 3 2003. URL <https://www.google.fr/patents/CA2451259A1?c1=en>. CA Patent App. CA 2,451,259.
- [227] EloTouch Systems. History of elo, 2015. URL <http://www.elotouch.com/AboutElo/History/default.asp>.
- [228] George Hurst and William Colwell. Discriminating contact sensor, October 7 1975. URL <http://www.google.com/patents/US3911215>. US Patent 3,911,215.
- [229] Robert Adler and Peter J Desmares. An economical touch panel using saw absorption. *Ultrasonics, Ferroelectrics, and Frequency Control, IEEE Transactions on*, 34(2) :195–201, 1987.



- [230] A.R. Hardie-Bick. Touch pad, May 30 2013. URL <https://www.google.com.ar/patents/US20130135261>. US Patent App. 13/685,082.
- [231] J.B. Roberts. Iterative method for determining touch location, March 3 2009. URL <https://www.google.com/patents/US7499039>. US Patent 7,499,039.
- [232] F. Ebeling, R. Johnson, and R. Goldhor. Infrared light beam x-y position encoder for display devices, November 27 1973. URL <http://www.google.com/patents/US3775560>. US Patent 3,775,560.
- [233] B.K. Funk and C.L. Smith. Position detection apparatus, March 13 1979. URL <http://www.google.com.ar/patents/US4144449>. US Patent 4,144,449.
- [234] Geoff Walker. Camera-based optical touch technology. *Information Display*, 3 (11) :11, 2011.
- [235] L.R. Kasday. Touch position sensitive surface, November 20 1984. URL <http://www.google.com/patents/US4484179>. US Patent 4,484,179.
- [236] Jefferson Y Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118. ACM, 2005.
- [237] H. Bergström and T. Craven-Bartle. Scanning ftir systems for touch detection, 2012. URL <https://www.google.fr/patents/US20120153134>. US Patent App. 13/325,902.
- [238] T. Christiansson. Extracting touch data that represents one or more objects on a touch surface, 2012. URL <https://www.google.fr/patents/EP2491479A1?cl=en>. EP Patent App. EP20,100,825,282.
- [239] Ilya Rosenberg, Ken Perlin, Charles Hendee, Alex Grau, and Nadim Awad. The unmousepad : the future of touch sensing. In *SIGGRAPH'09 : Posters*, page 23. ACM, 2009.
- [240] Ilya D Rosenberg, Alexander Grau, Charles Hendee, Nadim Awad, and Ken Perlin. Impad : an inexpensive multi-touchpressure acquisition device. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pages 3217–3222. ACM, 2009.
- [241] I.D. Rosenberg. Method for manufacturing long force sensors using screen printing technology, July 26 2011. URL <http://www.google.com/patents/US7984544>. US Patent 7,984,544.
- [242] Sensel. Sensel morph on kickstarter. 2015. URL <https://www.kickstarter.com/projects/1152958674/the-sensel-morph-interaction-evolved>.

- [243] A.W.C.L. Murakami, T.W.C.L. Yamanami, T.W.C.L. Funahashi, T.W.C.L. Senda, and T.W.C.L. Chikami. Coordinate input system with input pen, April 20 1994. URL <https://www.google.fr/patents/EP0307667B1?cl=en>. EP Patent 0,307,667.
- [244] A.C.O.W.C.L. Murakami, T.C.O.W.C.L. Yamanami, T.C.O.W.C.L. Funahashi, T.C.O.W.C.L. Senda, K.C.O.W.C.L. Aoki, K.C.O.W.C.L. Sugiyama, N.C.O.W.C.L. Miyamori, A.C.O.W.C.L. Kikuchi, and M.C.O.W.C.L. Abe. Position detecting device, July 26 1995. URL <https://www.google.fr/patents/EP0322675B1?cl=en>. EP Patent 0,322,675.
- [245] I.L. Lai. Digital pressure sensitive electromagnetic pen, June 26 2001. URL <https://www.google.fr/patents/US6252182>. US Patent 6,252,182.
- [246] S. Yamamoto, M. Ito, Y. Oda, A. Tozer, and I. Scholey. An electronic device having a position sensor, March 30 2006. URL <https://www.google.fr/patents/WO2006032294A1?cl=en>. WO Patent App. PCT/EP2004/010,713.
- [247] C.C. Chao and C.W. Hsu. Electromagnetic inductive system with multi-induction loop layout and battery less pointer device and its method for locating the coordinate, April 15 2008. URL <https://www.google.fr/patents/US7358964>. US Patent 7,358,964.
- [248] Y. Stern and S. ROGEL. Pressure sensitive stylus for a digitizer, October 31 2013. URL <https://www.google.fr/patents/WO2013160887A1?cl=en>. WO Patent App. PCT/IL2012/050,143.
- [249] Y. Stern. Stylus for a digitizer system, August 28 2014. URL <https://www.google.fr/patents/WO2014128712A1?cl=en>. WO Patent App. PCT/IL2014/050,192.
- [250] Y. Stern. Piezoresistive sensor for a stylus, February 26 2015. URL <https://www.google.fr/patents/WO2014188417A3?cl=en>. WO Patent App. PCT/IL2014/050,442.
- [251] J.A. Harley and D.I. Simon. Active stylus, December 27 2012. URL <https://www.google.fr/patents/US20120327041>. US Patent App. 13/166,726.
- [252] S. Shahparnia, K. Sundara-Rajan, Y. Ali, and I. Bentov. Active stylus with high voltage, October 21 2014. URL <https://www.google.fr/patents/US8866767>. US Patent 8,866,767.
- [253] P. Ericson. Apparatus and methods relating to images, June 6 2002. URL <https://www.google.fr/patents/WO2001071644A8?cl=en>. WO Patent App. PCT/SE2001/000,609.

- [254] P.J. Richter, M.J. Kardauskas, and F.J. Bottari. Method of manufacturing a touch screen panel, January 3 2003. URL <https://www.google.fr/patents/CA2451259A1?c1=en>. CA Patent App. CA 2,451,259.
- [255] C. Fahraeus. Recording of information, May 20 2008. URL <https://www.google.fr/patents/CA2374808C?c1=en>. CA Patent 2,374,808.
- [256] T. Craven-Bartle. Optical component for a camera pen, November 13 2008. URL <https://www.google.fr/patents/W02008118085A3?c1=en>. WO Patent App. PCT/SE2008/050,335.
- [257] S. Lynggaard. Printing of coding patterns, August 27 2009. URL <https://www.google.fr/patents/W02008048169A3?c1=en>. WO Patent App. PCT/SE2007/000,917.
- [258] T. Craven-Bartle. Different aspects of electronic pens, January 13 2010. URL <https://www.google.fr/patents/EP2143040A2?c1=en>. EP Patent App. EP20,080,724,282.
- [259] Peter Vandoren, Tom Van Laerhoven, Luc Claesen, Johannes Taelman, Fabian Di Fiore, Frank Van Reeth, and Eddy Flerackers. Dip-it : digital infrared painting on an interactive table. In *CHI'08 Extended Abstracts on Human Factors in Computing Systems*, pages 2901–2906. ACM, 2008.
- [260] Peter Vandoren, Tom Van Laerhoven, Luc Claesen, Johannes Taelman, Chris Raymaekers, and Frank Van Reeth. Intupaint : Bridging the gap between physical and digital painting. In *Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on*, pages 65–72. IEEE, 2008.
- [261] Kimiko Ryokai, Stefan Marti, and Hiroshi Ishii. I/o brush : drawing with everyday objects as ink. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 303–310. ACM, 2004.
- [262] Kimiko Ryokai, Stefan Marti, and Hiroshi Ishii. Designing the world as your palette. In *CHI'05 Extended Abstracts on Human Factors in Computing Systems*, pages 1037–1049. ACM, 2005.
- [263] <http://www.isketchnote.com/fr/>. URL <http://www.isketchnote.com/fr/>.
- [264] Tristan Hautson, Timothée Jobert, Dominique David, Andrea Vassilev, and Saïfeddine Aloui. Procède et dispositif de localisation d'un objet magnetique, 2013. URL <http://bases-brevets.inpi.fr/fr/document/W02013144342.html?s=1438847872340&p=6&cHash=f717e2a61a54bfb021dc5d71b0bd2111>.

- [265] Tristan Hautson, Timothée Jobert, and Jean-Luc Vallejo. Ecran avec localisation d'objet magnetique, 2013. URL <http://bases-brevets.inpi.fr/fr/document/W02013144337.html?s=1438847872340&p=6&cHash=66428456c1c1082a87fa4d94898244ed>.
- [266] Tristan Hautson, Timothée Jobert, Rabeb Aloui, and Saifeddine Aloui. Procédé de reconnaissance automatique d'un objet magnetique mobile, 2014. URL <http://bases-brevets.inpi.fr/fr/document/W02014079740.html?s=1438847872340&p=6&cHash=7821694ef3ec390ca0f36bc48df492a5>.
- [267] Tristan Hautson, Timothée Jobert, and Jean-Luc Vallejo. Anneau magnetique apte a etre fixe de façon amovible sur un crayon ou un effaceur, 2014. URL <http://bases-brevets.inpi.fr/fr/document/W02014053526.html?s=1438847872340&p=6&cHash=9c7b04a1a81dc1e2ffb5bb277d21c567>.
- [268] Tristan Hautson, Timothée Jobert, and Jean-Luc Vallejo. Procédé de localisation d'ustensiles mobiles présents devant un réseau de magnétomètres, 2015. URL <http://bases-brevets.inpi.fr/fr/document/EP2889648.html?s=1438847872340&p=6&cHash=052f5ef6062bd04254024cf81fe3728f>.
- [269] Tristan Hautson, Timothée Jobert, and Jean-Luc Vallejo. Procédé de reconnaissance automatique d'un objet magnétique, 2015. URL <http://bases-brevets.inpi.fr/fr/document/W02014135421.html?s=1438847872340&p=6&cHash=93885a5defe62f918edbb52729980a42>.
- [270] I.D. Rosenberg, B.J. Bozarth, J.G. Beguin, T. Moscovich, S.J. Gao, and T. YUN. Magnetic stylus, January 10 2013. URL <http://www.google.com/patents/US20130009907>. US Patent App. 13/247,412.